Willkommen zu diesem dritten und letzten Kurs dieser Spezialisierung auf unbeaufsichtigtes Lernen, Empfehlungssysteme und verstärkendes Lernen. Während wir in den ersten beiden Kursen viel Zeit mit überwachtem Lernen verbracht haben, werden wir in diesem dritten und letzten Kurs über eine Reihe neuer Techniken sprechen, die über überwachtes Lernen hinausgehen, und Ihnen ein zusätzliches Set leistungsstarker Techniken an die Hand geben

Ich hoffe, dass es Ihnen Spaß macht, Ihre beiden Sets zu ergänzen, und wenn Sie diesen Kurs und diese Spezialisierung abgeschlossen haben, sind Sie meiner Meinung nach auf dem besten Weg, ein Experte für maschinelles Lernen zu werden. Lass uns einen Blick darauf werfen. Diese Woche beginnen wir mit betreutem Lernen. Insbesondere erfahren Sie mehr über Clustering-Algorithmen, eine Möglichkeit, Daten in Clustern zu gruppieren, sowie über die Erkennung von Anomalien. Beides sind Techniken, die heute von vielen Unternehmen in wichtigen kommerziellen Anwendungen eingesetzt werden. Am Ende dieser Woche wissen Sie, wie diese Algorithmen funktionieren, und sind in der Lage, sie auch bei Ihnen selbst zum Laufen zu bringen.

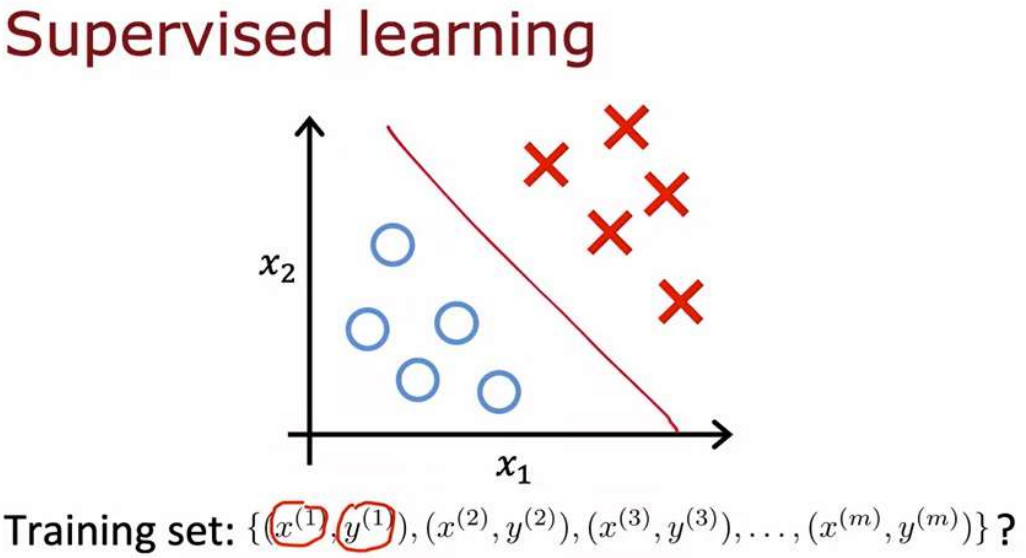
In der zweiten Woche lernen Sie Empfehlungssysteme kennen. Wie werden Ihnen Produkte oder Filme empfohlen, wenn Sie eine Online-Shopping-Website oder eine Video-Streaming-Website besuchen? Empfehlungssysteme sind eine der kommerziell wichtigsten Technologien für maschinelles Lernen. Sie transportieren Werte oder Produkte oder andere Dinge im Wert von vielen Milliarden US-Dollar und sind eine der Technologien, die trotz ihrer Bedeutung überraschend wenig Beachtung in der Wissenschaft finden. Aber ich hoffe, dass Sie in der zweiten Woche lernen, wie diese Systeme funktionieren, und in der Lage sind, eines selbst zu implementieren. Wenn Sie neugierig sind, wie Online-Anzeigensysteme funktionieren, vermittelt Ihnen die Beschreibung der Empfehlungssysteme auch einen Eindruck davon, wie diese großen Online-Anzeigentechnologieunternehmen entscheiden, welche Anzeigen Ihnen angezeigt werden.

In der dritten und letzten Woche dieses Kurses erfahren Sie mehr über das verstärkende Lernen. Sie haben vielleicht in den Nachrichten gelesen, dass Reinforcement Learning beim Spielen einer Vielzahl von Videospielen großartig ist und sogar Menschen übertrifft. Ich selbst habe Reinforcement Learning auch schon oft genutzt, um verschiedene Roboter zu steuern. Auch wenn Reinforcement Learning eine neue und aufstrebende Technologie ist, d. h. die Zahl der kommerziellen Anwendungen von Reinforcement Learning bei weitem nicht so groß ist wie die der anderen Techniken, die in dieser Woche oder in den beiden vorherigen Kursen dieser Spezialisierung behandelt werden, handelt es sich dennoch um eine Technologie ist spannend und eröffnet neue Möglichkeiten für die Möglichkeiten von Lernalgorithmen. In der letzten Woche implementieren Sie selbst Reinforcement Learning und nutzen es, um einen simulierten Mondlander zu landen. Wenn Sie später in diesem Kurs sehen, wie Sie selbst mit Ihrem eigenen Code arbeiten, werden Sie wahrscheinlich beeindruckt sein, was Sie durch Verstärkungslernen erreichen können. Ich freue mich sehr, mit Ihnen hier zu sein und über unüberwachtes Lernen, Empfehlungssysteme und verstärkendes Lernen zu sprechen. Fahren wir mit dem nächsten Video fort, in dem Sie mehr über einen wichtigen unbeaufsichtigten Lernalgorithmus namens Clustering-Algorithmus erfahren.

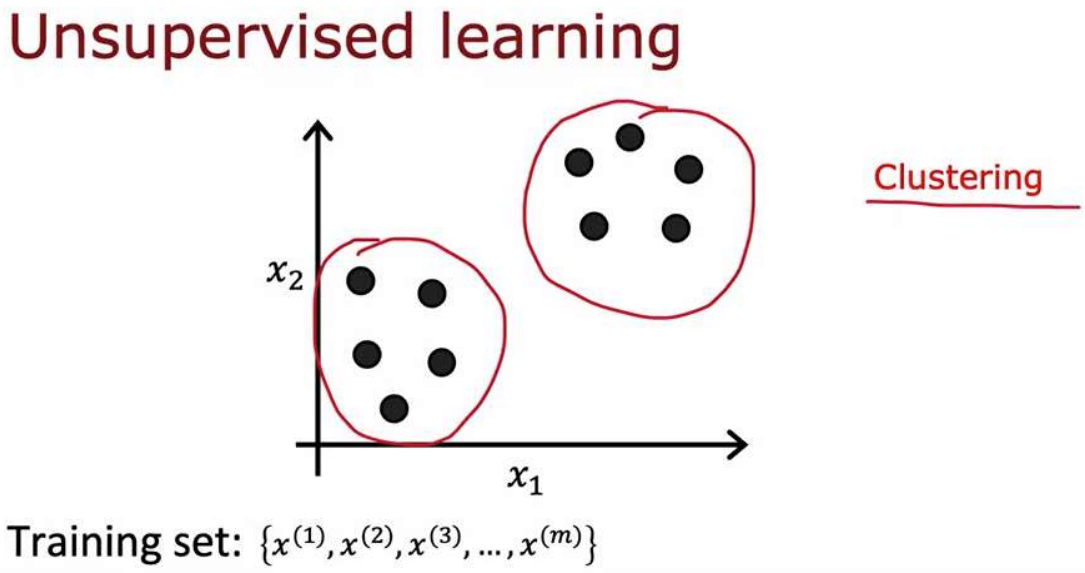
Was ist Clustering?

Ein Clustering-Algorithmus untersucht eine Reihe von Datenpunkten und findet automatisch Datenpunkte, die miteinander in Zusammenhang stehen oder ähnlich sind. Werfen wir einen Blick darauf, was das bedeutet. Lassen Sie mich Clustering, einen unüberwachten Lernalgorithmus, mit dem vergleichen, was Sie zuvor beim überwachten Lernen für die binäre Klassifizierung gesehen haben. Gegeben sei ein Datensatz wie dieser mit den Merkmalen x_1 und x_2. Beim überwachten Lernen hatten wir einen Trainingssatz mit den Eingabemerkmalen x und den Beschriftungen y. Wir könnten einen Datensatz wie diesen grafisch darstellen und beispielsweise einen logistischen

Regressionsalgorithmus oder ein neuronales Netzwerk anpassen, um eine solche Entscheidungsgrenze zu lernen. Beim überwachten Lernen umfasste der Datensatz sowohl die Eingaben x als auch die Zielausgaben y. Im Gegensatz dazu erhalten Sie beim unüberwachten Lernen einen Datensatz wie diesen nur mit x, aber nicht mit den Beschriftungen oder den Zielbeschriftungen y. Wenn ich einen Datensatz zeichne, sieht er daher so aus, mit nur Punkten und nicht mit zwei Klassen, die durch x und o gekennzeichnet sind.



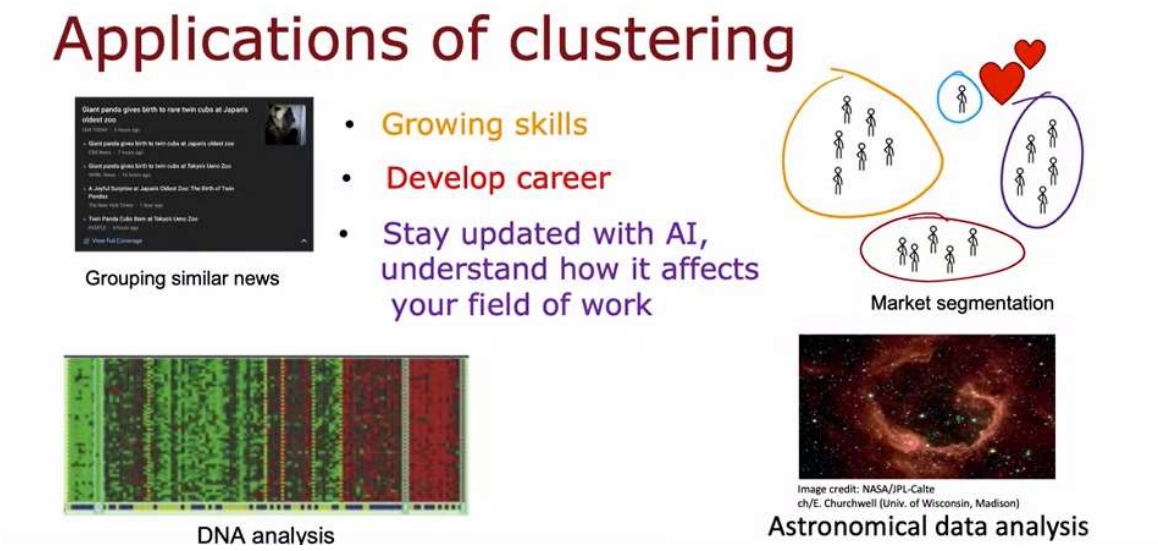
Da wir keine Zielbezeichnungen y haben, können wir dem Algorithmus nicht sagen, welche „richtige Antwort, y“ wir vorhersagen wollten. Stattdessen bitten wir den Algorithmus, etwas Interessantes an den Daten zu finden, das heißt, eine interessante Struktur dieser Daten zu finden. Aber der erste unbeaufsichtigte Lernalgorithmus, den Sie kennenlernen, wird Clustering-Algorithmus genannt, der nach einem bestimmten Strukturtyp in den Daten sucht. Schauen Sie sich den Datensatz nämlich so an und versuchen Sie herauszufinden, ob er in Cluster gruppiert werden kann, also Gruppen von Punkten, die einander ähnlich sind.



In diesem Fall könnte ein Clustering-Algorithmus feststellen, dass dieser Datensatz aus Daten aus zwei hier gezeigten Clustern besteht. Hier sind einige Anwendungen des Clusterings. In der ersten Woche des ersten Kurses haben Sie mich über das Gruppieren ähnlicher Nachrichtenartikel sprechen

hören, wie zum Beispiel die Geschichte über Pandas oder Marktsegmentierung, wo wir bei deeplearning.ai festgestellt haben, dass es viele Lernende gibt, die hierher kommen, weil Sie es vielleicht möchten Erweitern Sie Ihre Fähigkeiten, entwickeln Sie Ihre Karriere weiter oder bleiben Sie über KI auf dem Laufenden und verstehen Sie, wie sie sich auf Ihren Arbeitsbereich auswirkt. Wir möchten jedem mit einer dieser Fähigkeiten dabei helfen, etwas über maschinelles Lernen zu lernen.

Wenn Sie nicht in eine dieser Gruppen fallen, ist das auch völlig in Ordnung.

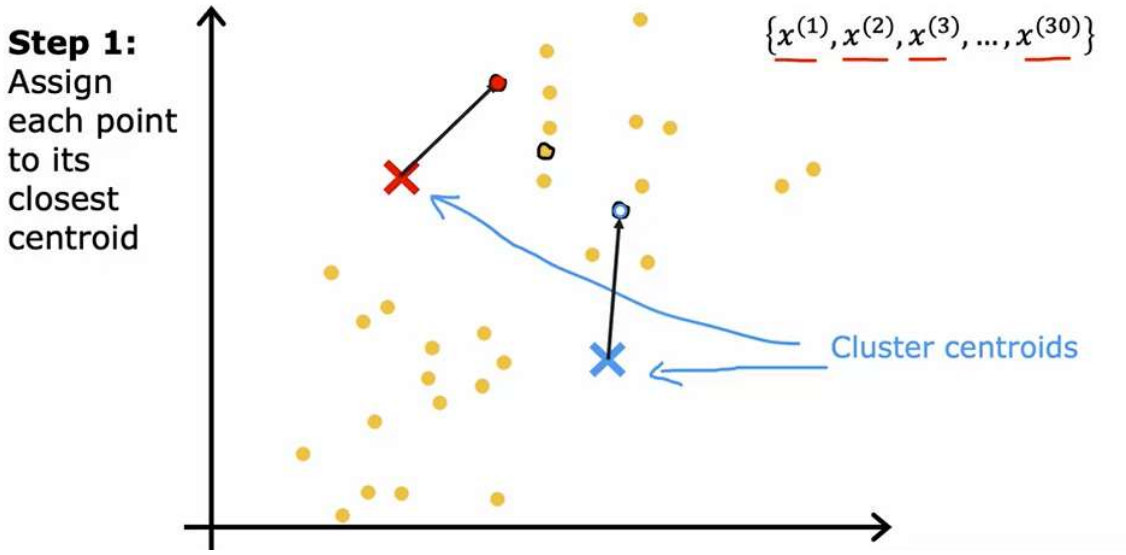


Ich hoffe, dass die Materialien von deeplearning.ai und Stanford Online auch für Sie nützlich sein werden. Clustering wurde auch zur Analyse von DNA-Daten verwendet, wobei Sie die genetischen Expressionsdaten verschiedener Personen betrachten und versuchen, sie in Personen zu gruppieren, die ähnliche Merkmale aufweisen. Ich finde Astronomie und Weltraumforschung faszinierend. Eine Anwendung, die ich für sehr spannend hielt, war die Verwendung von Clustering für die Analyse astronomischer Daten durch Astronomen, um Körper im Weltraum zu gruppieren und so zu analysieren, was im Weltraum vor sich geht. Eine der Anwendungen, die ich faszinierend fand, war, dass Astronomen die Clusterbildung nutzten, um Körper zu gruppieren, um herauszufinden, welche eine Galaxie bilden oder welche kohärente Strukturen im Raum bilden. Clustering wird heute für all diese und viele weitere Anwendungen eingesetzt. Im nächsten Video werfen wir einen Blick auf den am häufigsten verwendeten Clustering-Algorithmus namens K-Means-Algorithmus und schauen uns an, wie er funktioniert.

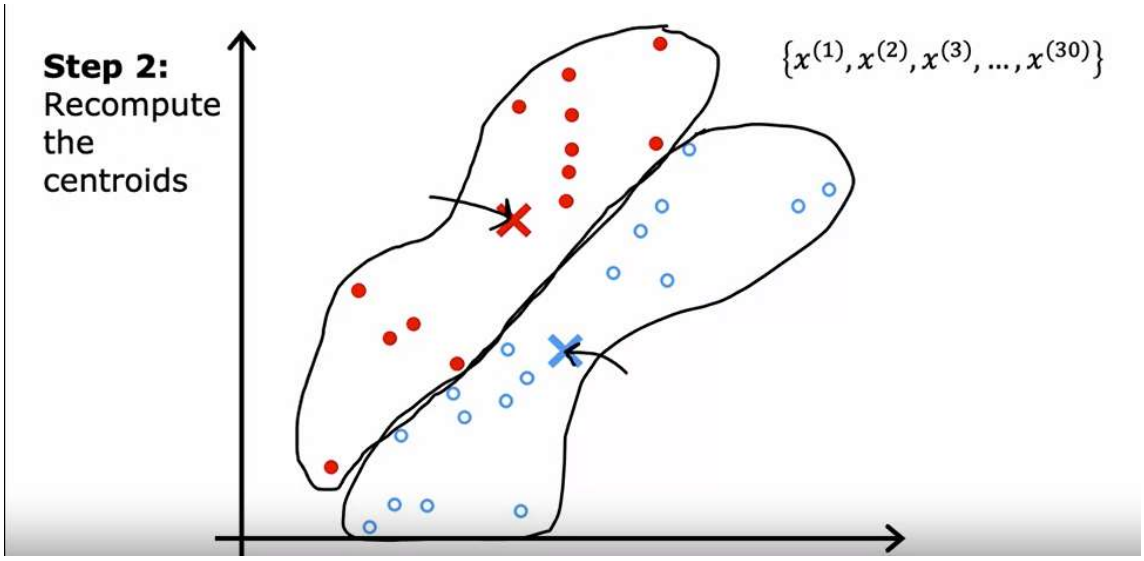
K-Means Intuition

Werfen wir einen Blick auf die Funktionsweise des K-Means-Clustering-Algorithmus. Lassen Sie mich mit einem Beispiel beginnen. Hier habe ich einen Datensatz mit 30 unbeschrifteten Trainingsbeispielen grafisch dargestellt. Es gibt also 30 Punkte. Und was wir gerne tun, ist, K-Means für diesen Datensatz auszuführen. Das erste, was der K-Means-Algorithmus tut, ist, dass er eine zufällige Vermutung anstellt, wo sich die Mittelpunkte der beiden Cluster befinden könnten, die Sie möglicherweise suchen. In diesem Beispiel werde ich es bitten, zu versuchen, zwei Cluster zu finden. Später in dieser Woche werden wir darüber sprechen, wie Sie entscheiden können, wie viele Cluster gefunden werden sollen. Aber der allererste Schritt besteht darin, dass er zufällig zwei Punkte auswählt, die ich hier als rotes und blaues Kreuz dargestellt habe, an denen sich die Mittelpunkte zweier verschiedener Cluster befinden könnten. Dies ist nur eine zufällige anfängliche Vermutung

und es handelt sich nicht um besonders gute Vermutungen. Aber es ist ein Anfang. Eine Sache, die Sie aus diesem Video hoffentlich mitnehmen können, ist, dass K-Means wiederholt zwei verschiedene Dinge tun wird. Die erste besteht darin, den Cluster-Schwerpunkten Punkte zuzuweisen, und die zweite darin, die Cluster-Schwerpunkte zu verschieben. Werfen wir einen Blick darauf, was das bedeutet. Der erste der beiden Schritte besteht darin, jeden dieser Punkte durchzugehen und zu prüfen, ob er näher am roten oder blauen Kreuz liegt.



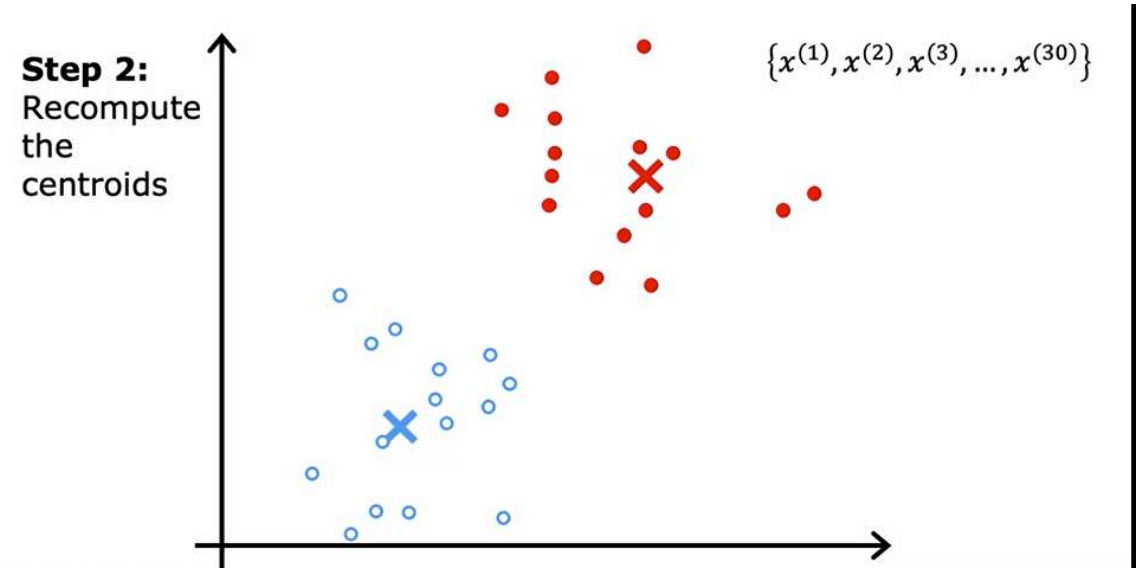
Das allererste, was K-means tut, ist, eine zufällige Schätzung darüber vorzunehmen, wo sich die Zentren des Clusters befinden. Und die Zentren des Clusters werden Clusterschwerpunkte genannt. Nachdem eine erste Vermutung angestellt wurde, wo sich der Clusterschwerpunkt befindet, werden alle diese Beispiele x(1) bis x(30), also meine 30 Datenpunkte, durchgegangen. Und für jeden von ihnen wird geprüft, ob er näher am roten Cluster-Schwerpunkt liegt, der durch das rote Kreuz angezeigt wird, oder ob er näher am blauen Cluster-Schwerpunkt liegt, der durch das blaue Kreuz angezeigt wird. Und es ordnet jeden dieser Punkte demjenigen der Clusterschwerpunkte zu, an dem er näher liegt. Ich werde das veranschaulichen, indem ich jedes dieser Beispiele, jeden dieser kleinen runden Punkte, entweder rot oder blau male, je nachdem, ob dieses Beispiel näher am roten oder am blauen Clusterschwerpunkt liegt. Dieser Punkt hier oben liegt also näher am roten Schwerpunkt, weshalb er rot eingefärbt ist. Wohingegen dieser Punkt dort unten näher am blauen Cluster-Schwerpunkt liegt, weshalb ich ihn jetzt blau gestrichen habe. Das war also das erste der beiden Dinge, die K-means immer wieder tut. Das ist ein Zeichen, das auf die Schwerpunkte von Clustern hinweist. Und das bedeutet lediglich, dass jeder Punkt eines der Clusterschwerpunkte mit der Farbe verknüpft wird, die ich hier veranschauliche. Der zweite der beiden Schritte, die K-means ausführt, besteht darin, alle roten Punkte zu betrachten und daraus einen Durchschnitt zu bilden. Und das rote Kreuz wird an die durchschnittliche Position der roten Punkte verschoben, die sich, wie sich herausstellt, hier befindet.



Und so wird sich das rote Kreuz, also der rote Cluster-Schwerpunkt, hierher bewegen. Und dann machen wir dasselbe für alle blauen Punkte. Schauen Sie sich alle blauen Punkte an, ermitteln Sie den Durchschnitt und verschieben Sie das blaue Kreuz dorthin. Damit haben Sie jetzt auch einen neuen Standort für den blauen Cluster-Schwerpunkt. Im nächsten Video schauen wir uns die mathematischen Formeln für die Durchführung dieser beiden Schritte an. Aber jetzt, da Sie diese neuen und hoffentlich leicht verbesserten Schätzungen für die Positionen der Schwerpunkte des Clusters haben, werden wir alle 30 Trainingsbeispiele noch einmal durchgehen. Und prüfen Sie für jeden von ihnen, ob er näher am roten oder blauen Clusterschwerpunkt der neuen Standorte liegt. Und dann werden wir sie, die durch die Farbe angezeigt werden, wieder mit jedem Punkt dem näheren Cluster-Schwerpunkt zuordnen. Und wenn Sie das tun, sehen Sie, dass die Feldpunkte ihre Farbe ändern. So wird dieser Punkt beispielsweise rot eingefärbt, da er zuvor näher am roten Clusterschwerpunkt lag. Aber wenn wir jetzt noch einmal hinsehen, ist es nun tatsächlich näher am blauen Cluster-Schwerpunkt, weil sich der blaue und rote Cluster-Schwerpunkt verschoben haben.

Wenn wir also jeden Punkt durchgehen und ihn mit den näheren Clusterschwerpunkten verknüpfen, erhalten Sie Folgendes. Und dann wiederholen wir einfach noch einmal den zweiten Teil von K-means. Schauen Sie sich alle roten Punkte an und berechnen Sie den Durchschnitt. Schauen Sie sich außerdem alle blauen Punkte an und berechnen Sie die durchschnittliche Position aller blauen Punkte. Und es stellt sich heraus, dass Sie am Ende das rote Kreuz dorthin und das blaue Kreuz hierher verschieben. Und wir wiederholen es. Schauen wir uns noch einmal alle Punkte an und färben sie entweder rot oder blau, je nachdem, an welchem Clusterschwerpunkt dieser näher liegt. Am Ende hast du also das hier. Schauen Sie sich dann noch einmal alle roten Punkte an und nehmen Sie deren durchschnittliche Position. Schauen Sie sich alle blauen Punkte an, nehmen Sie die

durchschnittliche Position und verschieben Sie die Cluster an die neuen Positionen.



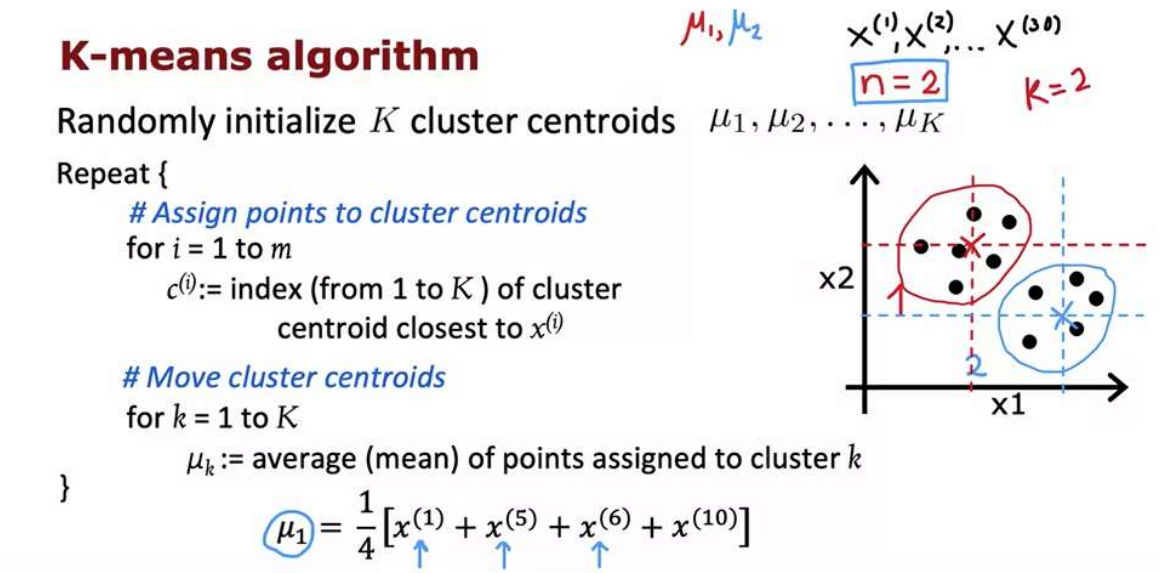
Und es stellt sich heraus, dass, wenn Sie diese beiden Schritte immer wieder wiederholen würden, Sie sich jeden Punkt ansehen und ihn dem nächstgelegenen Cluster-Schwerpunkt zuweisen und dann auch jeden Cluster-Schwerpunkt auf den Mittelwert aller Punkte mit derselben Farbe verschieben würden. Wenn Sie diese beiden Schritte weiterhin ausführen, stellen Sie fest, dass es keine weiteren Änderungen an den Farben der Punkte oder an den Positionen der Cluster-Schwerpunkte gibt. Das bedeutet also, dass der K-Means-Clustering-Algorithmus zu diesem Zeitpunkt konvergiert hat. Denn die wiederholte Anwendung dieser beiden Schritte führt zu keinen weiteren Änderungen an der Zuordnung, auf die Schwerpunkte zu zeigen, oder an der Position der Cluster-Schwerpunkte. In diesem Beispiel sieht es so aus, als hätte K-means ziemlich gute Arbeit geleistet. Es wurde festgestellt, dass diese Punkte hier oben einem Cluster entsprechen und diese Punkte hier unten einem zweiten Cluster entsprechen. Jetzt haben Sie eine Illustration gesehen, wie K-means funktioniert. Die beiden wichtigsten Schritte bestehen darin, jeden Punkt dem Clusterschwerpunkt zuzuordnen, je nachdem, welcher Clusterschwerpunkt am nächsten liegt. Und zweitens verschieben Sie jeden Clusterschwerpunkt auf den Durchschnitt oder Mittelwert aller ihm zugewiesenen Punkte. Im nächsten Video schauen wir uns an, wie man dies formalisiert, und schreiben den Algorithmus auf, der das tut, was Sie gerade in diesem Video gesehen haben. Kommen wir zum nächsten Video.

K-Means-Algorithmus

Im letzten Video haben Sie eine Darstellung des laufenden K-Means-Algorithmus gesehen. Schreiben wir nun den K-Means-Algorithmus im Detail auf, damit Sie ihn selbst implementieren können. Hier ist der K-Means-Algorithmus. Der erste Schritt besteht darin, die K-Clusterschwerpunkte Mu 1 Mu 2 bis Mu k zufällig zu initialisieren. In dem Beispiel, das wir hatten, entsprach dies der Zeit, als wir zufällig einen Ort für das rote Kreuz und für das blaue Kreuz wählten, die den beiden Clusterschwerpunkten entsprachen.

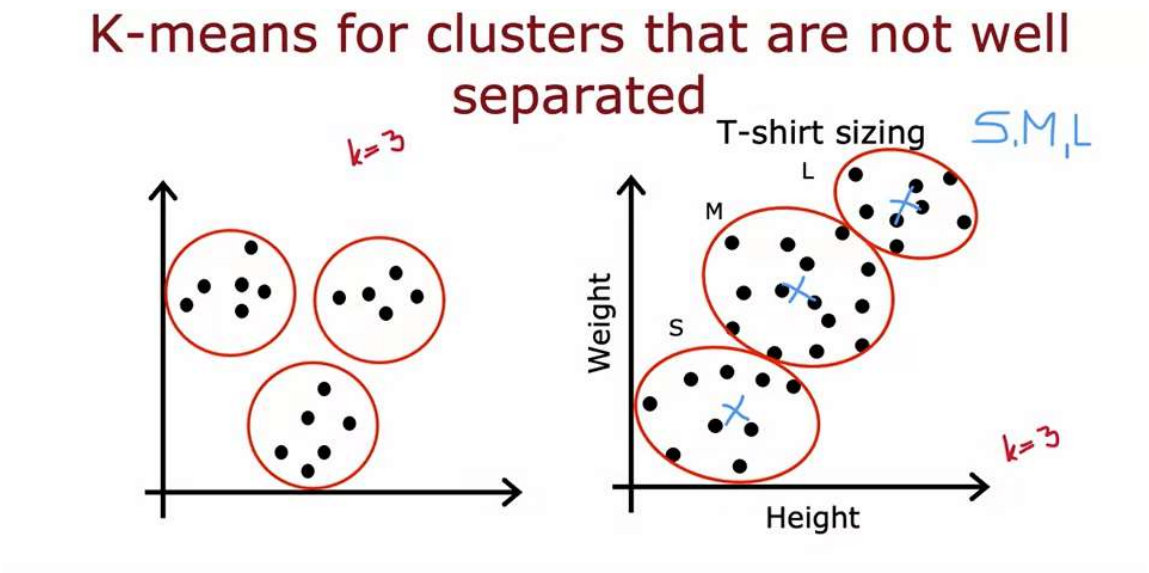
In unserem Beispiel war K gleich zwei. Wenn das rote Kreuz Cluster-Schwerpunkt eins und das blaue Kreuz Cluster-Schwerpunkt zwei wäre. Dies sind nur zwei Indizes zur Bezeichnung des ersten und zweiten Clusters. Dann wäre das rote Kreuz die Position von Mu 1 und das blaue Kreuz die Position von Mu 2. Um es klarzustellen: Mu 1 und Mu 2 sind *Vektoren*, die dieselbe Dimension wie Ihre Trainingsbeispiele haben, x1 bis x30. in unserem Beispiel. All diese sind Listen mit zwei Zahlen oder zweidimensionale Vektoren oder welche Dimension auch immer die Trainingsdaten hatten.

Wir hatten für jedes der Trainingsbeispiele n gleich zwei Merkmale, dann sind Mu 1 und Mu 2 ebenfalls zweidimensionale Vektoren, also Vektoren mit zwei Zahlen darin. Nachdem die K-Cluster-Schwerpunkte zufällig initialisiert wurden, führt K-means dann wiederholt die beiden Schritte aus, die Sie im letzten Video gesehen haben. Der erste Schritt besteht darin, Punkte Clustern, Schwerpunkten, d. h. Farbe, zuzuweisen, wobei jeder der Punkte entweder rot oder blau ist, was der Zuweisung zu Cluster-Schwerpunkten eins oder zwei entspricht, wenn K gleich zwei ist. Spülen Sie es ausreichend aus.



Das bedeutet, dass wir, da I für alle m Trainingsbeispiele gleich eins bis m ist, c^i gleich dem Index setzen werden, der alles von eins bis K des nächstgelegenen Clusterschwerpunkts sein kann zum Trainingsbeispiel x^i. Mathematisch kann man dies so darstellen, dass man den Abstand zwischen x^i und Mu k berechnet. In der Mathematik wird der Abstand zwischen zwei Punkten oft so geschrieben. Sie wird auch L2-Norm genannt. Was Sie suchen möchten, ist der Wert von k, der dies minimiert, da dieser dem Clusterschwerpunkt Mu k entspricht, der dem Trainingsbeispiel x^i am nächsten liegt. Dann wird der Wert von k, der dies minimiert, auf c^i gesetzt. Wenn Sie diesen Algorithmus implementieren, stellen Sie fest, dass es tatsächlich etwas praktischer ist, den quadratischen Abstand zu minimieren, da der Clusterschwerpunkt mit dem kleinsten quadratischen Abstand mit dem Clusterschwerpunkt mit dem kleinsten Abstand identisch sein sollte. Wenn Sie sich die optionalen Labore und Übungslabore dieser Woche ansehen, erfahren Sie, wie Sie dies selbst in Code implementieren können. Als konkretes Beispiel liegt dieser Punkt hier oben näher an den roten oder zwei Clusterschwerpunkten 1. Wenn dies das Trainingsbeispiel x^1 wäre, würden wir c^1 gleich 1 setzen. Wohingegen dieser Punkt hier drüben, wenn dies der Fall wäre Im 12. Trainingsbeispiel liegt dies näher am zweiten Cluster-Schwerpunkt, dem blauen. Wir werden diese, die entsprechende Clusterzuweisungsvariable, auf zwei setzen, da sie näher am Clusterschwerpunkt 2 liegt. Das ist der erste Schritt des K-Means-Algorithmus: Punkte den Clusterschwerpunkten zuweisen. Der zweite Schritt besteht darin, die Clusterschwerpunkte zu verschieben. Das bedeutet, dass Kleinbuchstabe k gleich 1 bis Großbuchstabe K ist, also die Anzahl der Cluster. Wir werden die zu aktualisierende Cluster-Schwerpunktposition auf den Durchschnitt oder den Mittelwert der diesem Cluster k zugewiesenen Punkte festlegen. Konkret bedeutet das, dass wir uns beispielsweise alle diese roten Punkte ansehen und ihre Position auf der horizontalen Achse sowie den Wert des ersten Merkmals x^1 betrachten und daraus den Durchschnitt bilden. Berechnen Sie auch den Durchschnittswert auf der vertikalen Achse. Nachdem Sie diese beiden Durchschnittswerte berechnet haben, stellen Sie fest, dass hier der Mittelwert liegt, weshalb Mu 1, also die Position, an der der Schwerpunkt des roten Clusters wie folgt aktualisiert wird. In ähnlicher Weise betrachten wir alle Punkte, die blau

gefärbt waren, d. h. mit c^i gleich 2, und berechnen den Durchschnitt des Werts auf der horizontalen Achse, den Durchschnitt ihres Merkmals x1. Berechnen Sie den Durchschnitt des Merkmals x2. Diese beiden Durchschnittswerte geben Ihnen die neue Position des blauen Clusterschwerpunkts, der sich daher hierher bewegt. Nur um das in Mathe aufzuschreiben. Wenn dem ersten Cluster Trainingsbeispiele 1,5,6,10 zugewiesen sind. Nur als Beispiel. Das bedeutet dann, dass Sie den Durchschnitt auf diese Weise berechnen. Beachten Sie, dass x^1, x^5, x^6 und x^10 Trainingsbeispiele sind. Vier Trainingsbeispiele, also dividieren wir durch 4 und das gibt Ihnen die neue Position von Mu1, dem neuen Clusterschwerpunkt für Cluster 1. Um es klar zu sagen: Jeder dieser x-Werte ist ein Vektor mit zwei Zahlen darin, oder n Zahlen darin, wenn Sie haben n Merkmale, und Mu enthält also auch zwei Zahlen oder n Zahlen, wenn Sie n Merkmale anstelle von zwei haben. Nun gibt es einen Eckfall dieses Algorithmus, der passiert, wenn einem Cluster keine Trainingsbeispiele zugewiesen sind. In diesem Fall würde der zweite Schritt, Mu k, darin bestehen, den Durchschnitt von Nullpunkten zu berechnen. Das ist nicht genau definiert. Sollte das jemals passieren, besteht die häufigste Vorgehensweise darin, diesen Cluster einfach zu beseitigen. Am Ende erhält man K minus 1 Cluster. Oder wenn Sie wirklich, wirklich K-Cluster benötigen, besteht eine Alternative darin, diesen Cluster-Schwerpunkt einfach zufällig neu zu initialisieren und zu hoffen, dass ihm beim nächsten Mal zumindest einige Punkte zugewiesen werden. Bei der Ausführung von K-Means ist es jedoch üblicher, einen Cluster einfach zu eliminieren, wenn ihm keine Punkte zugewiesen sind. Auch wenn ich hauptsächlich K-Mittel für gut getrennte Cluster beschrieben habe. Cluster, die so aussehen könnten. Wenn Sie sie bitten würden, drei Cluster zu finden, werden sie hoffentlich diese drei unterschiedlichen Cluster finden. Es stellt sich heraus, dass K-Means häufig auch auf Datensätze angewendet wird, bei denen die Cluster nicht so gut getrennt sind. Wenn Sie beispielsweise Designer und Hersteller von coolen T-Shirts sind und entscheiden möchten, wie ich die Größe meiner kleinen, mittleren und großen T- Shirts festlegen möchte. Wie klein sollte ein kleines T-Shirt sein, wie groß sollte ein großes T-Shirt sein und was sollte ein mittelgroßes T-Shirt wirklich sein? Sie könnten beispielsweise Daten von Personen sammeln, die Ihre T-Shirts aufgrund ihrer Größe und ihres Gewichts kaufen würden. Sie stellen fest, dass Größe und Gewicht von Menschen im Spektrum tendenziell kontinuierlich variieren, ohne dass es klare Cluster gibt. Wenn Sie jedoch K-means beispielsweise mit drei Cluster-Schwerpunkten ausführen würden, könnten Sie feststellen, dass K-means diese Punkte in einem Cluster, diese Punkte in einem zweiten Cluster und diese Punkte in einem dritten Cluster gruppieren würde.



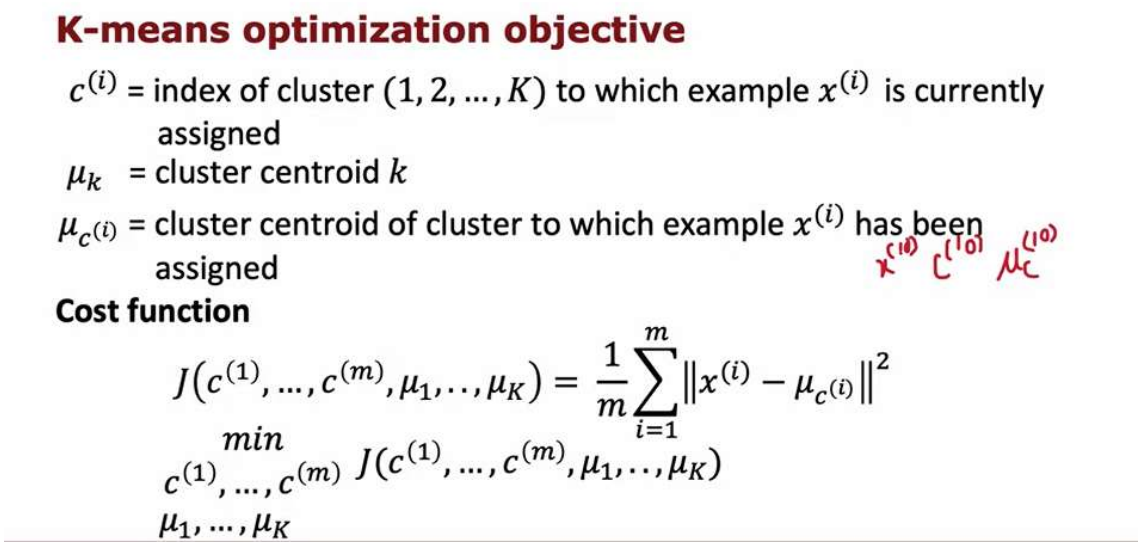
Wenn Sie sich für die genaue Größe Ihrer kleinen, mittleren und großen T-Shirts entscheiden möchten, können Sie die Größe Ihres kleinen T-Shirts so wählen, dass es diesen Personen gut passt.

Das mittelgroße T-Shirt soll versuchen, diesen Personen gut zu passen, und das große T-Shirt soll diesen Personen gut passen, wobei möglicherweise die Cluster-Schwerpunkte Ihnen einen Eindruck davon vermitteln, welche Größe und welches Gewicht am repräsentativsten sind, die Sie sich wünschen Passen Sie Ihre drei T-Shirt-Größen an. Dies ist ein Beispiel dafür, dass K-Means einwandfrei funktioniert und nützliche Ergebnisse liefert, selbst wenn die Daten nicht in gut getrennten Gruppen oder Clustern liegen. Das war der K-Means-Clustering-Algorithmus. Weisen Sie Cluster-Schwerpunkte nach dem Zufallsprinzip zu und weisen Sie den Cluster-Schwerpunkten dann wiederholt Punkte zu und verschieben Sie die Cluster-Schwerpunkte. Aber was macht dieser Algorithmus wirklich und glauben wir, dass dieser Algorithmus konvergieren wird, oder läuft er einfach ewig weiter und konvergiert nie? Um einen tieferen Einblick in den K-Means-Algorithmus zu gewinnen und auch zu verstehen, warum wir hoffen, dass dieser Algorithmus konvergiert, gehen wir zum nächsten Video über, in dem Sie sehen, dass K-Means tatsächlich versucht, eine bestimmte Kostenfunktion zu optimieren. Schauen wir uns das im nächsten Video an.

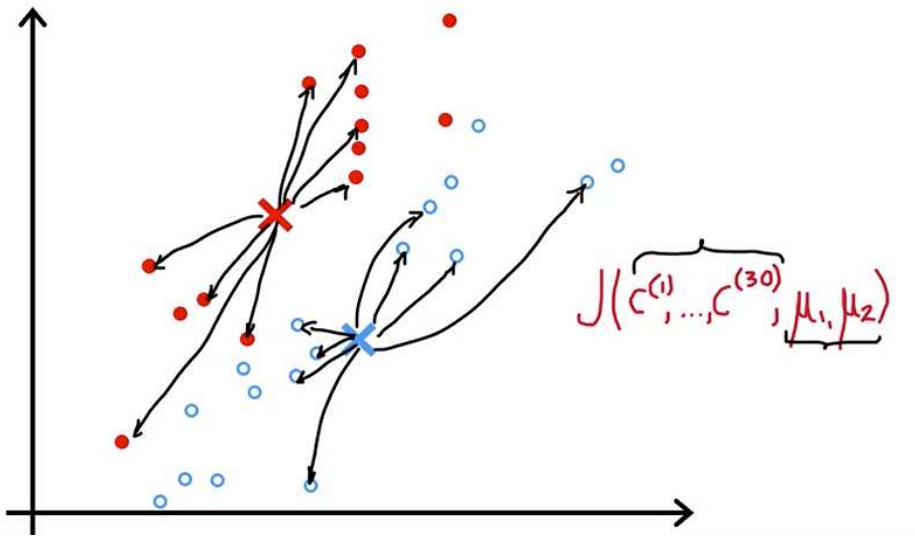
Optimierungsziel

In den früheren Kursen, den Kursen eins und zwei der Spezialisierung, sahen Sie viele überwachte Lernalgorithmen als Trainingssätze, die eine Kostenfunktion darstellen. Und dann verwenden wir Grading Descent oder andere Algorithmen, um diese Kostenfunktion zu optimieren. Es stellt sich heraus, dass der Schlüsselmittelalgorithmus, den Sie im letzten Video gesehen haben, auch die Optimierung einer bestimmten Kostenfunktion ist. Obwohl der Optimierungsalgorithmus, den es zur Optimierung verwendet, keinen Widerspruch hervorruft, ist es tatsächlich der Algorithmus, den Sie bereits im letzten Video gesehen haben. Werfen wir einen Blick darauf, was das alles bedeutet. Werfen wir einen Blick auf die Kostenfunktion für einen Mittelwert. Zur Erinnerung: Dies ist eine Notation, die wir verwendet haben, während CI der Index des Clusters ist. CI ist also eine Zahl aus einem Su K des Indexes des Clusters, dem Trainingsbeispiel XI aktuell zugeordnet ist, und new K ist die Position des Clusterschwerpunkts k. Lassen Sie mich noch einen weiteren Teil der Notation vorstellen, nämlich den Fall, dass das Kleinbuchstabe K gleich CI ist. Mu-Index CI ist also der Clusterschwerpunkt des Clusters, dem Beispiel XI zugeordnet wurde. Wenn ich mir zum Beispiel ein Trainingsbeispiel C, Trainingsbeispiel 10, ansehe und frage: Wo befindet sich das konstante Jahrhundert, dem das 10. Trainingsbeispiel zugeordnet wurde? Nun, ich würde dann nach C10 suchen. Dadurch erhalte ich eine Zahl von eins bis K. Das sagt mir, dass Beispiel 10 dem roten oder blauen oder einem anderen Cluster-Schwerpunkt zugewiesen wurde, und dann ist mu-Index C-10 die Position des Cluster-Schwerpunkts, dem das Ausmaß zugewiesen wurde. Mit dieser Notation ausgestattet, möchte ich nun die Kostenfunktion aufschreiben, die K bedeutet, dass sie sich als minimierend erweist. Die Kostenfunktion J, die eine Funktion von C1 bis CM ist. Dies sind alle Zuordnungen von Punkten zu Android-Clustern sowie neue durch Mu-Kapsel K. Dies sind die Positionen aller Cluster-Schwerpunkte, die als dieser Ausdruck auf der rechten Seite definiert sind. Es ist der Durchschnitt, also eins über M einige von Michaels eins zu ihm des quadratischen Abstands zwischen jedem Trainingsbeispiel XI, während ich von eins bis M geht, ist es ein quadratischer Abstand zwischen XI. Und Nu-Index C hoch. Diese Größe hier oben, mit anderen Worten, die für Kenianer gute Kostenfunktion ist der durchschnittliche quadratische Abstand zwischen jedem Trainingsbeispiel XI. Und die Position des Clusterschwerpunkts, dem das Trainingsbeispiel-Exil zugewiesen wurde. Für dieses Beispiel hier oben haben wir also den Abstand zwischen X10 und dem Index C10 gemessen. Der Cluster-Schwerpunkt, in dem das Ausmaß zugewiesen wurde, nimmt das Quadrat dieser Entfernung und das wäre hier einer der Begriffe, über die wir den Mittelwert bilden. Und es stellt sich heraus, dass das K-Mittel-Album versucht, Zuordnungen von Punkten von Cluster-Schwerpunkten sowie Positionen von Cluster-Schwerpunkten zu finden, die den quadrierten Abstand minimieren. Visuell sehen Sie hier, was Sie mitten in der K-Bedeutung im früheren Video gesehen haben. Und in diesem Schritt die Kostenfunktion. Wenn Sie einen Computer verwenden würden,

müssten Sie jeden an den blauen Punkten betrachten und diese Abstände und das Computerquadrat messen. Und dann schauen Sie sich auf ähnliche Weise jeden einzelnen der roten Punkte an, berechnen Sie diese Abstände und berechnen Sie das Quadrat. Und dann ist der Durchschnitt der Quadrate aller dieser Differenzen für die roten und blauen Punkte der Wert der Kostenfunktion J bei dieser speziellen Konfiguration der Parameter für Könige. Und sie werden bei jedem Schritt versuchen, die Clusterzuweisungen C1 bis C30 in diesem Beispiel zu aktualisieren. Oder aktualisieren Sie die Positionen des Cluster-Zentralismus, U1 und U2. Um diese Kostenfunktion J immer weiter zu reduzieren.



Diese Kostenfunktion J hat in der Literatur übrigens auch einen Namen, der als Verzerrungsfunktion bezeichnet wird. Ich weiß nicht, ob das ein toller Name ist. Aber wenn Sie jemanden über den Schlüsselnachrichtenalgorithmus und die Verzerrung oder die Verzerrungskostenfunktion sprechen hören, ist es genau das, was diese Formel J berechnet. Werfen wir nun einen genaueren Blick auf den Algorithmus und warum der Algorithmus versucht, diese Kostenfunktion J zu minimieren. Oder warum versucht er, die Verzerrung hier zusätzlich zu der von der vorherigen Folie kopierten Kostenfunktion zu minimieren. Es stellt sich heraus, dass der erste Teil von K bedeutet, wo Sie dem Clusterschwerpunkt Punkte zuweisen. Es stellt sich heraus, dass versucht wird, C1 über CM zu aktualisieren. Versuchen Sie, die Kostenfunktion J so weit wie möglich zu minimieren und gleichzeitig eine neue durch mu K zu fixieren. Und im zweiten Schritt, im Gegensatz dazu, wo Sie den benutzerdefinierten Schwerpunkt verschieben, stellt sich heraus, dass versucht wird, C1 durch CM-Fix zu verlassen. Aber um eine neue durch mu K zu aktualisieren, versuchen Sie, die Kostenfunktion oder die Verzerrung so weit wie möglich zu minimieren. Werfen wir einen Blick darauf, warum das so ist. Wenn Sie im ersten Schritt die Werte von C1 bis CM auswählen oder einen bestimmten Wert von Ci speichern möchten, versuchen Sie, diesen zu minimieren. Nun, was würde Xi minus mu CI so klein wie möglich machen? Dies ist der Abstand bzw. die Quadratdistanz zwischen einem Trainingsbeispiel XI. Und der Ort der Klasse ist von zentraler Bedeutung, der zugewiesen wurde.



Wenn Sie also diesen Abstand bzw. den Quadratabstand minimieren möchten, sollten Sie XI dem nächstgelegenen Heiligtum zuweisen. Um ein vereinfachtes Beispiel zu nehmen: Wenn Sie zwei Cluster-Schwerpunkte haben, sagen wir, in der Nähe des Mittelpunkts liegen eins und zwei und nur ein einziges Trainingsbeispiel, XI. Wenn Sie es als Cluster in der Mitte bezeichnen würden, wäre dieser quadratische Abstand hier dieser große Abstand, gut quadriert. Und wenn Sie es zu nah an der Mitte unterzeichnen würden, wäre dieser Quadratabstand das Quadrat dieses viel kleineren Abstands. Wenn Sie diesen Begriff also minimieren möchten, nehmen Sie XI und weisen ihn dem näheren Schwerpunkt zu, was genau das ist, was das Album hier oben tut. Aus diesem Grund besteht der Schritt, bei dem Sie Punkte für das Clusterjahrhundert signieren, darin, die Werte für CI auszuwählen, um zu versuchen, J zu minimieren.

K-means optimization objective

$c^{(i)}$ = index of cluster $(1, 2, \ldots, K)$ to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

$x^{(10)}$ $c^{(10)}$ $\mu_c^{(10)}$

Cost function

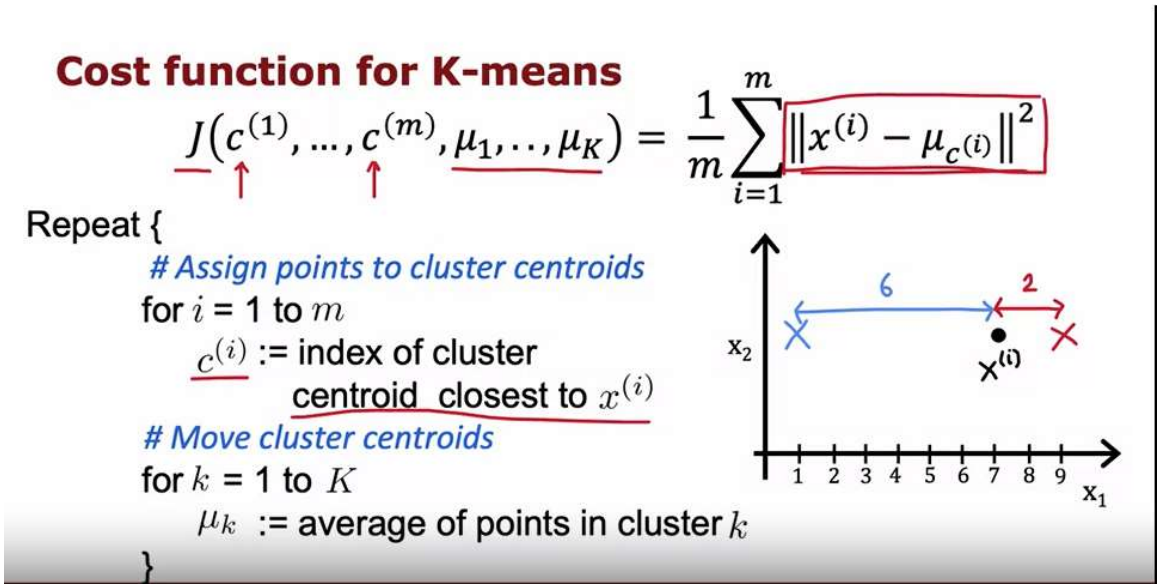
$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, .., \mu_K) = \frac{1}{m}\sum_{i=1}^{m} \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

$$\min_{\substack{c^{(1)}, \ldots, c^{(m)} \\ \mu_1, \ldots, \mu_K}} J(c^{(1)}, \ldots, c^{(m)}, \mu_1, .., \mu_K)$$

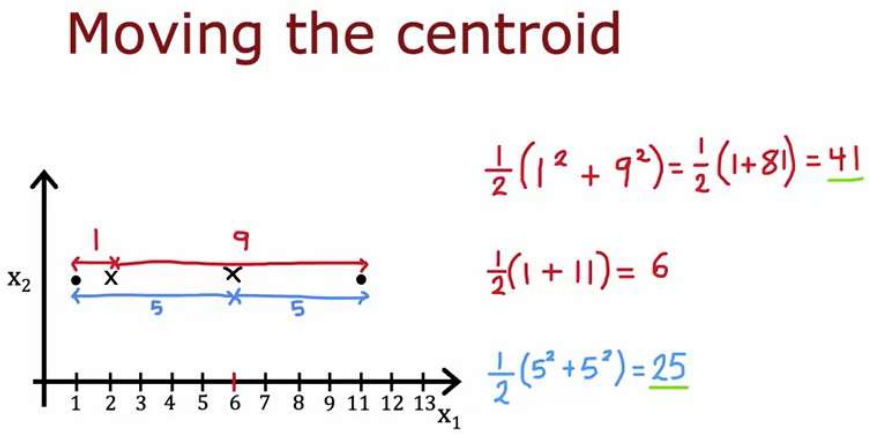
Distortion

Ohne Änderungen sind wir vorerst durch Großbritannien gegangen, haben aber nur die Werte von C1 bis CM ausgewählt, um zu versuchen, diese Terme zu erstellen so klein wie möglich. Wie wäre es mit dem zweiten Schritt des K-Means-Algorithmus, der darin besteht, zu Clusterschwerpunkten zu wechseln? Es stellt sich heraus, dass die Wahl von mu K als Durchschnitt und Mittelwert der zugewiesenen Punkte die Wahl dieser Terme mu ist, die diesen Ausdruck minimiert. Nehmen wir ein vereinfachtes Beispiel: Angenommen, Sie haben einen Cluster mit nur zwei Punkten daneben, wie im Folgenden dargestellt. Wenn also hier der nächstgelegene Schwerpunkt vorliegt, wäre der Durchschnitt der quadratischen Abstände ein Abstand von hier eins zum Quadrat plus dieser Abstand

hier, der 9 zum Quadrat beträgt. Und dann nehmen Sie den Durchschnitt dieser beiden Zahlen. Das ergibt also die Hälfte von 1 plus 81, was 41 ergibt. Aber wenn Sie den Durchschnitt dieser beiden Punkte nehmen würden, also 1 + 11/2, wäre das gleich 6. Und wenn ja Um den Schwerpunkt des Clusters hier in die Mitte zu verschieben, als der Durchschnitt dieser beiden quadratischen Abstände beträgt, ergibt sich eine Entfernung von fünf und fünf Jahren.



Am Ende erhalten Sie also die Hälfte von 5 zum Quadrat plus 5 zum Quadrat, was 25 entspricht. Und das ist ein viel kleinerer durchschnittlicher Quadratabstand als 41. Und tatsächlich können Sie mit der Position dieses Clusterschwerpunkts herumspielen und vielleicht überzeugen Sie selbst, dass Sie diesen gemeinen Standort einnehmen. Diese durchschnittliche Position in der Mitte dieser beiden Trainingsbeispiele ist tatsächlich der Wert, der den Quadratabstand minimiert. Die Tatsache, dass der K-Means-Algorithmus eine Kostenfunktion J optimiert, bedeutet also, dass die Konvergenz garantiert ist, und zwar bei jeder einzelnen Iteration.

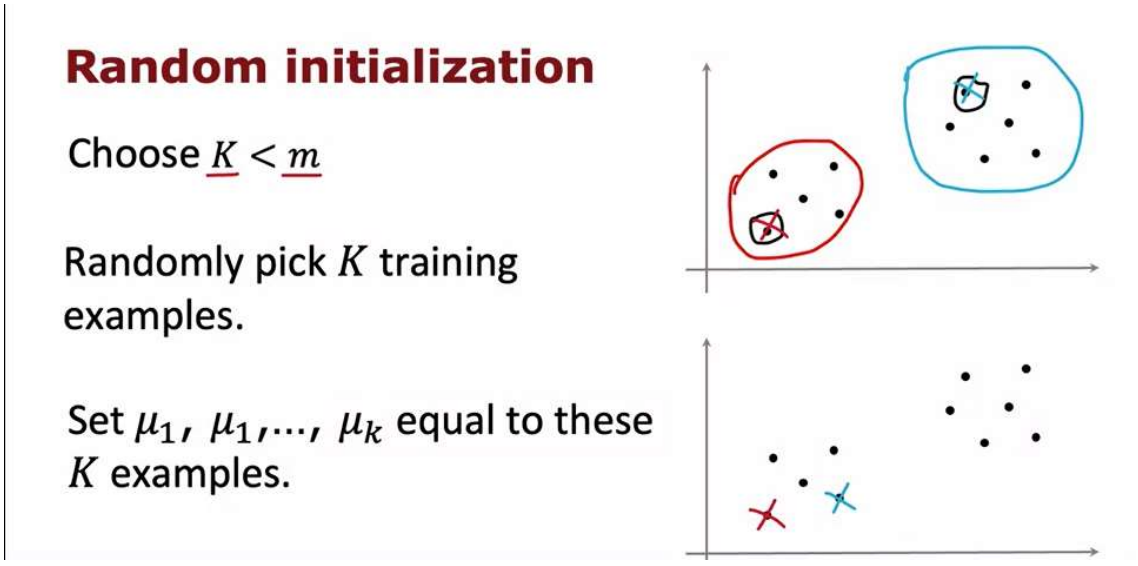


Die Verzerrungskostenfunktion sollte sinken oder gleich bleiben, aber wenn sie jemals nicht sinkt oder gleich bleibt, im schlimmsten Fall, wenn sie jemals steigt. Das bedeutet, dass es einen Fehler im Code gibt, er sollte niemals nach oben gehen, da jeder einzelne Schritt von K bedeutet, dass der Wert CI und mu K festgelegt werden, um zu versuchen, die Kostenfunktion zu reduzieren. Auch wenn die Kostenfunktion jemals nicht mehr sinkt, haben Sie damit auch eine Möglichkeit zu testen, ob die K-

Mittel konvergiert haben. Sobald es eine einzige Dauer gibt, bleibt es gleich. Das bedeutet normalerweise, dass die K-Mittel konvergiert sind und Sie einfach aufhören sollten, das Album noch weiter laufen zu lassen, oder in einigen seltenen Fällen werden Sie die K-Mittel über einen längeren Zeitraum laufen lassen. Und die Kostenfunktion der Verzerrung nimmt nur sehr, sehr langsam ab, und das ist ein bisschen wie bei einem großartigen Inter-Send, wo es vielleicht etwas helfen könnte, noch länger zu laufen. Aber wenn die Geschwindigkeit, mit der die Kostenfunktion sinkt, sehr, sehr langsam geworden ist. Man könnte auch einfach sagen, das ist gut genug. Ich möchte nur sagen, dass es nahe genug an der Konvergenz ist und nicht noch mehr Rechenzyklen benötigt, um das Album noch länger laufen zu lassen. Dies sind also einige der Möglichkeiten, wie die Berechnung der Kostenfunktion hilfreich ist, um herauszufinden, ob das Album konvergiert hat. Es stellt sich heraus, dass es noch eine weitere sehr nützliche Möglichkeit gibt, die Kostenfunktion zu nutzen, nämlich die Verwendung mehrerer verschiedener zufälliger Initialisierungen des Clusterschwerpunkts. Es stellt sich heraus, dass Sie mit K-Mitteln oft viel bessere Cluster finden können, wenn Sie dies tun. Schauen wir uns das nächste Video an, wie das geht.

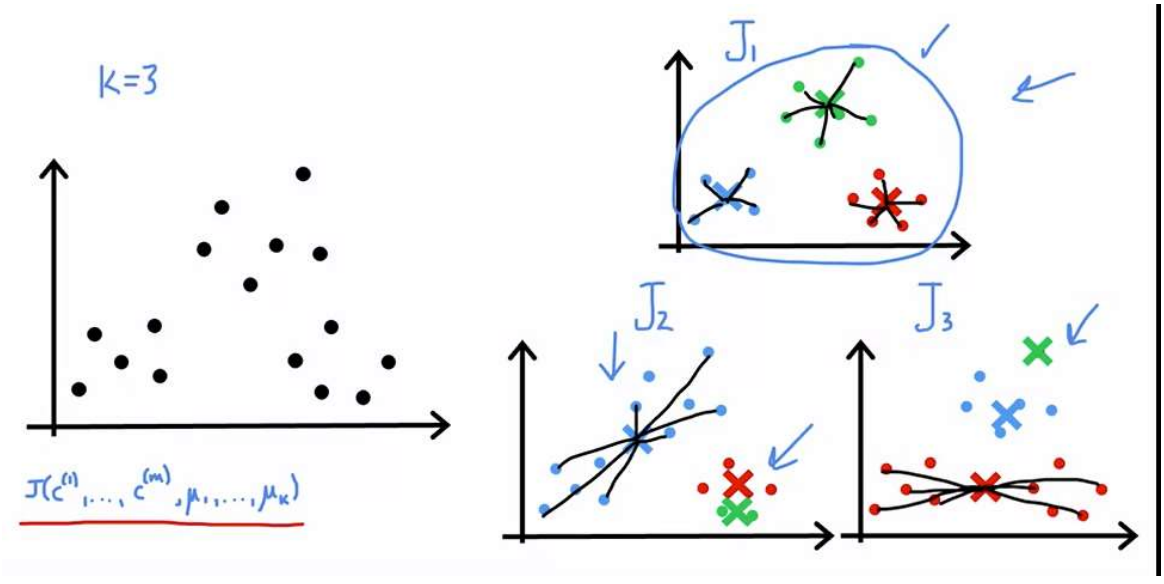
K-Mittel werden initialisiert

Der allererste Schritt des K-Means-Clustering-Algorithmus bestand darin, zufällige Orte als anfängliche Schätzungen für die Clusterschwerpunkte mu eins bis mu K auszuwählen. Aber wie nimmt man diese zufällige Schätzung eigentlich an? Schauen wir uns das in diesem Video an und erfahren Sie auch, wie Sie mehrere Versuche mit den anfänglichen Schätzungen von mu eins bis mu K unternehmen können. Das wird dazu führen, dass Sie eine bessere Gruppe von Clustern finden. Werfen wir einen Blick darauf, hier ist noch einmal der K-Mittelalgorithmus und in diesem Video schauen wir uns an, wie Sie diesen ersten Schritt umsetzen können. Wenn Sie K ausführen, sollten Sie so gut wie immer die Anzahl der K der Cluster-Zentrale wählen, die auf Trainingsbeispiele m reduziert werden soll. Es macht eigentlich keinen Sinn, K größer als m zu haben, denn dann gibt es nicht einmal genug Trainingsbeispiele, um mindestens ein Trainingsbeispiel pro Clusterschwerpunkt zu haben. In unserem früheren Beispiel hatten wir also K gleich zwei und m gleich 30.



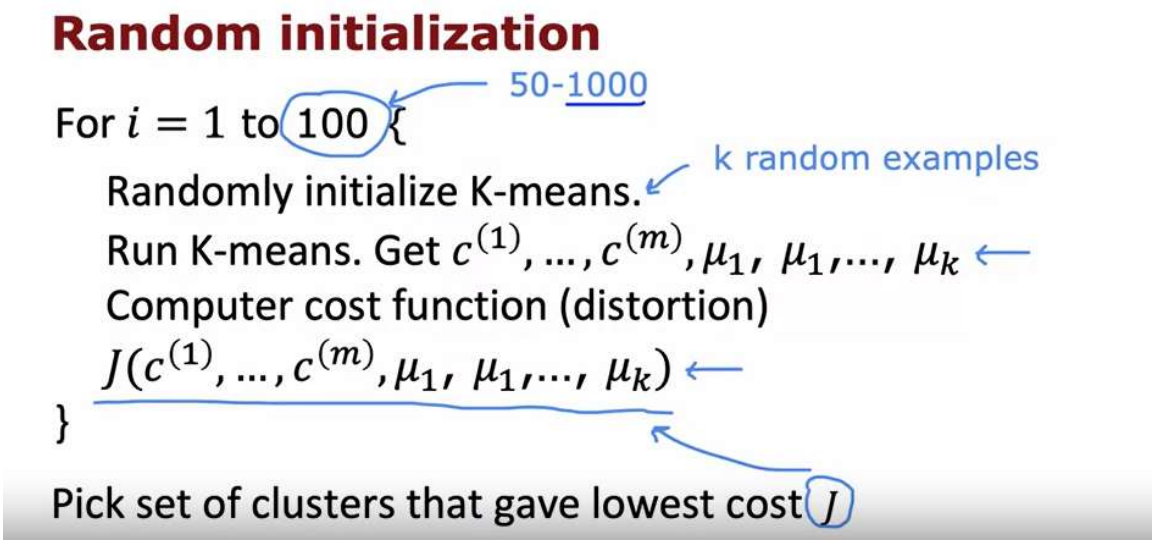
Um die Clusterschwerpunkte auszuwählen, besteht die häufigste Methode darin, zufällig K Trainingsbeispiele auszuwählen. Hier ist also ein Trainingssatz, bei dem ich, wenn ich zufällig zwei Trainingsbeispiele auswählen würde, am Ende vielleicht dieses und dieses auswählen würde. Und dann würden wir ein neues durch mu K gleich diesen K Trainingsbeispielen setzen. Ich könnte also

meinen roten Cluster-Schwerpunkt hier initialisieren und meinen blauen Cluster initialisieren, der hierher geschickt wurde, in dem Beispiel, in dem K gleich zwei war. Und es stellt sich heraus, dass, wenn dies Ihre zufällige Initialisierung war und Sie K ausführen würden, Sie beten würden, dass K am Ende die Entscheidung bedeutet, dass dies die beiden Klassen im Datensatz sind. Beachten Sie, dass sich diese Methode zur Initialisierung der Zentralkosten ein wenig von der unterscheidet, die ich in der Abbildung in den früheren Videos verwendet habe. Dabei habe ich die Clusterschwerpunkte mu eins und mu zwei so initialisiert, dass sie nur zufällige Punkte sind, anstatt auf bestimmten Trainingsbeispielen zu sitzen. Ich habe das getan, um die Abbildungen in den früheren Videos klarer zu machen. Aber was ich auf dieser Folie zeige, ist tatsächlich eine viel häufiger verwendete Methode zur Initialisierung der Cluster-Schwerpunkte. Bei dieser Methode besteht nun die Möglichkeit, dass Sie am Ende eine Initialisierung der Clusterschwerpunkte erhalten, bei denen sich hier das rote Kreuz und möglicherweise hier das blaue Kreuz befindet. Und je nachdem, wie Sie die zufälligen anfänglichen zentralen Schwerpunkte auswählen, wählt Kineys \g raw am Ende einen unterschiedlichen Satz von Ursachen für Ihren Datensatz aus. Schauen wir uns ein etwas komplexeres Beispiel an, in dem wir uns diesen Datensatz ansehen und versuchen, drei Cluster zu finden, sodass k in diesen Daten gleich drei ist. Wenn Sie K-Mittel mit einer zufälligen Initialisierung des Clusterschwerpunkts ausführen würden, erhalten Sie möglicherweise dieses Ergebnis hier und dies scheint eine ziemlich gute Wahl zu sein. Ziemlich gute Gruppierung der Daten in drei verschiedene Cluster. Aber mit einer anderen Initialisierung, sagen wir, Sie hätten zufällig zwei der Clusterschwerpunkte innerhalb dieser Punktgruppe initialisiert. Und einer innerhalb dieser Gruppe von Punkten bedeutet nach dem Ausführen von k, dass es möglicherweise zu dieser Clusterbildung kommt, die nicht so gut aussieht. Und es stellt sich heraus, dass es sich um ein lokales Optimum handelt, bei dem K-means versucht, die Verzerrungskostenfunktion zu minimieren, also die Kostenfunktion J von C eins bis CM und mu eins bis mu K, die Sie im letzten Video gesehen haben. Aber mit dieser weniger glücklichen Wahl der zufälligen Initialisierung war es zufällig in einem lokalen Minimum stecken geblieben.



Und hier ist ein weiteres Beispiel für ein lokales Minimum, bei dem ein anderer Zufallsinitialisierungskurs eingesetzt wurde, um diese Clusterung der Daten in drei Cluster zu finden, was wiederum nicht so gut zu sein scheint wie der, den Sie hier oben gesehen haben. Wenn Sie also k angeben möchten, bedeutet dies, dass Sie mehrere Versuche unternehmen müssen, um das beste lokale Optimum zu finden. Wenn Sie mehrere zufällige Initialisierungen ausprobieren möchten,

geben Sie ihm eine bessere Chance, dieses gute Clustering oben zu finden. Eine andere Möglichkeit, den Kings-Algorithmus zu nutzen, besteht darin, ihn mehrmals auszuführen und dann zu versuchen, die besten lokalen Optima zu finden. Und es stellt sich heraus, dass wenn Sie k ausführen würden, sagen wir dreimal, und am Ende diese drei unterschiedlichen Clusterings erhalten würden. Dann besteht eine Möglichkeit, zwischen diesen drei Lösungen zu wählen, darin, die Kostenfunktion J für alle drei dieser Lösungen zu berechnen, wobei alle drei dieser Auswahlmöglichkeiten von Clustern durch k Mittelwerte gefunden werden. Und dann wählen Sie eine dieser drei aus, je nachdem, welche von ihnen Ihnen den niedrigsten Wert für die Kostenfunktion J liefert. Und tatsächlich, wenn Sie sich diese Gruppierung von Clustern hier oben ansehen, hat dieses grüne Kreuz insgesamt relativ kleine quadratische Abstände die grünen Punkte. Das rote Kreuz hat einen relativ kleinen Abstand und rote Punkte und ebenso das blaue Kreuz. Daher wird die Kostenfunktion J für dieses Beispiel obendrein relativ klein sein. Allerdings hat das blaue Kreuz hier größere Abstände zu allen blauen Punkten. Und hier hat das rote Kreuz größere Abstände zu allen roten Punkten, weshalb die Kostenfunktion J für diese Beispiele unten größer wäre. Aus diesem Grund wählen Sie aus diesen drei Optionen diejenige mit der geringsten Verzerrung der kleinsten Kostenfunktion J aus. Am Ende wählen Sie diese Wahl der Clusterschwerpunkte. Lassen Sie mich dies also formeller in einen Algorithmus umwandeln, und ich wünschte, Sie würden die Rangfolge mehrmals unter Verwendung unterschiedlicher zufälliger Initialisierungen festlegen.

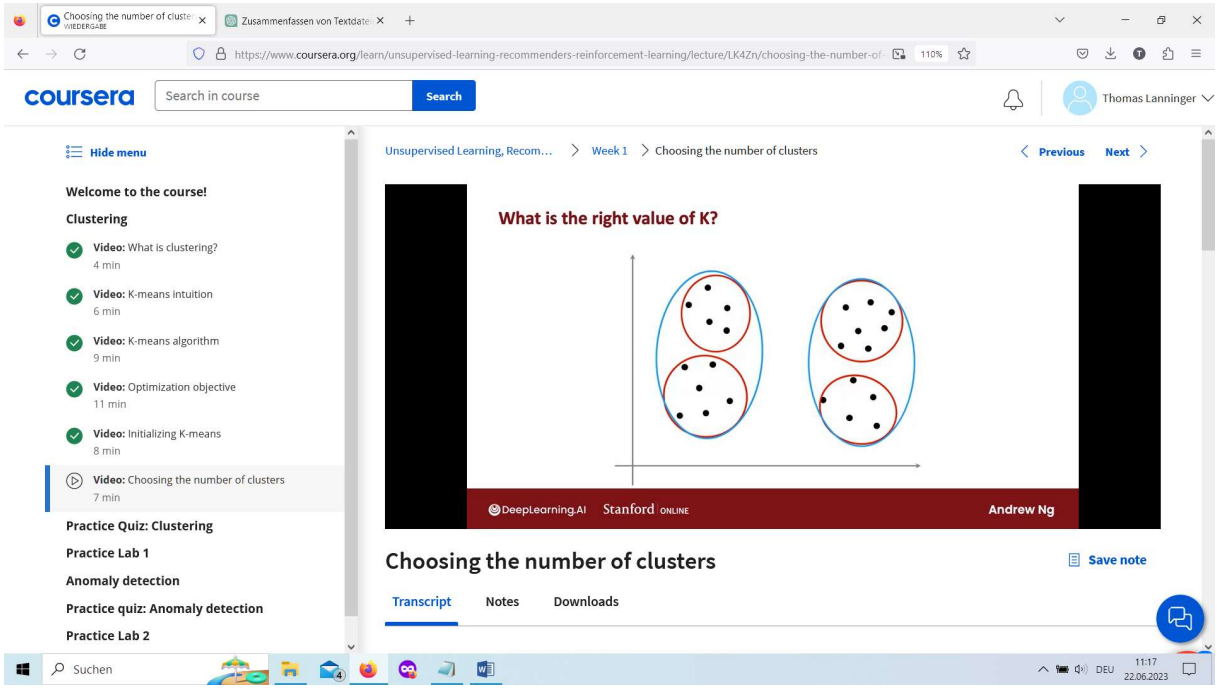


Hier ist der Algorithmus: Wenn Sie 100 zufällige Initialisierungen für K-Means verwenden möchten, würden Sie 100 Mal zufällig initialisierte K-Means mit der Methode ausführen, die Sie zuvor in diesem Video gesehen haben. Wählen Sie K-Trainingsbeispiele aus und lassen Sie die Cluster-Jahrhunderte zunächst die Orte dieser K-Trainingsbeispiele sein. Führen Sie mithilfe dieser zufälligen Initialisierung den K-Means-Algorithmus zur Konvergenz aus. Dadurch haben Sie die Wahl zwischen Clusterzuweisungen und Clusterschwerpunkten. Und schließlich würden Sie die Verzerrung und die Kostenfunktion wie folgt berechnen. Nachdem Sie dies beispielsweise 100 Mal durchgeführt haben, würden Sie schließlich die Gruppe von Clustern auswählen, die die niedrigsten Kosten verursacht. Und es stellt sich heraus, dass Sie auf diese Weise oft einen viel besseren Clustersatz mit einer viel geringeren Verzerrungsfunktion erhalten, als wenn Sie K nur einmal ausführen würden. Ich habe die Zahl hier oben als 100 eingegeben. Wenn ich diese Methode verwende, wäre es ziemlich üblich, dies etwa 50 bis 1000 Mal zu tun. Wenn Sie diesen Vorgang viel mehr als 1000 Mal ausführen, wird er tendenziell rechenintensiv. Und die Rendite sinkt tendenziell, wenn man es oft ausführt. Wenn Sie jedoch mindestens 50 oder 100 zufällige Initialisierungen ausprobieren, erhalten Sie oft ein viel besseres Ergebnis, als wenn Sie nur eine Chance hätten, eine gute zufällige Initialisierung

auszuwählen. Aber mit dieser Technik ist es viel wahrscheinlicher, dass Sie am Ende diese gute Auswahl an Clustern an der Spitze haben. Und diese weniger überlegenen lokalen Minima unten. Wenn ich selbst den K-Means-Algorithmus verwende, verwende ich fast immer mehr als eine zufällige Initialisierung. Da es nur dazu führt, dass K die Verzerrungskostenfunktion viel besser minimiert und eine viel bessere Wahl für die Kosten der Schwerpunkte findet. Bevor wir unsere Diskussion über K-Mittel abschließen, gibt es noch ein weiteres Video, das ich hoffentlich mit Ihnen besprechen kann. Die Frage: Wie wählt man die Anzahl der Cluster-Schwerpunkte? Wie wählt man den Wert von K? Schauen wir uns das im nächsten Video an.

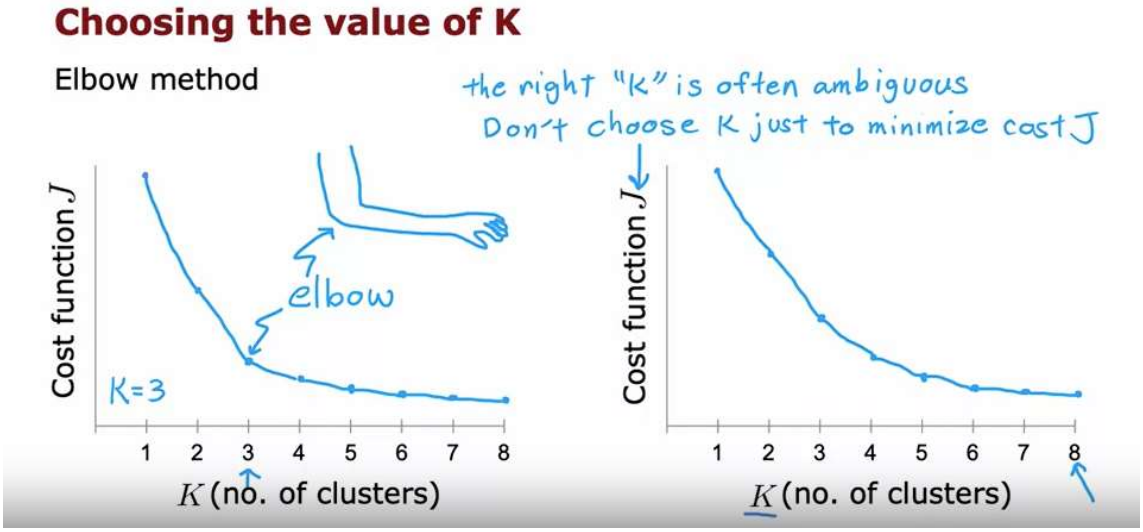
Auswahl der Anzahl der Cluster

Der k-means-Algorithmus benötigt als eine seiner Eingaben, k, die Anzahl der Cluster, die er finden soll, aber wie entscheiden Sie, wie viele Cluster verwendet werden sollen? Möchten Sie zwei Cluster oder drei Cluster mit fünf Clustern oder 10 Clustern? Lass uns einen Blick darauf werfen. Bei vielen Clusterproblemen ist der richtige Wert von K wirklich nicht eindeutig. Wenn ich verschiedenen Personen denselben Datensatz zeigen würde und fragen würde, wie viele Cluster sehen Sie? Es wird definitiv Leute geben, die sagen, es sieht so aus, als gäbe es zwei unterschiedliche Cluster, und sie werden Recht haben. Es gibt auch andere, die tatsächlich vier verschiedene Cluster sehen. Sie hätten auch Recht. Da es sich bei Clustering um einen unbeaufsichtigten Lernalgorithmus handelt, erhalten Sie nicht die richtigen Antworten in Form spezifischer Bezeichnungen, die Sie replizieren können. Es gibt viele Anwendungen, bei denen die Daten selbst keinen klaren Hinweis darauf geben, wie viele Cluster darin enthalten sind.

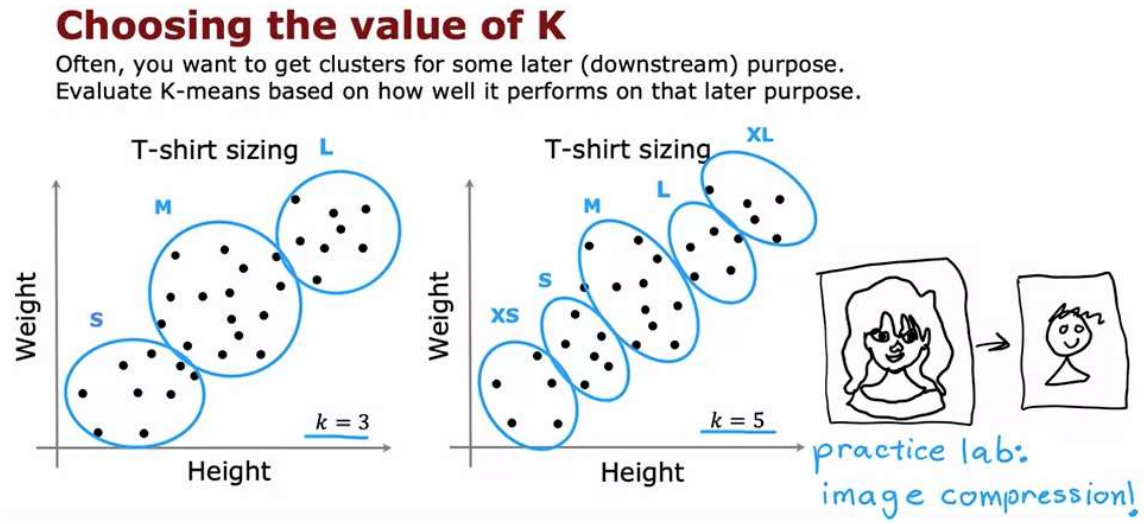


Ich denke, es ist wirklich nicht eindeutig, ob diese Daten zwei, vier oder vielleicht drei Cluster haben. Wenn Sie sagen, sagen der Rote hier und die beiden Blauen hier. Wenn Sie sich die wissenschaftliche Literatur zu K-Means ansehen, gibt es einige Techniken, mit denen Sie versuchen können, die Anzahl der Cluster, die für eine bestimmte Anwendung verwendet werden sollen, automatisch auszuwählen. Ich werde hier kurz eine erwähnen, auf die sich andere vielleicht beziehen, obwohl ich persönlich sagen muss, dass ich diese Methode selbst nicht verwende. Aber eine Möglichkeit, den Wert von K zu wählen, ist die sogenannte Ellbogenmethode. Dabei führt man K-means mit verschiedenen Werten von K durch und zeichnet die Kostenfunktion oder die Verzerrungsfunktion J als Funktion der Zahl auf von Clustern. Sie stellen fest, dass die Verzerrungsfunktion oder die Kostenfunktion J hoch ist, wenn

Sie nur sehr wenige Cluster haben, beispielsweise einen Cluster, und dass sie mit zunehmender Anzahl der Cluster sinkt, möglicherweise wie folgt. Und wenn die Kurve so aussieht, sagen Sie: Nun, es sieht so aus, als ob die Kostenfunktion schnell abnimmt, bis wir drei Cluster erreichen, aber danach erfolgt die Abnahme langsamer. Nehmen wir an, dass K gleich 3 ist, und das nennt man übrigens einen Ellbogen, denn stellen Sie sich das analog vor: Das ist Ihre Hand und das ist Ihr Ellbogen hier. Das Zeichnen der Kostenfunktion als Funktion von K könnte hilfreich sein und Ihnen helfen, einen Einblick zu gewinnen. Ich persönlich verwende die Ellbogenmethode selbst kaum, um die richtige Anzahl von Clustern auszuwählen, weil ich denke, dass die richtige Anzahl von Clustern für viele Anwendungen wirklich nicht eindeutig ist und man stellt fest, dass viele Kostenfunktionen so aussehen und einfach sanft abfallen und es gibt keinen eindeutigen Winkel, mit dem Sie den Wert von K auswählen könnten.



Eine Technik, die übrigens nicht funktioniert, besteht darin, K so zu wählen, dass die Kostenfunktion J minimiert wird, da dies dazu führen würde Wählen Sie fast immer einfach den größtmöglichen Wert von K, da mehr Cluster praktisch immer die Kostenfunktion J verringern. Die Wahl von K zur Minimierung der Kostenfunktion J ist keine gute Technik. Wie wählt man den Wert von K und wie übt man ihn? Häufig führen Sie K-Means aus, um Cluster für spätere oder nachgelagerte Zwecke zu nutzen. Das heißt, Sie nehmen die Cluster und machen etwas mit diesen Clustern. Was ich normalerweise mache und was ich Ihnen empfehle, ist, K-Means danach zu bewerten, wie gut es für den späteren Downstream-Zweck funktioniert. Lassen Sie mich das am Beispiel der T-Shirt-Größe veranschaulichen. Eine Sache, die Sie tun könnten, wäre, K-means auf diesen Datensatz anzuwenden, um die Cluster zu finden. In diesem Fall könnten Sie solche Cluster finden und so die Größe Ihrer kleinen, mittleren und großen T-Shirts bestimmen, aber wie viele t -Hemdgrößen sollte es geben? Nun, es ist nicht eindeutig. Wenn Sie K-means auch mit fünf Clustern ausführen würden, könnten Sie Cluster erhalten, die so aussehen.

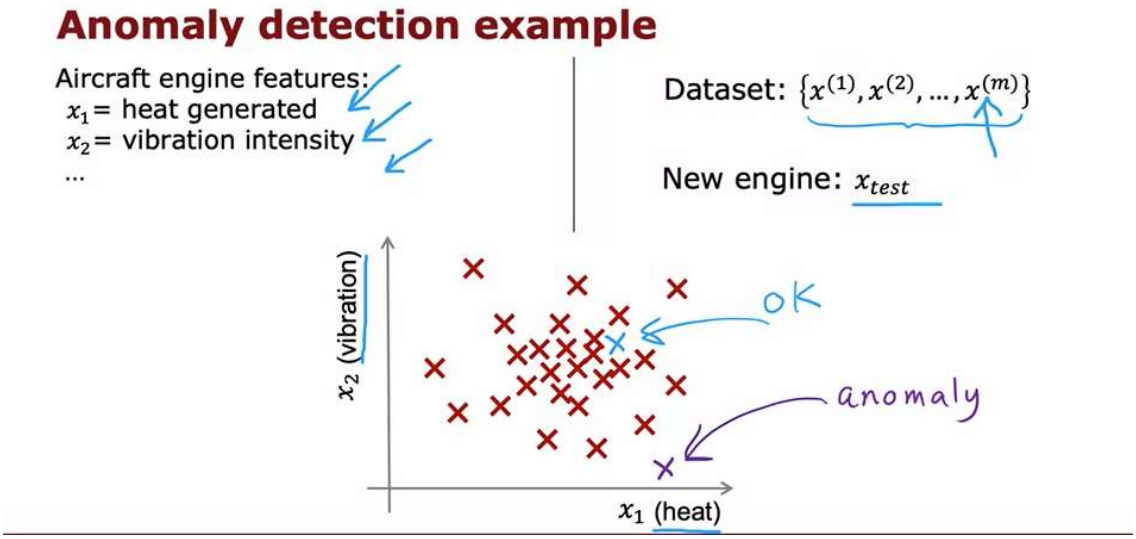


Damit lassen sich T-Shirts in Schuhgrößen in den Größen Extra Small, Small, Medium, Large und Extra Large herstellen. Beides sind völlig gültige und völlig feine Gruppierungen der Daten in Cluster, aber ob Sie drei Cluster oder fünf Cluster verwenden möchten, können Sie jetzt basierend darauf entscheiden, was für Ihr T-Shirt-Geschäft sinnvoll ist. Gibt einen Kompromiss zwischen der Passform der T-Shirts ab, je nachdem, ob Sie drei Größen oder fünf Größen haben, aber es fallen auch zusätzliche Kosten für die Herstellung und den Versand von fünf Arten von T-Shirts anstelle von drei verschiedenen Arten an von T-Shirts. Was ich in diesem Fall tun würde, wäre, K-means mit K gleich 3 und K gleich 5 auszuführen und mir dann diese beiden Lösungen anzusehen, um basierend auf dem Kompromiss zwischen den Passformen von T-Shirts mit mehr Größen zu sehen, dass sich eine bessere Passform ergibt im Vergleich zu den zusätzlichen Kosten für die Herstellung von mehr T-Shirts, bei denen die Herstellung von weniger T-Shirts einfacher und kostengünstiger ist. Versuchen Sie zu entscheiden, was für das T-Shirt-Geschäft sinnvoll ist. Wenn Sie zur Programmierübung kommen, sehen Sie dort auch eine Anwendung von K-Means auf die Bildkomprimierung. Dies ist tatsächlich eines der unterhaltsamsten visuellen Beispiele für K-Mittel, und hier sehen Sie, dass es einen Kompromiss zwischen der Qualität des komprimierten Bildes gibt, d. h. wie gut das Bild aussieht und wie stark Sie es komprimieren können Bild, um Platz zu sparen. In dieser Programmübung sehen Sie, dass Sie diesen Kompromiss nutzen können, um möglicherweise manuell zu entscheiden, was der beste Wert von K ist, basierend darauf, wie gut das Bild aussehen soll und wie groß das komprimierte Bild sein soll. Das ist alles für den K-Means-Clustering-Algorithmus. Herzlichen Glückwunsch zum Erlernen Ihres ersten unbeaufsichtigten Lernalgorithmus. Sie wissen jetzt nicht nur, wie man überwachtes Lernen durchführt, sondern auch wie man unüberwachtes Lernen durchführt. Ich hoffe, Sie haben auch Spaß mit dem Übungslabor, es ist tatsächlich eine der unterhaltsamsten Übungen, die ich über die K-Mittel kenne. Damit können wir mit unserem zweiten unbeaufsichtigten Lernalgorithmus fortfahren, bei dem es sich um eine nichtlineare Erkennung handelt. Wie betrachten Sie den Datensatz und finden darin ungewöhnliche oder anormale Dinge? Dies erweist sich als eine weitere, eine der kommerziell wichtigsten Anwendungen des unbeaufsichtigten Lernens. Ich habe dies selbst schon oft in vielen verschiedenen Anwendungen verwendet. Fahren wir mit dem nächsten Video fort, um über die Anomalie-Erkennung zu sprechen.

Ungewöhnliche Ereignisse finden

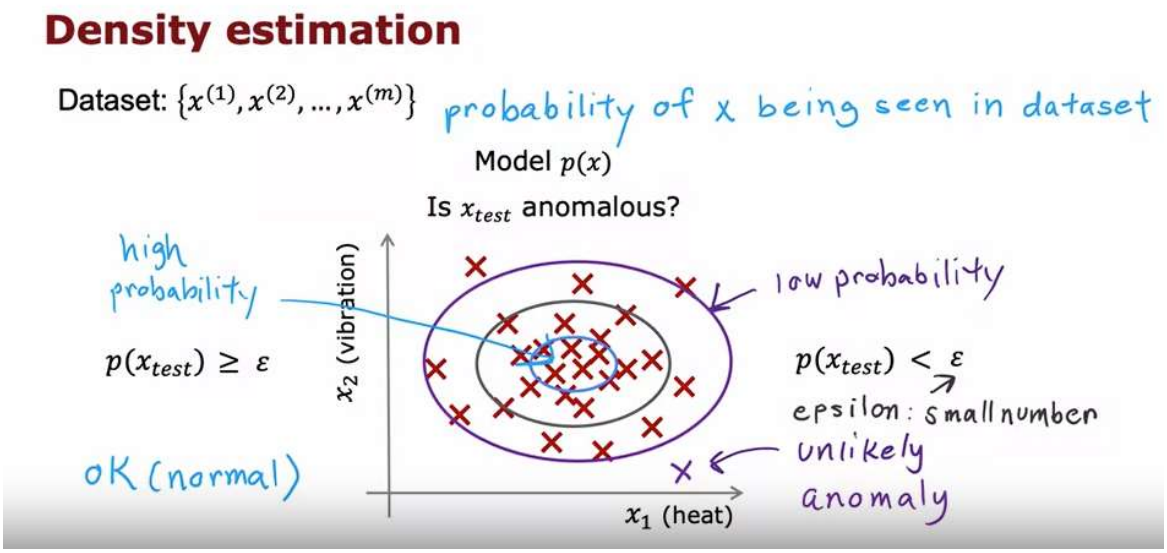
Schauen wir uns unseren zweiten unbeaufsichtigten Lernalgorithmus an. Anomalie-Erkennungsalgorithmen betrachten einen unbeschrifteten Datensatz normaler Ereignisse und lernen so, ein ungewöhnliches oder anomales Ereignis zu erkennen oder eine Warnmeldung zu setzen.

Schauen wir uns ein Beispiel an. Einige meiner Freunde arbeiteten daran, mithilfe der Anomalie-Erkennung mögliche Probleme mit Flugzeugtriebwerken zu erkennen, die gerade hergestellt wurden. Wenn ein Unternehmen ein Flugzeugtriebwerk herstellt, möchte man wirklich, dass dieses Flugzeugtriebwerk zuverlässig ist und gut funktioniert, denn ein Ausfall eines Flugzeugtriebwerks hat sehr negative Folgen. Einige meiner Freunde nutzten die Anomalie-Erkennung, um zu überprüfen, ob ein Flugzeugtriebwerk nach seiner Herstellung anormal schien oder ob irgendetwas daran nicht in Ordnung zu sein schien.



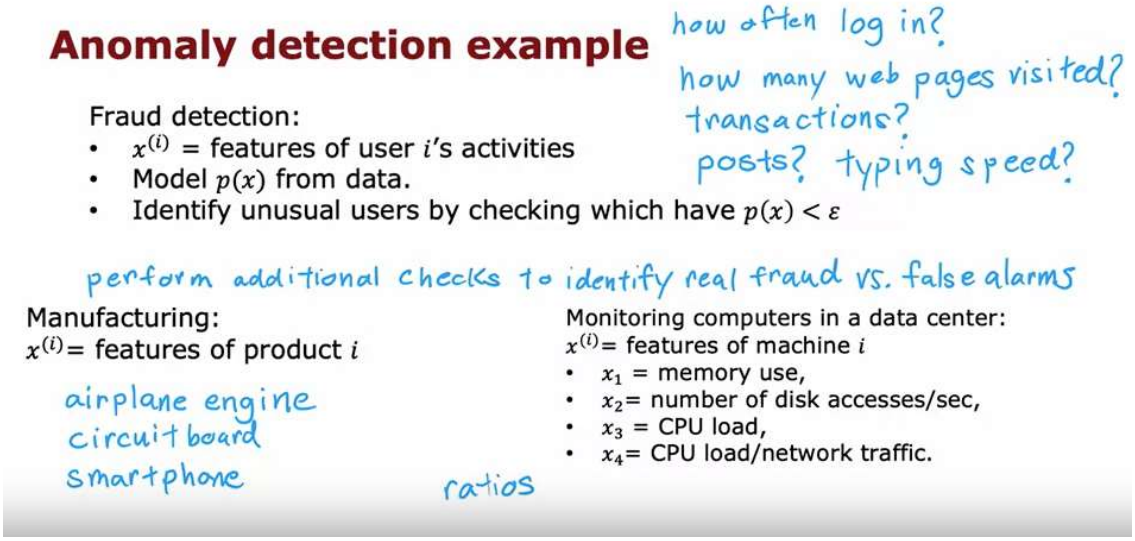
Hier ist die Idee: Nachdem ein Flugzeugtriebwerk vom Band läuft, können Sie eine Reihe unterschiedlicher Merkmale des Flugzeugtriebwerks berechnen. Angenommen, Funktion x1 misst die vom Motor erzeugte Wärme. Feature x2 misst die Vibrationsintensität und so weiter und so weiter für zusätzliche Funktionen. Aber um die Folie ein wenig zu vereinfachen, werde ich nur zwei Merkmale x1 und x2 verwenden, die der Hitze und den Vibrationen des Motors entsprechen. Nun stellt sich heraus, dass Flugzeugtriebwerkshersteller nicht so viele schlechte Triebwerke herstellen. Die leichter zu sammelnde Art von Daten wäre also, wenn man m Flugzeugtriebwerke hergestellt hätte, um die Merkmale x1 und x2 darüber zu sammeln, wie sich diese m Triebwerke verhalten, und wahrscheinlich sind die meisten von ihnen völlig in Ordnung als normale Triebwerke und nicht solche mit einem Defekt oder Durchfluss in ihnen. Und das Problem der Anomalie-Erkennung besteht darin, dass der Lernalgorithmus diese m Beispiele dafür gesehen hat, wie sich Flugzeugtriebwerke typischerweise in Bezug auf die erzeugte Wärme und die Stärke ihrer Vibrationen verhalten. Wenn ein brandneues Flugzeugtriebwerk vom Band laufen würde und es einen neuen Feature-Vektor von Xtest hätte, würden wir gerne wissen, ob dieses Triebwerk denen ähnelt, die zuvor hergestellt wurden? Ist das also wahrscheinlich in Ordnung? Oder gibt es etwas wirklich Seltsames an diesem Motor, das dazu führen könnte, dass seine Leistung verdächtig ist? Das heißt, wir sollten ihn vielleicht noch genauer prüfen, bevor wir ihn ausliefern und in ein Flugzeug einbauen lassen, damit dann hoffentlich nichts schiefgeht. So funktioniert ein Anomalie-Erkennungsalgorithmus. Lassen Sie mich die Beispiele x1 bis xm hier über diese Kreuze darstellen, wobei jeder Kreuzpunkt jedes Datenpunkts in diesem Diagramm einem bestimmten Motor mit einer bestimmten Wärmemenge und einer bestimmten Vibrationsmenge entspricht. Wenn dieser neue Xtest- Flugzeugmotor vom Fließband läuft und Sie diese Werte von x1 und x2 grafisch darstellen würden und wenn er hier wäre, würden Sie sagen: „Okay, das sieht wahrscheinlich in Ordnung aus.“ Sieht anderen Flugzeugtriebwerken sehr ähnlich. Vielleicht brauche ich mir darüber keine Sorgen zu machen. Aber wenn dieses neue Flugzeugtriebwerk eine Hitze- und Vibrationssignatur hat, sagen wir, bis hier unten, dann sieht dieser Datenpunkt hier unten ganz anders aus als damals, als wir ihn oben gesehen haben. Und deshalb

werden wir wahrscheinlich sagen: Junge, das sieht nach einer Anomalie aus. Das sieht nicht wie die Beispiele aus, die ich zuvor gesehen habe. Wir sollten es uns besser genauer ansehen, bevor wir diesen Motor in ein Flugzeug einbauen lassen. Wie kann ein Algorithmus dieses Problem lösen? Die gebräuchlichste Methode zur Anomalie-Erkennung ist eine Technik namens Dichteschätzung.



Und das bedeutet, dass Sie, wenn Sie Ihre Trainingssätze dieser m Beispiele erhalten, als Erstes ein Modell für die Wahrscheinlichkeit von x erstellen. Mit anderen Worten: Der Lernalgorithmus versucht herauszufinden, welche Werte der Merkmale x1 und x2 eine hohe Wahrscheinlichkeit haben und welche Werte weniger wahrscheinlich sind oder eine geringere Wahrscheinlichkeit bzw. geringere Wahrscheinlichkeit haben, im Datensatz angezeigt zu werden. In diesem Beispiel, das wir hier haben, denke ich, dass es sehr wahrscheinlich ist, dass es Beispiele in dieser kleinen Ellipse in der Mitte gibt, so dass dieser Bereich in der Mitte eine hohe Wahrscheinlichkeit hätte, vielleicht haben Dinge in dieser Ellipse eine etwas geringere Wahrscheinlichkeit. Dinge in dieser Ellipse dieses Ovals haben eine noch geringere Wahrscheinlichkeit und Dinge außerhalb dieser Ellipse haben eine noch geringere Wahrscheinlichkeit. Die Details dazu, wie Sie anhand des Trainingssatzes entscheiden, welche Regionen eine höhere oder eine niedrigere Wahrscheinlichkeit haben, werden wir in den nächsten Videos sehen. Und Sie haben modelliert oder gelernt, für p von x zu modellieren. Wenn Sie das neue Testbeispiel Xtest erhalten. Anschließend berechnen Sie die Wahrscheinlichkeit von Xtest. Und wenn es klein ist, oder genauer gesagt, wenn es kleiner als eine kleine Zahl ist, die ich Epsilon nennen werde, ist dies ein Epsilon des griechischen Alphabets. Was Sie sich also als kleine Zahl vorstellen sollten, bedeutet, dass p von x sehr klein ist oder mit anderen Worten, der spezifische Wert von x, den Sie für einen bestimmten Benutzer gesehen haben, war im Vergleich zu anderen Benutzern, die Sie gesehen haben, sehr unwahrscheinlich. Aber der p von Wenn also beispielsweise Xtest ganz hier unten war, ist die Wahrscheinlichkeit, dass ein Beispiel ganz hier landet, eigentlich ziemlich gering. Hoffentlich ist p von Xtest für diesen Wert von Xtest kleiner als Epsilon und wir würden dies als Anomalie kennzeichnen. Wenn dagegen p von Xtest nicht kleiner als Epsilon ist und p von Und diese Antwort auf die Frage, ob Sie hier ein Beispiel hätten, bei dem unser Modell p von x sagen würde, dass Beispiele hier in der Nähe der Mitte tatsächlich eine ziemlich hohe Wahrscheinlichkeit haben. Es besteht eine sehr hohe Wahrscheinlichkeit, dass das neue Flugzeugtriebwerk Merkmale aufweist, die diesen inneren Ellipsen nahekommen. Daher wird p von Xtest für diese Beispiele groß sein und wir sagen, dass es in Ordnung ist und keine Anomalie darstellt. Die Anomalie-Erkennung wird heute in vielen Anwendungen eingesetzt. Es wird häufig zur Betrugserkennung eingesetzt, beispielsweise wenn Sie eine Website mit vielen verschiedenen Funktionen betreiben. Wenn Sie xi berechnen, sind dies die Merkmale der Aktivitäten des Benutzers

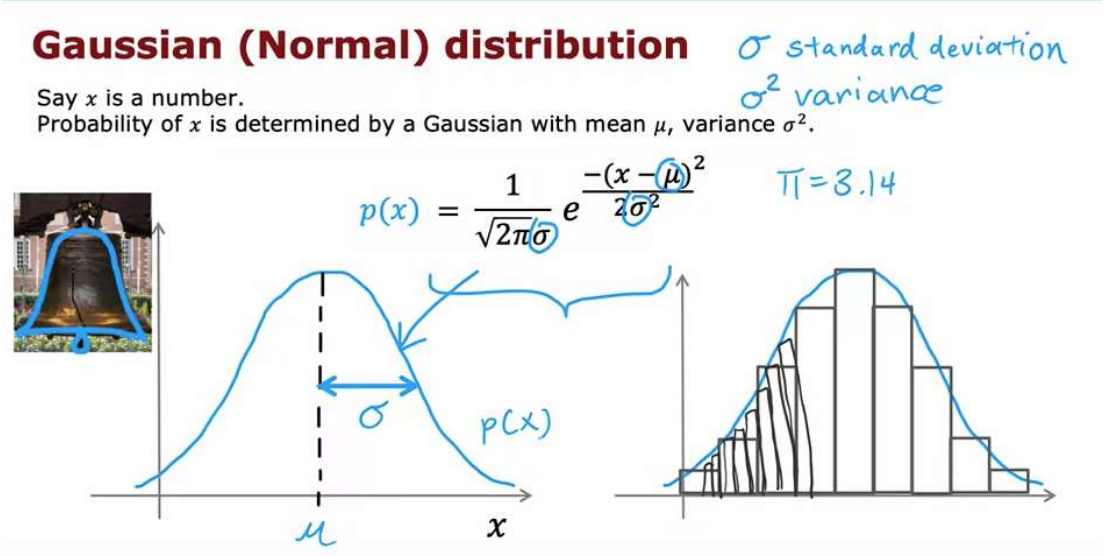
i. Und Beispiele für Funktionen könnten sein: Wie oft meldet sich dieser Benutzer an und wie viele Webseiten besucht er? Wie viele Transaktionen tätigen sie oder wie viele Beiträge im Diskussionsforum erstellen sie bei welcher Schreibgeschwindigkeit? Wie viele Zeichen scheinen sie pro Sekunde tippen zu können? Mit Daten wie diesen können Sie dann p von x aus Daten modellieren, um das typische Verhalten eines bestimmten Benutzers zu modellieren. Im üblichen Arbeitsablauf zur Betrugserkennung würden Sie ein Konto nicht automatisch schließen, nur weil es ungewöhnlich erscheint. Stattdessen können Sie das Sicherheitsteam bitten, genauer hinzusehen oder zusätzliche Sicherheitsüberprüfungen durchzuführen, z. B. den Benutzer zu bitten, seine Identität anhand einer Mobiltelefonnummer zu bestätigen, oder ihn zu bitten, eine Erfassung zu bestehen, um zu beweisen, dass er ein Mensch ist usw An. Aber Algorithmen wie dieser werden heutzutage routinemäßig verwendet, um ungewöhnliche oder vielleicht leicht verdächtige Aktivitäten zu finden. So können Sie diese Konten sorgfältiger überprüfen, um sicherzustellen, dass kein Betrug vorliegt. Und diese Art der Betrugserkennung wird sowohl zum Auffinden gefälschter Konten als auch häufig verwendet, um Finanzbetrug zu erkennen, beispielsweise wenn ein sehr ungewöhnliches Kaufmuster vorliegt. Dann könnte es sich durchaus lohnen, dass das Sicherheitsteam sich das genauer anschaut. Auch in der Fertigung kommt die Anomalie-Erkennung häufig zum Einsatz. Auf der vorherigen Folie haben Sie ein Beispiel zur Herstellung von Flugzeugtriebwerken gesehen. Aber viele Hersteller auf mehreren Kontinenten und in vielen, vielen Fabriken verwendeten routinemäßig die Anomalie-Erkennung, um zu sehen, ob das, was sie gerade hergestellt hatten. Alles, von einem Flugzeugmotor über eine Leiterplatte, ein Smartphone bis hin zu einem Motor, bis hin zu vielen, vielen Dingen, die man sehen kann, wenn man gerade das Gerät hergestellt hat, das sich irgendwie seltsam verhält. Denn das kann darauf hindeuten, dass etwas mit Ihrem Flugzeugmotor oder den Leiterplatten nicht stimmt oder was auch immer, was dazu führen könnte, dass Sie genauer hinsehen möchten, bevor Sie das Objekt an den Kunden versenden. Wird auch zur Überwachung von Computern in Clustern und in Rechenzentren verwendet, wo XI die Funktionen einer bestimmten Maschine sind, z. B. ob die Funktionen die Speicherbenutzer erfassen, die Anzahl der Festplattenzugriffe pro Sekunde. CPU-Auslastungsmerkmale können auch rassistisch sein, beispielsweise das Verhältnis von CPU-Auslastung zu Netzwerkverkehr. Wenn sich dann ein bestimmter Computer jemals ganz anders verhält als andere Computer, lohnt es sich möglicherweise, einen Blick auf diesen Computer zu werfen, um festzustellen, ob etwas mit ihm nicht stimmt. Zum Beispiel, wenn ein Hot-Test oder ein Netzwerk-Hack fehlgeschlagen ist, etwas nicht stimmt oder möglicherweise gehackt wurde.



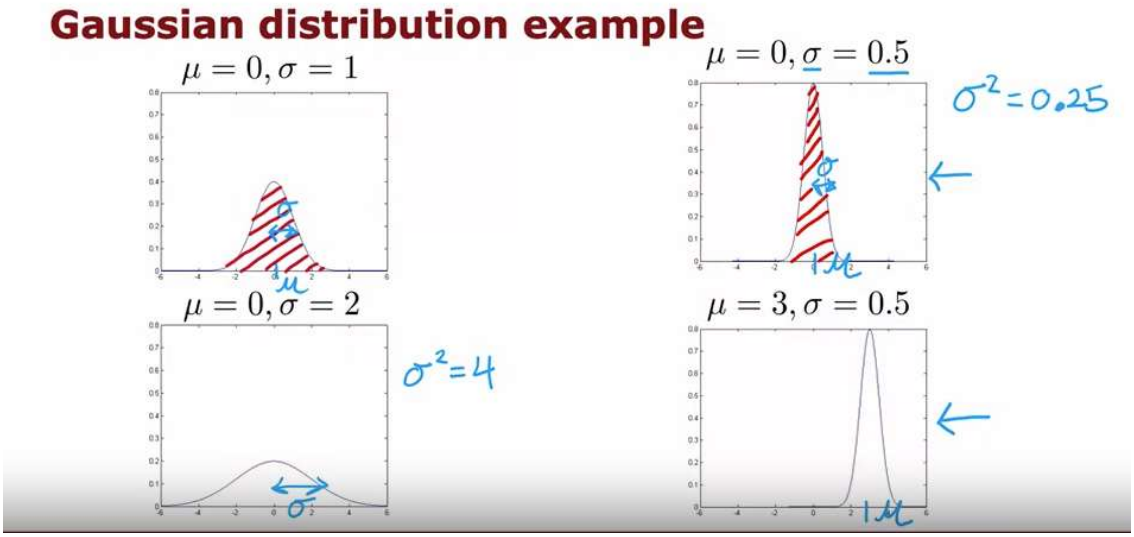
Die Anomalie-Erkennung ist einer dieser Algorithmen, der sehr weit verbreitet ist, auch wenn man anscheinend nicht so viel darüber reden hört. Ich erinnere mich an das erste Mal, dass ich an der kommerziellen Anwendung der Anomalie-Erkennung arbeitete, als ich einem Telekommunikationsunternehmen dabei half, die Anomalie-Erkennung einzuführen, um zu erkennen, wann sich einer der Mobilfunkmasten ungewöhnlich verhielt. Denn das bedeutete wahrscheinlich, dass mit dem Mobilfunkmast etwas nicht stimmte, und deshalb wollen sie einen Techniker damit beauftragen, damit hoffentlich mehr Menschen eine gute Mobilfunkabdeckung erhalten. Und ich habe die Anomalie-Erkennung auch verwendet, um betrügerische Finanztransaktionen aufzudecken. Heutzutage nutze ich sie oft, um produzierenden Unternehmen dabei zu helfen, anomale Teile zu finden, die sie möglicherweise hergestellt haben, aber häufiger prüfen sollten. Daher ist es ein sehr nützliches Werkzeug, das Sie in Ihrer Werkzeugkiste haben sollten. Und in den nächsten Videos werden wir darüber sprechen, wie Sie diese Algorithmen selbst erstellen und zum Laufen bringen können. Damit anonyme Erkennungsalben funktionieren, müssen wir eine Gaußsche Verteilung verwenden, um die Daten p von x zu modellieren. Fahren wir also mit dem nächsten Video fort, um über Gaußsche Verteilungen zu sprechen.

Gaußsche (normale) Verteilung

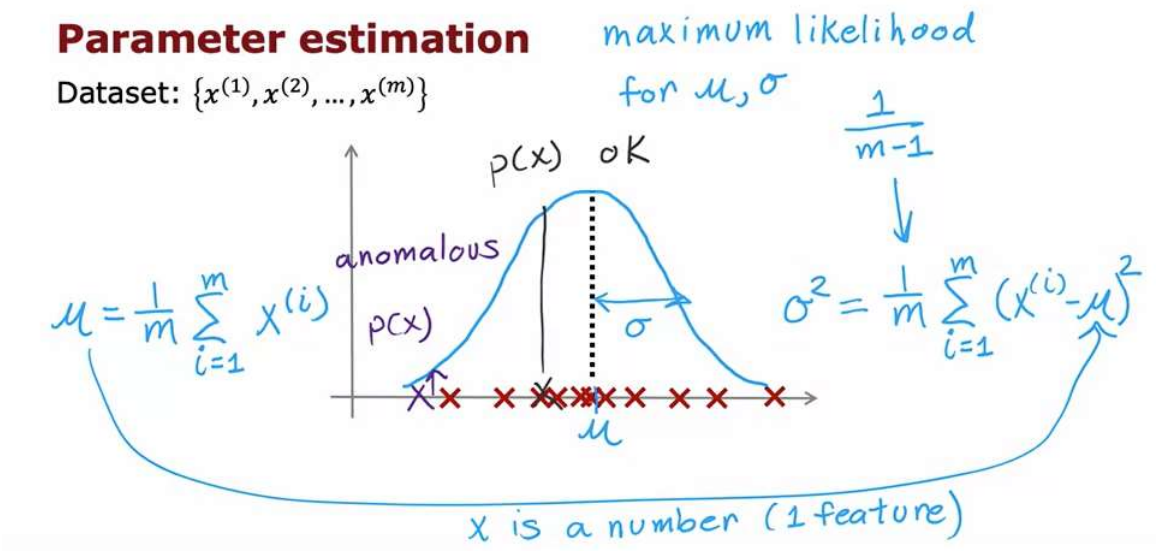
Um die Anomalie-Erkennung anwenden zu können, müssen wir die Gaußsche Verteilung verwenden, die auch Normalverteilung genannt wird. Wenn Sie mich entweder Gaußsche Verteilung oder Normalverteilung sagen hören, meinen sie genau dasselbe. Wenn Sie die glockenförmige Verteilung gehört haben, bezieht sich das auch auf dasselbe. Aber wenn Sie noch nie von der glockenförmigen Verteilung gehört haben, ist das auch in Ordnung. Aber werfen wir einen Blick darauf, was die Gauß- oder Normalverteilung ist. Angenommen, x ist eine Zahl, und wenn x eine Zufallszahl ist, manchmal auch Zufallsvariable genannt, kann x Zufallswerte annehmen. Wenn die Wahrscheinlichkeit von x durch eine Gauß- oder Normalverteilung mit dem Mittelwertparameter Mu und der Varianz im Sigma-Quadrat gegeben ist. Das bedeutet, dass die Wahrscheinlichkeit von x wie eine Kurve aussieht, die so verläuft. Der Mittelpunkt oder die Mitte der Kurve wird durch den Mittelwert Mu angegeben, und die Standardabweichung oder die Breite dieser Kurve wird durch den Varianzparameter Sigma angegeben. Technisch gesehen wird Sigma als Standardabweichung und das Quadrat von Sigma oder Sigma-Quadrat als Varianz der Verteilung bezeichnet. Diese Kurve hier zeigt, was p von x oder die Wahrscheinlichkeit von x ist. Wenn Sie von der glockenförmigen Kurve gehört haben, dann ist dies diese glockenförmige Kurve, denn viele klassische Glocken in Türmen sagen, dass sie so geformt waren, dass der Glockenklöppel hier herunterhing, und daher erinnert die Form dieser Kurve vage daran von der Form der großen Glocken, die man noch heute in manchen alten Gebäuden findet. Sieht besser aus als mein handgezeichnetes. Es gibt ein Bild der Freiheitsglocke. Tatsächlich ist die Form der Freiheitsglocke an der Oberseite annähernd glockenförmig geschwungen. Wenn Sie sich fragen, was p von x wirklich bedeutet? Hier ist eine Möglichkeit, es zu interpretieren. Das heißt, wenn Sie beispielsweise 100 Zahlen aus dieser Wahrscheinlichkeitsverteilung erhalten würden und ein Histogramm dieser 100 aus dieser Verteilung gezogenen Zahlen zeichnen würden, könnten Sie ein Histogramm erhalten, das so aussieht. Es sieht vage glockenförmig aus. Was diese Kurve auf der linken Seite anzeigt, ist nicht, ob Sie nur 100 Beispiele oder 1.000 oder eine Million oder eine Milliarde haben.



Aber wenn Sie eine praktisch unendliche Anzahl von Beispielen hätten und ein Histogramm dieser praktisch unendlichen Anzahl von Beispielen mit einem sehr feinen Histogrammbereich zeichnen würden. Dann erhalten Sie im Wesentlichen diese glockenförmige Kurve hier links. Die Formel für p von x wird durch diesen Ausdruck gegeben; p von x ist gleich 1 über der Quadratwurzel 2 Pi. Pi ist hier 3,14159 oder etwa 22 über 7. Verhältnis von Durchmesser und Umfang eines Kreises mal Sigma mal e zum negativen x minus Mu, dem mittleren Parameterquadrat dividiert durch 2 Sigmaquadrat. Wenn Sie diese Funktion für jeden gegebenen Wert von Mu und Sigma als Funktion von x darstellen, erhalten Sie diese Art von glockenförmiger Kurve, die bei Mu zentriert ist und deren Breite durch bestimmt wird der Parameter Sigma. Schauen wir uns nun einige Beispiele an, wie sich eine Änderung von Mu und Sigma auf die Gaußsche Verteilung auswirkt. Lassen Sie mich zunächst Mu gleich 0 und Sigma gleich 1 setzen. Hier ist mein Diagramm der Gaußschen Verteilung mit Mittelwert 0, Mu gleich 0 und Standardabweichung Sigma gleich 1. Sie bemerken, dass diese Verteilung bei Null zentriert ist und das die Standardabweichung ist Sigma ist gleich 1. Reduzieren wir nun die Standardabweichung Sigma auf 0,5. Wenn Sie die Gaußsche Verteilung zeichnen, wobei Mu gleich 0 und Sigma gleich 0,5 ist, sieht es jetzt so aus. Beachten Sie, dass es immer noch auf Null zentriert ist, da Mu Null ist. Aber die Kurve ist viel dünner geworden, weil Sigma jetzt 1,5 beträgt. Sie erinnern sich vielleicht, dass Sigma die Standardabweichung von 0,5 ist, während Sigma im Quadrat auch Varianz genannt wird. Das entspricht 0,5 zum Quadrat oder 0,25. Sie haben vielleicht gehört, dass sich Wahrscheinlichkeiten immer zu eins summieren müssen, weshalb die Fläche unter der Kurve immer gleich eins ist. Deshalb muss die Gauß-Verteilung auch größer werden, wenn sie dünner wird. Schauen wir uns einen anderen Wert von Mu und Sigma an. Jetzt erhöhe ich Sigma auf 2, sodass die Standardabweichung 2 und die Varianz 4 beträgt. Dies führt nun zu einer viel breiteren Verteilung, da Sigma hier jetzt viel größer ist, und weil es jetzt eine breitere Verteilung ist, ist es kürzer geworden Nun, weil die Fläche unter der Kurve immer noch gleich 1 ist.



Versuchen wir abschließend, den Mittelwertparameter Mu zu ändern, und ich belasse Sigma bei 0,5. In diesem Fall verschiebt sich das Zentrum der Verteilung Mu hier nach rechts. Die Breite der Verteilung ist jedoch dieselbe wie oben, da die Standardabweichung in beiden Fällen rechts 0,5 beträgt. Auf diese Weise wirken sich unterschiedliche Entscheidungen von Mu und Sigma auf die Gaußsche Verteilung aus. Wenn Sie dies auf die Anomalie-Erkennung anwenden, müssen Sie Folgendes tun. Sie erhalten einen Datensatz mit m Beispielen, und hier ist x nur eine Zahl. Hier sind Diagramme der Trainingssätze mit 11 Beispielen. Was wir tun müssen, ist abzuschätzen, was eine gute Wahl für den Mittelwertparameter Mu sowie für den Varianzparameter Sigma-Quadrat ist. Angesichts eines Datensatzes wie diesem scheint es, dass eine Gaußsche Verteilung möglicherweise so aussieht, mit einem Zentrum hier und einer solchen Standardabweichung. Das könnte ziemlich gut zu den Daten passen. Die Art und Weise, wie Sie Mu und Sigma im Quadrat mathematisch berechnen würden, ist, dass unsere Schätzung für Mu nur der Durchschnitt aller Trainingsbeispiele ist. Es ist 1 über m mal die Summe von i gleich 1 bis m der Werte Ihrer Trainingsbeispiele. Der Wert, den wir zur Schätzung des Sigma-Quadrats verwenden werden, ist der Durchschnitt der quadrierten Differenz zwischen zwei Beispielen und dem Mu, das Sie gerade hier links geschätzt haben. Es stellt sich heraus, dass Sie, wenn Sie diese beiden Formeln mit diesem Wert für Mu und diesem Wert für Sigma im Quadrat implementieren, so ziemlich die Gaußsche Verteilung erhalten, die ich oben von Hand gezeichnet habe. Dadurch haben Sie die Wahl zwischen Mu und Sigma für eine Gauß- Verteilung, sodass es aussieht, als wären die 11 Trainingsstichproben aus dieser Gauß-Verteilung gezogen worden.

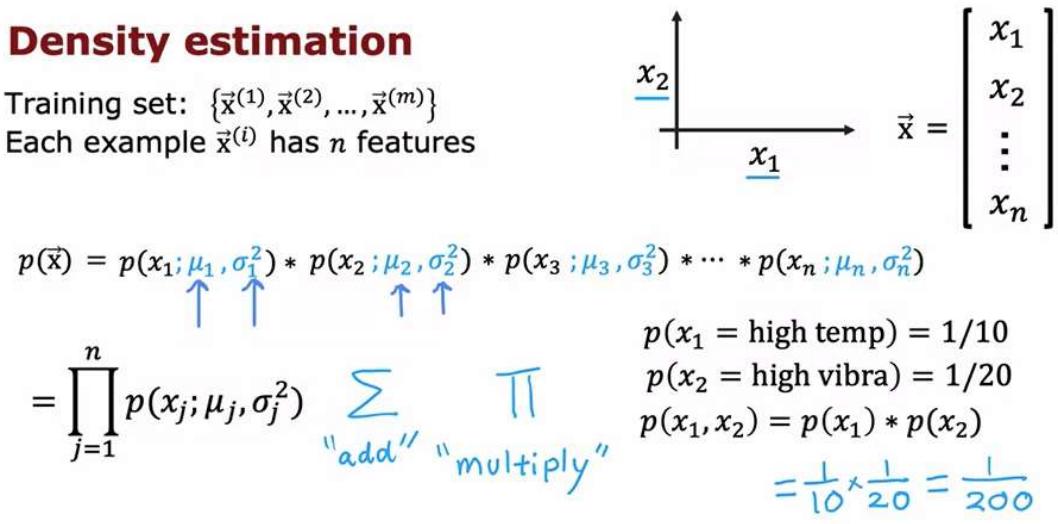


Wenn Sie an einem Statistikkurs für Fortgeschrittene teilgenommen haben, haben Sie vielleicht gehört, dass diese Formeln für Mu und Sigma im Quadrat technisch als Maximum-Likelihood-Schätzungen für Mu und Sigma bezeichnet werden. In einigen Statistikklassen werden Sie aufgefordert, die Formel 1 über n minus 1 anstelle von 1 über m zu verwenden. In der Praxis macht die Verwendung von 1 über m oder 1 über n minus 1 kaum einen Unterschied. Ich verwende immer 1 über m, aber es sind nur einige andere Eigenschaften der Division durch m minus 1, die einige Statistiker bevorzugen. Aber wenn Sie nicht verstehen, was sie gerade gesagt haben, machen Sie sich darüber keine Sorgen. Alles, was Sie wissen müssen, ist, dass Sie, wenn Sie Mu gemäß dieser Formel und Sigma quadrieren gemäß dieser Formel festlegen, eine ziemlich gute Schätzung von Mu und Sigma erhalten und insbesondere eine Gaußsche Verteilung erhalten, die eine mögliche Wahrscheinlichkeit darstellt Verteilung im Hinblick auf die Wahrscheinlichkeitsverteilung, aus der die Trainingsbeispiele stammen. Sie können wahrscheinlich erraten, was als nächstes kommt. Wenn Sie hier ein Beispiel sehen würden, dann ist p von x ziemlich hoch. Wenn Sie dagegen ein Beispiel bekommen würden, dann wäre p von x ziemlich niedrig, weshalb wir dieses Beispiel als okay betrachten würden, nicht wirklich anomal, den anderen nicht sehr ähnlich. Während ein Beispiel, das wir hier haben, im Vergleich zu den Beispielen, die wir gesehen haben, ziemlich ungewöhnlich ist und daher anomaler, weil p von x, das ist die Höhe dieser Kurve, hier links viel niedriger ist als an diesem Punkt hier, näher an der Mitte. Wir haben dies nur für den Fall getan, dass x eine Zahl ist, als ob Sie nur eine einzige Funktion für Ihr Anomalie-Erkennungsproblem hätten. Für praktische Anomalie-Erkennungsanwendungen stehen Ihnen normalerweise viele verschiedene Funktionen zur Verfügung. Sie haben jetzt gesehen, wie die Gaußsche Verteilung funktioniert. Wenn x eine einzelne Zahl ist, entspricht dies dem Fall, wenn Sie beispielsweise nur eine Funktion für Ihr Anomalie-Erkennungsproblem hätten. Aber für praktische Anomalie-Erkennungsanwendungen stehen Ihnen viele Funktionen zur Verfügung, zwei oder drei oder eine noch größere Anzahl n von Funktionen. Nehmen wir, was Sie für einen einzelnen Gaußschen Wert gesehen haben, und verwenden Sie es, um einen ausgefeilteren Algorithmus zur Anomalie-Erkennung zu erstellen. Sie können mehrere Funktionen verarbeiten. Machen wir das im nächsten Video.

Algorithmus zur Anomalie-Erkennung

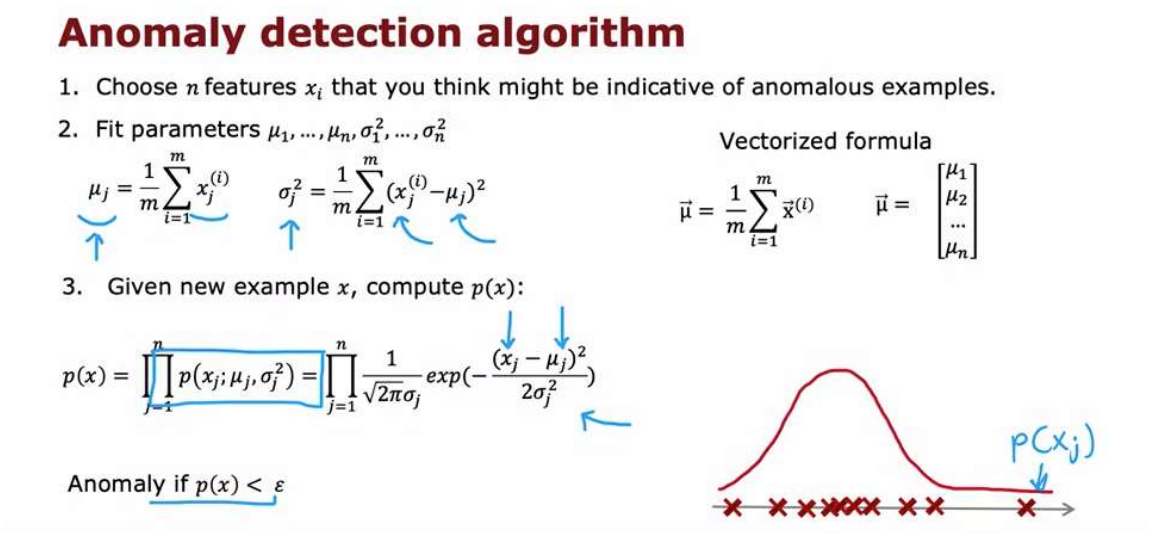
Nachdem Sie nun gesehen haben, wie die Gauß- oder Normalverteilung für eine einzelne Zahl funktioniert, können wir unseren Algorithmus zur Anomalie-Erkennung erstellen. Lassen Sie uns eintauchen. Sie haben einen Trainingssatz x1 bis xm, wobei hier jedes Beispiel x Endmerkmale hat. Jedes Beispiel x ist also ein Vektor innerhalb von Zahlen. Im Beispiel des Flugzeugtriebwerks hatten

wir zwei Merkmale, die der Hitze und den Vibrationen entsprachen. Jede dieser Verbrauchsteuern wäre also ein zweidimensionaler Vektor und n wäre gleich 2. Für viele praktische Anwendungen kann n jedoch viel größer sein, und Sie können dies möglicherweise mit Dutzenden oder sogar Hunderten von Merkmalen tun. Angesichts dieses Trainingssatzes möchten wir eine Dichteschätzung durchführen. Das bedeutet lediglich, dass wir ein Modell erstellen oder die Wahrscheinlichkeit für p(x) schätzen. Wie hoch ist die Wahrscheinlichkeit eines bestimmten Merkmalsvektors? Und unser Modell für p(x) wird wie folgt aussehen: x ist ein Merkmalsvektor mit den Werten x1, x2 usw. bis hin zu xn. Und ich werde p(x) als die Wahrscheinlichkeit von x1 mal der Wahrscheinlichkeit von x2 mal der Wahrscheinlichkeit von x3 mal der Wahrscheinlichkeit von xn für die n -ten Merkmale in den Merkmalsvektoren modellieren. Wenn Sie wahrscheinlich schon einmal einen Kurs für Fortgeschrittene in Statistik besucht haben, erkennen Sie vielleicht, dass diese Gleichung der Annahme entspricht, dass die Merkmale x1, x2 usw. bis xm statistisch unabhängig sind. Es stellt sich jedoch heraus, dass dieser Algorithmus oft gut funktioniert, auch wenn die Merkmale tatsächlich statistisch nicht unabhängig sind. Aber wenn Sie nicht verstehen, was ich gerade gesagt habe, machen Sie sich darüber keine Sorgen. Das Verständnis der physischen Unabhängigkeit ist nicht erforderlich, um die Anomalie-Erkennungsalgorithmen vollständig zu besprechen und sehr effektiv zu nutzen. Um diese Gleichung noch ein wenig zu ergänzen, sagen wir, dass die Wahrscheinlichkeit aller Merkmale dieses Vektors x das Produkt von p(x) 1 und p(x2) ist und so weiter bis p(xn).). Und um die Wahrscheinlichkeit von x1 zu modellieren, sagen wir, die Wärmefunktion in diesem Beispiel, werden wir zwei Parameter haben, neu 1 und Sigma 1 oder Sigma im Quadrat ist 1. Und das bedeutet, dass wir schätzen werden, die Mittelwert des Merkmals x1 und auch die Varianz des Merkmals x1, und das wird neu 1 und Sigma 1 sein. Die Modellierung von p(x2) x2 ist ein völlig anderes Merkmal, das die Vibrationen des Flugzeugtriebwerks misst. Wir werden zwei verschiedene Parameter haben, die ich als Mu 2, Sigma 2 im Quadrat schreiben werde.



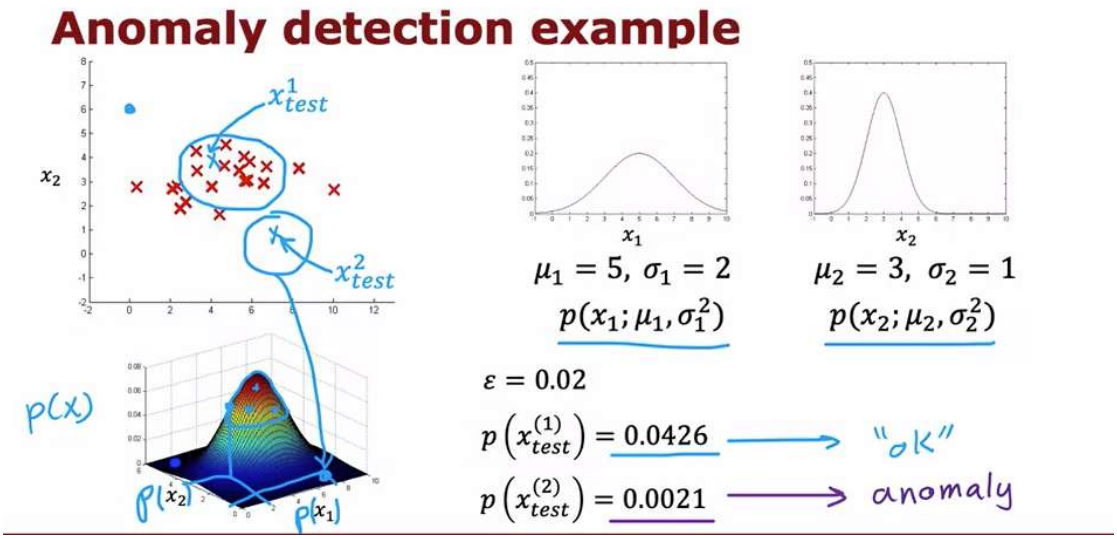
Und es stellt sich heraus, dass dies dem Mittelwert oder Durchschnitt des Schwingungsmerkmals und der Varianz des Schwingungsmerkmals usw. entspricht. Wenn Sie zusätzliche Funktionen haben, mu 3 Sigma 3 zum Quadrat bis mu n und Sigma n zum Quadrat. Falls Sie sich fragen, warum wir Wahrscheinlichkeiten multiplizieren, finden Sie hier vielleicht ein Beispiel, das die Intuition fördern könnte. Angenommen, bei einem Flugzeugtriebwerk besteht eine Wahrscheinlichkeit von 1/10, dass es wirklich heiß ist, ungewöhnlich heiß, und möglicherweise eine Wahrscheinlichkeit von 1 zu 20, dass es sehr stark vibriert. Wie groß ist dann die Wahrscheinlichkeit, dass es richtig heiß läuft und sehr stark vibriert? Wir sagen, dass die Wahrscheinlichkeit dafür 1/10 mal 1/20 beträgt, also 1/200. Daher ist es wirklich unwahrscheinlich, dass der Motor an beiden Fronten sehr heiß ist und sehr stark vibriert. Es ist das Produkt dieser beiden Wahrscheinlichkeiten. Eine etwas kompaktere Art, diese

Gleichung hier oben zu schreiben, besteht darin, zu sagen, dass sie gleich dem Produkt von j = 1 bis n von p(xj) ist. Wären die Parameter mu j und Sigma zum Quadrat j. Und dieses Symbol hier ähnelt stark dem Summationssymbol, mit der Ausnahme, dass das Summationssymbol der Addition entspricht, dieses Symbol hier jedoch der Multiplikation dieser Terme hier für j = 1 bis n entspricht. Lassen Sie uns also alles zusammenfassen, um zu sehen, wie Sie das Nominierungserkennungssystem integrieren können. Der erste Schritt besteht darin, Merkmale xi auszuwählen, die Ihrer Meinung nach auf anomale Beispiele hinweisen könnten. Nachdem Sie die Features gefunden haben, die Sie verwenden möchten, passen Sie dann die Parameter mu 1 bis mu n und Sigma-Quadrat 1 bis Sigma-Quadrat n für die n Features in Ihrem Datensatz an. Wie Sie vielleicht erraten haben, ist der Parameter mu j nur der Durchschnitt von xj des Merkmals j aller Beispiele in Ihrem Trainingssatz. Und Sigma-Quadrat j ist der Durchschnitt der quadratischen Differenz zwischen dem Merkmal und dem Wert mu j, den Sie gerade berechnet haben. Und übrigens, wenn Sie eine vektorisierte Implementierung haben, können Sie mu auch wie folgt als Durchschnitt der Trainingsbeispiele berechnen: Wir sind hier, x und mu sind beide Vektoren. Dies wäre also die vektorisierte Art, mu 1 bis mu und alle gleichzeitig zu berechnen. Und indem Sie diese Parameter auf Ihrem unbeschrifteten Trainingssatz schätzen, haben Sie nun alle Parameter Ihres Modells berechnet. Wenn Sie schließlich ein neues Beispiel erhalten, einen x-Test oder ich werde hier einfach ein neues Beispiel als x schreiben, würden Sie p(x) berechnen und prüfen, ob es groß oder klein ist. Wie Sie auf der letzten Folie gesehen haben, ist p(x) also das Produkt von j = 1 bis n der Wahrscheinlichkeit der einzelnen Merkmale. Also p(x) j mit den Parametern mu j und Einzelquadrat j. Und wenn Sie die Formel für diese Wahrscheinlichkeit einsetzen, erhalten Sie diesen Ausdruck 1 über Wurzel 2 pi sigma j von e zu diesem Ausdruck hier. Und so sind xj die Merkmale, dies ist ein j-Merkmal Ihres neuen Beispiels, mu j und Sigma j sind Zahlen oder Parameter, die Sie im vorherigen Schritt berechnet haben.



Und wenn Sie diese Formel berechnen, erhalten Sie eine Zahl für p(x). Und der letzte Schritt besteht darin, zu sehen, dass ap(x) kleiner als Epsilon ist. Und wenn ja, dann weisen Sie darauf hin, dass es sich um eine Anomalie handelt. Eine Intuition hinter dem, was dieses Album tut, ist, dass es dazu neigt, ein Beispiel als anomal zu kennzeichnen, wenn eines oder mehrere der Features im Vergleich zu dem, was es im Trainingssatz gesehen hat, entweder sehr groß oder sehr klein sind. Für jedes der Merkmale xj passen Sie also eine Gaußsche Verteilung wie diese an. Wenn also beispielsweise auch nur eines der Merkmale des neuen Beispiels hier weit draußen wäre, dann wäre P f xJ sehr klein. Und wenn nur einer der Terme in diesem Produkt sehr klein ist, dann wird dieses Gesamtprodukt, wenn man es miteinander multipliziert, tendenziell sehr klein sein und p(x) wird klein sein. Und was die Anomalie-Erkennung in den Algorithmus einbezieht, ist eine systematische Methode zur

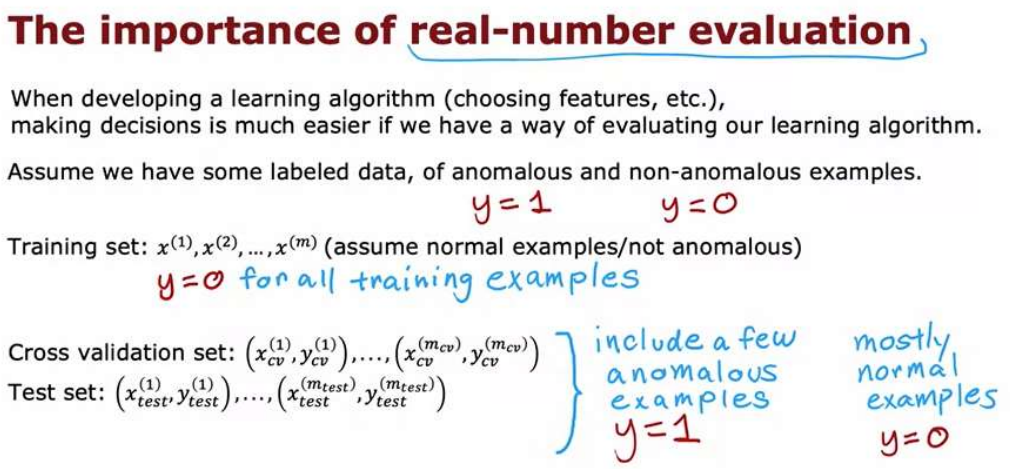
Quantifizierung, ob dieses neue Beispiel x ungewöhnlich große oder ungewöhnlich kleine Merkmale aufweist oder nicht. Schauen wir uns nun anhand eines Beispiels an, was das alles tatsächlich bedeutet. Hier ist der Datensatz mit den Merkmalen x 1 und x 2. Und Sie bemerken, dass die Merkmale x 1 einen viel größeren Wertebereich annehmen als die Merkmale x 2. Wenn Sie das Meme der Features x 1 berechnen würden, erhalten Sie am Ende fünf, weshalb Sie gleich 1 wollen. Und es stellt sich heraus, dass für diese Daten, wenn Sie Sigma 1 berechnen, es gleich sein wird etwa 2. Und wenn Sie mu berechnen würden, beträgt der Durchschnitt der Merkmale neben dem Durchschnitt drei und in ähnlicher Weise ist die Varianz oder Standardabweichung viel kleiner, weshalb Sigma 2 gleich 1 ist. Das entspricht also dieser Gaußschen Funktion Verteilung für x 1 und diese Gaußsche Verteilung für x 2. Wenn Sie p(x) 1 und p(x) 2 tatsächlich multiplizieren würden, dann erhalten Sie dieses Drei-D-Oberflächendiagramm für p(x), wobei jeder Punkt, der Die Höhe davon ist das Produkt von p(x) 1 mal p(x) 2. Für die entsprechenden Werte von x 1 und x 2. Und dies bedeutet, dass Werte, bei denen p(x) höher oder wahrscheinlicher ist. Daher sind hier Werte in der Nähe der Mittellinie wahrscheinlicher.



Während Werte hier weit draußen liegen, sind Werte hier draußen viel unwahrscheinlicher. Ich habe eine viel geringere Chance. Lassen Sie mich nun Testbeispiele auswählen, das erste 1 hier, ich werde diesen x-Test 1 und den zweiten 1 hier unten als x-Test 2 schreiben. Und schauen wir uns an, welches dieser beiden Beispiele das Album als anomal markiert. Ich werde den Parameter ε gleich 0,02 wählen. Und wenn Sie p(x)-Test 1 berechnen würden, ergibt sich ein Wert von etwa 0,4, was viel größer als Epsilon ist. Und so wird das Album sagen, dass es okay aussieht und nicht wie eine Anomalie aussieht. Wenn Sie dagegen p(x) für diesen Punkt hier unten berechnen würden, entspricht x 1 etwa acht und x 2 etwa 0,5. Irgendwie hier unten, dann ist p(x)-Test 2 0,0021. Das ist also viel kleiner als Epsilon. Daher wird das Album dies als wahrscheinliche Anomalie kennzeichnen. So, wie Sie vielleicht hoffen, kommt es zu dem Schluss, dass x-Test 1 ganz normal aussieht. Wohingegen ein Überschuss, der viel weiter entfernt ist als alles, was Sie im Trainingssatz sehen, so aussieht, als könnte es sich um eine Anomalie handeln. Sie haben also gesehen, wie ein Anomalie-Erkennungssystem aufgebaut wird. Aber wie wählt man den Parameter Epsilon? Und woher wissen Sie, ob Ihr Anomalie-Erkennungssystem im nächsten Video gut funktioniert? Lassen Sie uns im nächsten Video etwas tiefer in den Prozess der Entwicklung und Bewertung der Leistung eines Anomalie-Erkennungssystems eintauchen. Kommen wir zum nächsten Video

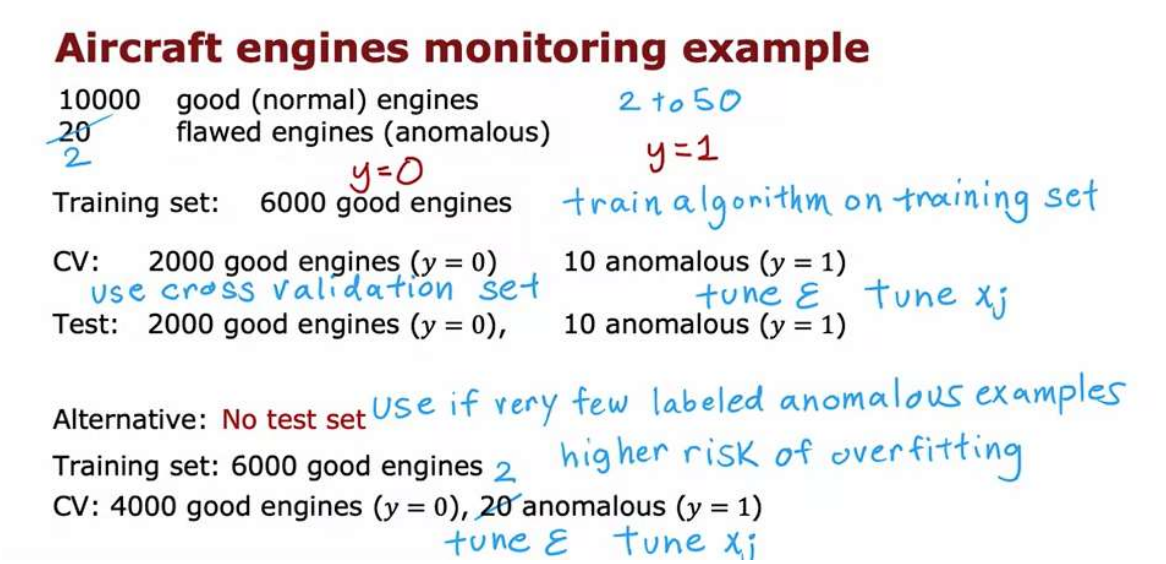
Entwicklung und Evaluierung eines Anomalie-Erkennungssystems

Ich möchte Ihnen einige praktische Tipps für die Entwicklung eines Anomalie-Erkennungssystems geben. Eine der Schlüsselideen wird sein, dass Sie, wenn Sie die Möglichkeit haben, ein System bereits während seiner Entwicklung zu bewerten, viel schneller Entscheidungen treffen, das System ändern und verbessern können. Werfen wir einen Blick darauf, was das bedeutet. Wenn Sie einen Lernalgorithmus entwickeln, beispielsweise indem Sie verschiedene Funktionen auswählen oder verschiedene Werte der Parameter wie Epsilon ausprobieren, Entscheidungen darüber treffen, ob eine Funktion auf eine bestimmte Weise geändert werden soll oder nicht oder Epsilon oder andere Parameter erhöht oder verringert werden sollen, ist das Treffen dieser Entscheidungen eine Herausforderung viel einfacher, wenn Sie eine Möglichkeit haben, den Lernalgorithmus zu bewerten. Dies wird manchmal als Auswertung reeller Zahlen bezeichnet. Dies bedeutet, dass Sie den Algorithmus schnell auf irgendeine Weise ändern können, z. B. eine Funktion oder einen Parameter ändern und eine Zahl berechnen können, die Ihnen sagt, ob der Algorithmus besser oder schlechter geworden ist Dies macht es viel einfacher zu entscheiden, ob diese Änderung am Algorithmus beibehalten werden soll oder nicht.

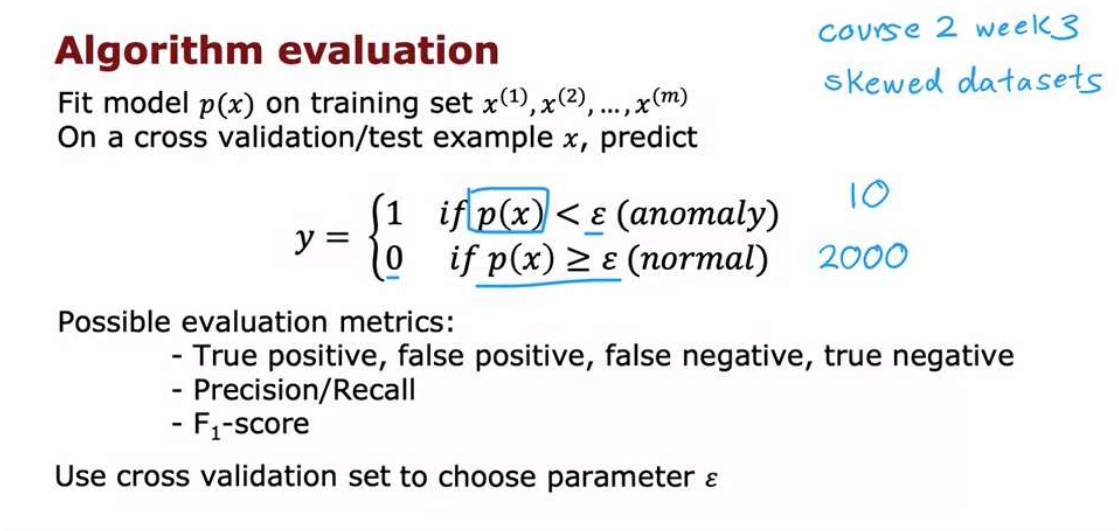


So wird es oft bei der Anomalie-Erkennung gemacht. Das heißt, auch wenn wir hauptsächlich über unbeschriftete Daten gesprochen haben, werde ich diese Annahme ein wenig ändern und davon ausgehen, dass wir einige beschriftete Daten haben, darunter nur eine kleine Anzahl normalerweise zuvor beobachteter Anomalien. Vielleicht haben Sie, nachdem Sie ein paar Jahre lang Flugzeugmotoren hergestellt haben, gerade ein paar Flugzeugmotoren gesehen, die anomal waren, und für Beispiele, von denen Sie wissen, dass sie anomal sind, werde ich eine Bezeichnung y gleich 1 zuordnen, um diese Anomalie anzuzeigen, und für Beispielen, die wir für normal halten, werde ich die Bezeichnung y gleich 0 zuordnen. Der Trainingssatz, aus dem der Anomalie-Erkennungsalgorithmus lernen wird, ist immer noch dieser unbeschriftete Trainingssatz von x1 bis xm, und ich werde an alles denken Wenn wir davon ausgehen, dass diese Beispiele normal und nicht anomal sind, ist y gleich 0. In der Praxis gibt es ein paar anomale Beispiele, bei denen Sie in diesen Trainingssatz einsteigen können, aber Ihr Algorithmus wird normalerweise immer noch gut funktionieren. Um Ihren Algorithmus zu evaluieren, überlegen Sie sich eine Möglichkeit, eine reelle Zahlenauswertung durchzuführen. Es erweist sich als sehr nützlich, wenn Sie eine kleine Anzahl anomaler Beispiele haben, damit Sie einen Kreuzvalidierungssatz erstellen können, den ich verwenden werde um x_cv^1, y_cv^1 bis x_cv^mcv, y_cv^mcv zu bezeichnen. Dies ist eine ähnliche Notation, wie Sie sie im zweiten Kurs dieser Spezialisierung gesehen haben. Ebenso sollten Sie einen Testsatz mit einigen Beispielen haben, wobei sowohl die Kreuzvalidierung als auch die Testsätze

hoffentlich ein paar anomale Beispiele enthalten. Mit anderen Worten: In den Kreuzvalidierungs- und Testsätzen gibt es einige Beispiele dafür, dass y gleich 1 ist, aber auch viele Beispiele, bei denen y gleich 0 ist. Auch hier funktioniert der Anomalie-Erkennungsalgorithmus in der Praxis einwandfrei, wenn es einige Beispiele dafür gibt sind tatsächlich anomal, wurden aber versehentlich mit y gleich 0 gekennzeichnet. Lassen Sie uns dies am Beispiel eines Flugzeugtriebwerks veranschaulichen. Nehmen wir an, Sie stellen seit Jahren Flugzeugtriebwerke her und haben daher Daten von 10.000 guten oder normalen Triebwerken gesammelt, aber im Laufe der Jahre haben Sie auch Daten von 20 fehlerhaften oder anormalen Triebwerken gesammelt. Normalerweise ist die Anzahl der anomalen Motoren, d. h. y gleich 1, viel kleiner. Es wird nicht üblich sein, diese Art von Algorithmus auf beispielsweise 2–50 bekannte Anomalien anzuwenden. Wir werden diesen Datensatz nehmen und ihn in einen Trainingssatz, einen Kreuzvalidierungssatz und den Testsatz aufteilen. Hier ist ein Beispiel. Ich werde 6.000 gute Motoren in das Trainingsset einbauen. Auch hier gilt: Wenn in diesem Set ein paar ungewöhnliche Motoren enthalten sind, ist das eigentlich in Ordnung, ich würde mir darüber keine allzu großen Sorgen machen. Dann fügen wir 2.000 gute Engines und 10 der bekannten Anomalien in den Kreuzvalidierungssatz ein und separate 2.000 gute und 10 anomale Engines in den Testsatz. Was Sie dann tun können, ist, den Algorithmus auf dem Trainingssatz zu trainieren, die Gaußschen Verteilungen an diese 6.000 Beispiele anzupassen und dann auf dem Kreuzvalidierungssatz zu sehen, wie viele der anomalen Engines er korrekt markiert. Beispielsweise könnten Sie den Kreuzvalidierungssatz verwenden, um den Parameter Epsilon zu optimieren und ihn höher oder niedriger festzulegen, je nachdem, ob der Algorithmus diese 10 Anomalien zuverlässig erkennt, ohne zu viele dieser 2.000 guten Engines zu nehmen und sie als Anomalien zu kennzeichnen. Nachdem Sie den Parameter Epsilon optimiert und möglicherweise auch die Funktionen X_J hinzugefügt, subtrahiert oder optimiert haben, können Sie den Algorithmus auf Ihrem Testsatz auswerten, um zu sehen, wie viele dieser 10 anomalen Engines er findet und wie viele Fehler er macht macht, indem es die guten Motoren als anomale markiert. Beachten Sie, dass es sich immer noch in erster Linie um einen unbeaufsichtigten Lernalgorithmus handelt, da die Trainingssätze tatsächlich keine Beschriftungen haben oder alle Beschriftungen haben, von denen wir annehmen, dass y gleich 0 ist. Daher haben wir aus dem Trainingssatz gelernt, indem wir die Gaußschen Verteilungen angepasst haben, wie Sie in gesehen haben das vorherige Video. Wenn Sie jedoch ein praktisches System zur Erkennung von Anomalien aufbauen, stellt sich heraus, dass eine kleine Anzahl von Anomalien zur Bewertung des Algorithmus in Ihren Kreuzvalidierungs- und Testsätzen sehr hilfreich für die Optimierung des Algorithmus ist.



Da die Anzahl der fehlerhaften Engines so gering ist, gibt es eine andere Alternative, die ich oft zur Anomalie-Erkennung sehe, nämlich keinen Testsatz zu verwenden, sondern nur einen Trainingssatz und einen Kreuzvalidierungssatz zu verwenden.



In diesem Beispiel werden Sie 6.000 gute Motoren trainieren, aber die restlichen Daten, die 4.000 verbleibenden guten Motoren sowie alle Anomalien nehmen und sie in den Kreuzvalidierungssatz einfügen. Anschließend würden Sie die Parameter Epsilon anpassen und fiktiv addieren oder subtrahieren x_j, um zu versuchen, dass es bei der Auswertung im Kreuzvalidierungssatz so gut wie möglich funktioniert. Wenn Sie nur sehr wenige fehlerhafte Engines haben, also nur zwei fehlerhafte Engines, dann ist es wirklich sinnvoll, all das in den Kreuzvalidierungssatz aufzunehmen. Sie verfügen einfach nicht über genügend Daten, um einen völlig separaten Testsatz zu erstellen, der sich von Ihrem Kreuzvalidierungssatz unterscheidet. Der Nachteil dieser Alternative besteht darin, dass Sie nach dem Optimieren Ihres Algorithmus keine faire Möglichkeit haben, zu sagen, wie gut dieser bei zukünftigen Beispielen tatsächlich funktionieren wird, da Sie nicht über den Testsatz verfügen. Wenn Ihr Datensatz jedoch klein ist, insbesondere wenn die Anzahl Ihrer Anomalien klein ist, ist dies möglicherweise die beste Alternative, die Sie haben. Ich sehe dies auch ziemlich oft, wenn Sie einfach nicht über genügend Daten verfügen, um einen separaten Testsatz zu erstellen. Wenn dies der Fall ist, müssen Sie sich darüber im Klaren sein, dass ein höheres Risiko besteht, dass Sie einige Ihrer Entscheidungen rund um Epsilon, die Auswahl der Funktionen usw. zu stark an den Kreuzvalidierungssatz und damit dessen Leistung bei realen Daten in der Zukunft anpassen ist möglicherweise nicht so gut, wie Sie erwartet haben.

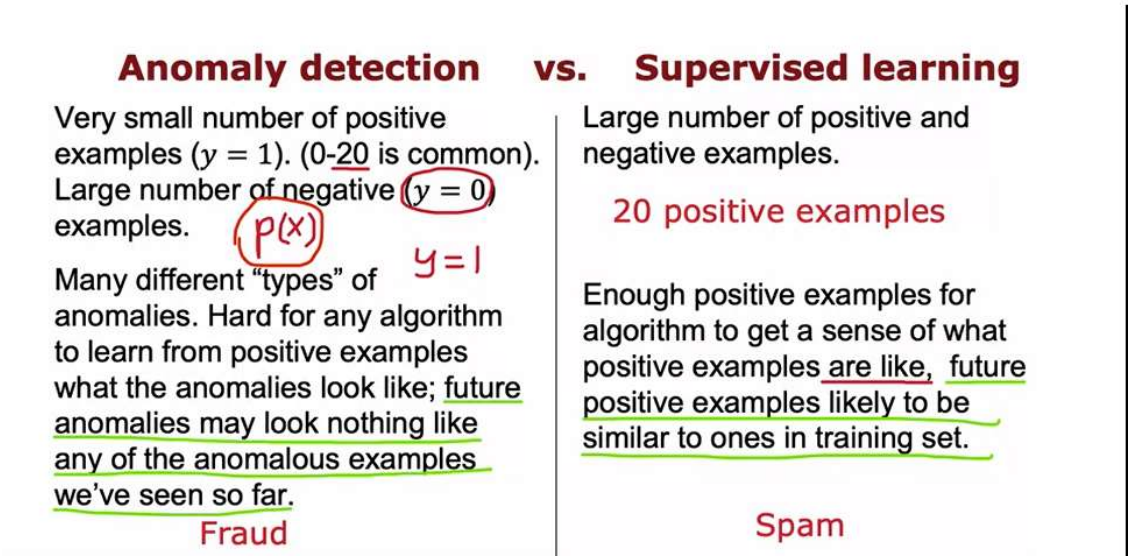
Schauen wir uns nun genauer an, wie Sie den Algorithmus in Ihren Kreuzvalidierungssätzen oder im Testsatz tatsächlich bewerten. Folgendes würden Sie tun. Sie würden zunächst das Modell p von x an den Trainingssatz anpassen. Dies waren 6.000 Exemplare von Güterlokomotiven. Dann würden Sie bei jeder Kreuzvalidierung oder jedem Testbeispiel x p von x berechnen und vorhersagen, dass y gleich 1 ist. Das ist anomal, wenn p von x kleiner als Epsilon ist und Sie y als 0 vorhersagen, wenn p von x größer als oder ist gleich Epsilon. Auf dieser Grundlage können Sie nun prüfen, wie genau die Vorhersagen dieses Algorithmus zur Kreuzvalidierung oder zum Testsatz mit den Beschriftungen übereinstimmen, die Sie in der Kreuzvalidierung oder den Testsätzen haben. In der dritten Woche des zweiten Kurses hatten wir einige optionale Videos zum Umgang mit stark verzerrten Datenverteilungen, bei denen die Anzahl der positiven Beispiele, y gleich 1, viel kleiner sein kann als die Anzahl der negativen Beispiele, bei denen y gleich 0 ist Dies gilt auch für viele Anomalie-

Erkennungen in Anwendungen, bei denen die Anzahl der Anomalien in Ihrem Kreuzvalidierungssatz viel geringer ist.

In unserem vorherigen Beispiel hatten wir vielleicht 10 positive Beispiele und 2.000 negative Beispiele, weil wir 10 Anomalien und 2.000 normale Beispiele hatten. Wenn Sie diese optionalen Videos gesehen haben, erinnern Sie sich vielleicht daran, dass wir gesehen haben, dass es nützlich sein kann, Dinge wie richtig positiv, falsch positiv, falsch negativ und wahr negativ [unverständlich] zu berechnen. Berechnen Sie außerdem den Präzisionsrückruf oder den F_1-Score und stellen Sie fest, dass es sich dabei um alternative Metriken und Klassifizierungsgenauigkeiten handelt, die besser funktionieren könnten, wenn Ihre Datenverteilung sehr verzerrt ist. Wenn Sie dieses Video gesehen haben, könnten Sie darüber nachdenken, auch diese Art von Bewertungsmetriken anzuwenden, um zu sehen, wie gut Ihr Lernalgorithmus darin ist, diese kleine Handvoll Anomalien oder positive Beispiele inmitten dieser viel größeren Menge negativer Beispiele normaler Flugzeugtriebwerke zu finden. Wenn Sie das Video nicht gesehen haben, machen Sie sich darüber keine Sorgen. Es ist okay. Die Intuition, die Sie hoffentlich bekommen, besteht darin, mithilfe des Kreuzvalidierungssatzes einfach zu prüfen, wie viele Anomalien gefunden werden und wie viele normale Engines fälschlicherweise als Anomalie gekennzeichnet werden. Dann versuchen wir einfach, eine gute Wahl für den Parameter Epsilon zu treffen. Sie werden feststellen, dass der praktische Prozess zum Aufbau eines Anomalie-Erkennungssystems viel einfacher ist, wenn Sie tatsächlich nur über eine kleine Anzahl gekennzeichneter Beispiele bekannter Anomalien verfügen. Dies wirft nun die Frage auf: Wenn Sie über ein paar beschriftete Beispiele verfügen und immer noch einen unbeaufsichtigten Lernalgorithmus verwenden, warum nehmen Sie dann nicht diese beschrifteten Beispiele und verwenden stattdessen einen überwachten Lernalgorithmus? Schauen wir uns im nächsten Video einen Vergleich zwischen Anomalie-Erkennung und überwachtem Lernen an und zeigen, wann Sie das eine dem anderen vorziehen könnten. Kommen wir zum nächsten Video.

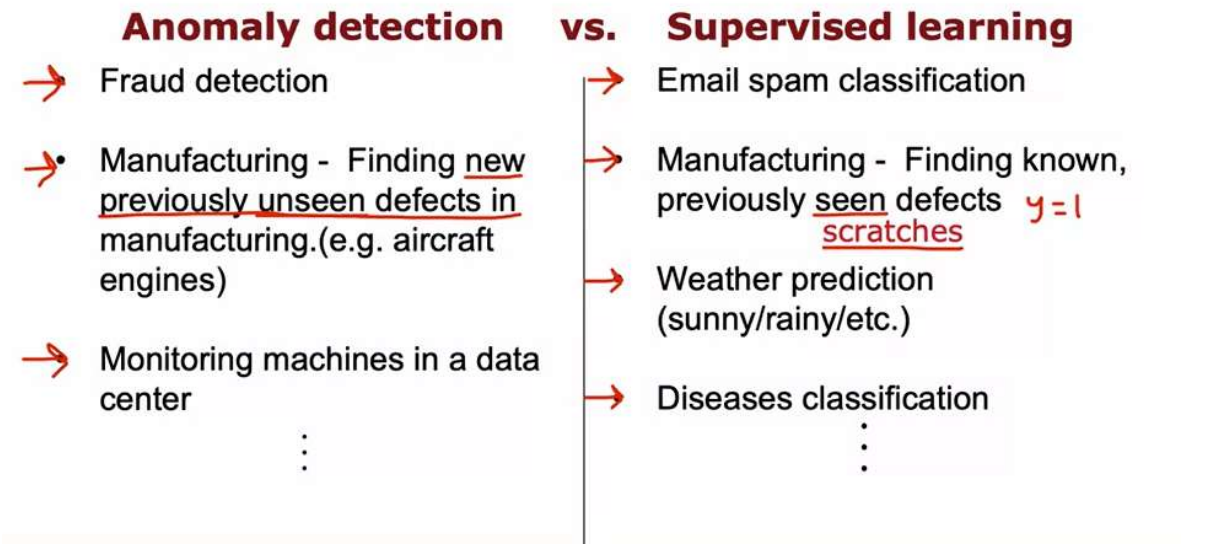
Anomalie-Erkennung vs. überwachtes Lernen

Wenn Sie ein paar positive Beispiele mit Michaels 1 und eine große Anzahl negativer Beispiele haben, sagen Sie y = 0? Wann sollten Sie die Anomalie-Erkennung und wann überwachtes Lernen verwenden? Die Entscheidung ist in manchen Anwendungen tatsächlich recht subtil. Lassen Sie mich einige Gedanken und Vorschläge mit Ihnen teilen, wie Sie zwischen diesen beiden Arten von Algorithmen wählen können. Und der Anomalie-Erkennungsalgorithmus ist in der Regel die geeignetere Wahl, wenn Sie über eine sehr kleine Anzahl positiver Beispiele verfügen; 0–20 positive Beispiele sind keine Seltenheit. Und eine relativ große Anzahl negativer Beispiele, mit denen man versuchen kann, ein Modell für p von x zu erstellen. Wenn Sie sich erinnern, dass die Parameter für p von x nur aus den negativen Beispielen gelernt werden und diese viel kleiner sind. Daher werden die positiven Beispiele nur in Ihrem Kreuzvalidierungssatz und Testsatz zur Parameteroptimierung und zur Auswertung verwendet. Wenn Sie hingegen über eine größere Anzahl positiver und negativer Beispiele verfügen, ist überwachtes Lernen möglicherweise besser anwendbar. Selbst wenn Sie nur 20 positive Trainingsbeispiele haben, könnte es in Ordnung sein, einen überwachten Lernalgorithmus anzuwenden. Es stellt sich jedoch heraus, dass die Art und Weise, wie die Anomalie-Erkennung den Datensatz betrachtet, im Vergleich zur Art und Weise, wie überwachtes Lernen den Datensatz betrachtet, ganz unterschiedlich ist. Sie ist der Hauptunterschied, der darin besteht, dass es Ihrer Meinung nach viele verschiedene Arten von offensichtlichen oder viele verschiedene Arten von positiven Beispielen gibt.



Dann könnte die Anomalie-Erkennung sinnvoller sein, wenn es viele verschiedene Möglichkeiten gibt, dass ein Flugzeugtriebwerk ausfällt. Und wenn es morgen vielleicht eine völlig neue Möglichkeit gibt, dass ein Flugzeugtriebwerk etwas falsch macht. Dann decken Ihre 20 positiven Beispiele möglicherweise nicht alle Möglichkeiten ab, wie ein Flugzeugtriebwerk kaputt gehen kann. Das macht es für jeden Algorithmus schwierig, aus der kleinen Menge positiver Beispiele zu lernen, was die Anomalien sind und wie die positiven Beispiele aussehen. Und zukünftige Anomalien sehen möglicherweise nicht wie die anomalen Beispiele aus, die wir bisher gesehen haben. Wenn Sie glauben, dass dies auf Ihr Problem zutrifft, würde ich zur Verwendung eines Anomalie-Erkennungsalgorithmus tendieren. Denn die Funktion der Anomalie-Erkennung besteht darin, die normalen Beispiele, d. h. die negativen Beispiele mit y = 0, zu betrachten und einfach zu modellieren, wie sie aussehen. Und alles, was stark vom Normalen abweicht, wird als Anomalie gekennzeichnet. Einschließlich der Frage, ob es eine völlig neue Möglichkeit für den Ausfall eines Flugzeugtriebwerks gibt, die in Ihrem Datensatz noch nie zuvor gesehen wurde. Im Gegensatz dazu wird das Problem beim überwachten Lernen anders betrachtet. Wenn Sie überwachtes Lernen anwenden, hoffen Sie im Idealfall, genügend positive Beispiele zu haben, damit der Durchschnitt ein Gefühl dafür bekommt, wie die positiven Beispiele aussehen. Und beim überwachten Lernen gehen wir tendenziell davon aus, dass die zukünftigen positiven Beispiele wahrscheinlich denen im Trainingsset ähneln. Lassen Sie mich dies anhand eines Beispiels veranschaulichen: Wenn Sie ein System verwenden, um beispielsweise Finanzbetrug aufzudecken. Leider gibt es viele verschiedene Möglichkeiten, wie manche Menschen versuchen, Finanzbetrug zu begehen. Und leider gibt es alle paar Monate oder jedes Jahr neue Arten von Finanzbetrugsversuchen. Und das bedeutet, dass sie immer wieder völlig neu auftauchen. Und einzigartige Formen der Erkennung von Finanzbetrugsanomalien werden häufig verwendet, um nach etwas zu suchen, das sich von den Transaktionen unterscheidet, die wir in der Vergangenheit gesehen haben. Betrachtet man dagegen das Problem der E-Mail-Spam-Erkennung, dann gibt es viele verschiedene Arten von Spam-E-Mails, allerdings auch über viele Jahre hinweg. Spam-E-Mails versuchen immer wieder, ähnliche Dinge zu verkaufen oder Sie dazu zu bringen, auf ähnliche Websites zu gehen und so weiter. Spam-E-Mails, die Sie in den nächsten Tagen erhalten, ähneln viel eher den Spam-E-Mails, die Sie in der Vergangenheit gesehen haben. Aus diesem Grund eignet sich überwachtes Lernen gut für Spam, da es versucht, mehr Arten von Spam-E-Mails zu erkennen, die Sie wahrscheinlich in der Vergangenheit in Ihrem Trainingssatz gesehen haben. Wenn Sie hingegen versuchen, völlig neue Arten von Betrug zu erkennen, die noch nie zuvor im Nominierungsbereich gesehen wurden, ist dies vielleicht besser geeignet. Gehen wir noch ein paar Beispiele durch. Wir haben bereits gesehen, dass die Betrugserkennung ein Anwendungsfall der Anomalie-Erkennung ist. Obwohl überwachtes Lernen

verwendet wird, um zuvor beobachtete Betrugsformen zu finden. Und wir haben gesehen, dass die E-Mail-Spam-Klassifizierung typischerweise mithilfe von überwachtem Lernen angegangen wird. Sie haben auch das Beispiel der Fertigung gesehen, bei der Sie möglicherweise neue, bisher nicht sichtbare Fehler finden möchten. Zum Beispiel, wenn es in der Zukunft völlig neue Möglichkeiten für den Ausfall eines Flugzeugtriebwerks gibt, die Sie noch erkennen möchten. Auch wenn Sie kein solches positives Beispiel in Ihrem Trainingssatz haben. Es stellt sich heraus, dass das fertigungsüberwachte Lernen auch zur Fehlersuche eingesetzt wird. Umso mehr, um bekannte und zuvor gesehene Mängel zu finden. Wenn Sie beispielsweise ein Smartphone-Hersteller sind, stellen Sie Mobiltelefone her. Und Sie wissen, dass Ihre Maschine zur Herstellung der Smartphone-Hülle gelegentlich versehentlich die Hülle dehnt. Dehnungen sind also ein häufiger Fehler bei Smartphones, und Sie können genügend Trainingsbeispiele für gestreckte Smartphones erhalten, die auf die Bezeichnung y = 1 reagieren. Und trainieren Sie einfach das System, um zu entscheiden, ob ein neues Smartphone, das Sie gerade hergestellt haben, Dehnungsstreifen aufweist. Und der Unterschied besteht darin, dass überwachtes Lernen gut funktioniert, wenn Sie immer wieder dehnbare Smartphones sehen und überprüfen möchten, ob Ihre Telefone gedehnt sind. Wenn Sie hingegen den Verdacht hegen, dass es sich hierbei um völlig neue Wege handelt, auf denen in Zukunft etwas schiefgehen kann, dann wird die Anomalie-Erkennung gut funktionieren. Einige andere Beispiele, die Sie von mir gehört haben, sind die Überwachung von Maschinen im Rechenzentrum, insbesondere die Maschine, die gehackt wurde. Es kann sich völlig anders verhalten als je zuvor in seinem Verhalten. Das würde sich also eher wie eine Anwendung zur Anomalie-Erkennung anfühlen. Ein Thema ist in der Tat, dass viele sicherheitsrelevante Anwendungen, weil Hacker oft ganz neue Wege finden, um in Systeme einzudringen.

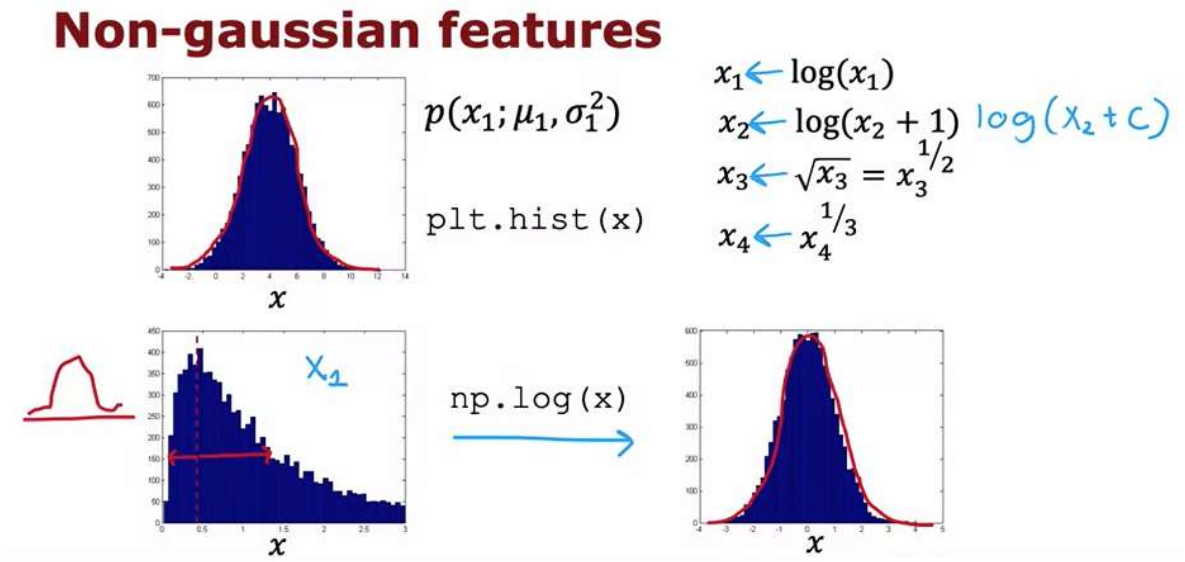


Viele sicherheitsrelevante Anwendungen nutzen die Anomalie-Erkennung. Zurück zum überwachten Lernen: Wenn Sie lernen möchten, das Wetter gut vorherzusagen, gibt es nur eine Handvoll Wetterarten, die Sie normalerweise sehen. Ist es sonnig, regnerisch, wird es schneien? Da Sie also immer wieder dieselben Ausgabebezeichnungen sehen, ist die Wettervorhersage in der Regel eine überwachte Lernaufgabe. Oder wenn Sie anhand der Symptome des Patienten feststellen möchten, ob der Patient eine bestimmte Krankheit hat, die Sie zuvor gesehen haben. Dann wäre das tendenziell auch eine überwachte Lernanwendung. Ich hoffe, das gibt Ihnen einen Rahmen für die Entscheidung, wann Sie über eine kleine Menge positiver Beispiele und vielleicht auch über eine große Menge negativer Beispiele verfügen. Ob Anomalie-Erkennung oder überwachtes Lernen. Die Anomalie-Erkennung versucht, brandneue positive Beispiele zu finden, die möglicherweise anders sind als alles, was Sie bisher gesehen haben. Beim überwachten Lernen werden Ihre positiven

Beispiele betrachtet und versucht zu entscheiden, ob ein zukünftiges Beispiel den positiven Beispielen ähnelt, die Sie bereits gesehen haben. Nun stellt sich heraus, dass beim Aufbau eines Anomalie-Erkennungsalgorithmus die Auswahl der Funktionen sehr wichtig ist und beim Aufbau von Anomalie-Erkennungssystemen. Im nächsten Video verbringe ich oft etwas Zeit damit, die Funktionen zu optimieren, die ich für das System verwende. Lassen Sie mich einige praktische Tipps geben, wie Sie die Funktionen, die Sie dem Anomalie-Erkennungsalgorithmus zuführen, optimieren können.

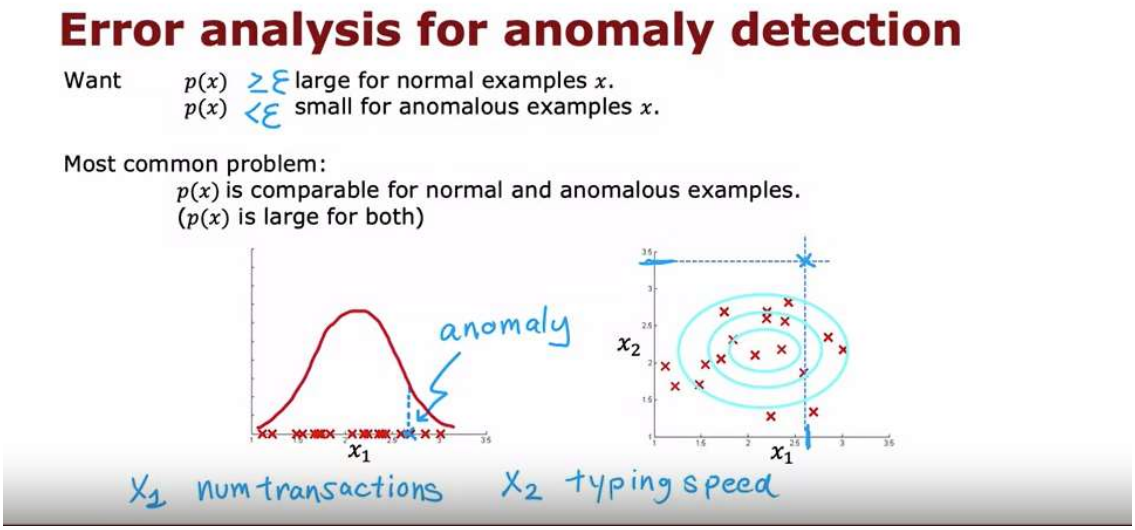
Auswählen der zu verwendenden Funktionen

Beim Erstellen eines Anomalie-Erkennungsalgorithmus habe ich festgestellt, dass die Auswahl einer guten Auswahl an Funktionen sehr wichtig ist. Wenn beim überwachten Lernen die Funktionen nicht ganz richtig sind oder wenn Sie über ein paar zusätzliche Funktionen verfügen, die für das Problem nicht relevant sind, ist das oft in Ordnung. Da der Algorithmus das Signal überwachen muss, gibt es genügend Beschriftungen, warum der Algorithmus herausfinden kann, welche Features ignoriert werden oder wie Features neu skaliert werden können, und um den größtmöglichen Nutzen aus den Features zu ziehen, die Sie ihm geben.



Aber bei einer Anomalie-Erkennung, die nur aus unbeschrifteten Daten läuft oder daraus lernt, ist es für die Anomalie schwieriger herauszufinden, welche Funktionen ignoriert werden sollen. Daher habe ich herausgefunden, dass die sorgfältige Auswahl der Funktionen für die Anomalie-Erkennung noch wichtiger ist als für überwachte Lernansätze. Schauen wir uns dieses Video an, um einige praktische Tipps zu erhalten, wie Sie die Funktionen zur Anomalie-Erkennung optimieren können, um die bestmögliche Leistung zu erzielen. Ein Schritt, der Ihrem Anomalie-Erkennungsalgorithmus helfen kann, besteht darin, sicherzustellen, dass die von Ihnen angegebenen Funktionen mehr oder weniger Gaußsch sind. Und wenn Ihre Merkmale nicht gaußförmig sind, können Sie sie manchmal ändern, um sie etwas gaußförmiger zu machen. Lassen Sie mich Ihnen zeigen, was ich meine. Wenn Sie ein Feature X haben, werde ich oft ein Gramm des Features zeichnen, was Sie mit dem Python-Befehl PLT tun können. Allerdings sehen Sie dies auch im Übungslabor, um sich den Verlauf der Daten anzusehen. Diese Verteilung hier sieht ziemlich gaußförmig aus. Das wäre also ein gutes Kandidatenfeature. Wenn Sie glauben, dass dies eine Funktion ist, die zwischen Anomalien und normalen Beispielen unterscheiden soll. Wenn Sie jedoch ein paar Gramm Ihrer Features grafisch darstellen, stellen Sie möglicherweise fest, dass das Feature eine solche Verteilung aufweist. Das sieht überhaupt nicht nach dieser symmetrischen glockenförmigen Kurve aus. Wenn das der Fall ist,

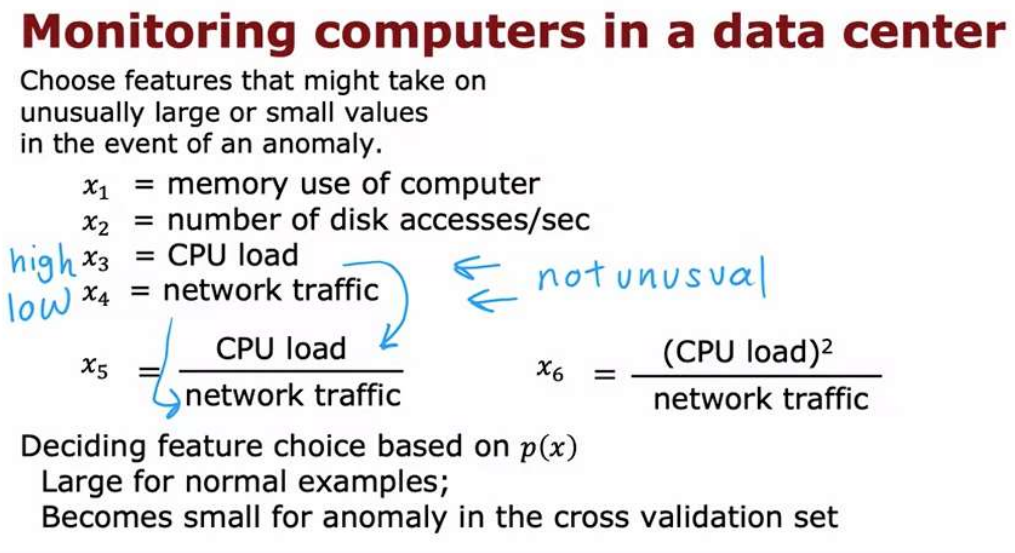
würde ich darüber nachdenken, ob Sie dieses Merkmal X nehmen und es transformieren können, um eine Gaußschere Funktion zu erhalten. Wenn Sie zum Beispiel den Logarithmus von Wenn es sich bei diesem Feature also um Feature Denn wenn X eins wird, wird es gaußförmiger. Wenn Anomalie-Erkennungsmodelle P von Abgesehen von der Protokollfunktion können Sie bei einem gegebenen anderen Merkmal X zwei auch Folgendes tun: Sie können es durch X zwei, Protokoll von X zwei plus eins, ersetzen. Dies wäre eine andere Art der Transformation von X zwei. Und allgemeiner gesagt wäre der Logarithmus von Oder für ein anderes Merkmal könnten Sie versuchen, die Quadratwurzel zu ziehen, sonst hätte das Quadrat diese X-Ableitung zur Potenz einer Hälfte ausgeführt, und Sie könnten diesen exponentiellen Term ändern. Für ein anderes Merkmal X vier könnten Sie beispielsweise X vier hoch ein Drittel verwenden.



Wenn ich also ein Anomalie-Erkennungssystem aufbaue, schaue ich mir manchmal meine Funktionen an, und wenn ich bei der grafischen Darstellung eines Gramms einen stark nicht-Gaußschen Wert entdecke, wähle ich möglicherweise Transformationen wie diese oder andere aus, um dies zu versuchen Machen Sie es gaußscher. Es stellt sich heraus, dass ein größerer Wert von C diese Verteilung letztendlich weniger transformiert. Aber in der Praxis probiere ich einfach eine Reihe verschiedener Werte von C aus und versuche dann, einen auszuwählen, der im Hinblick auf eine Gauß-Verteilung besser aussieht. Lassen Sie mich nun veranschaulichen, wie ich das tatsächlich mache und dass Sie ein Notizbuch hineinlegen. So könnte also der Prozess der Erkundung verschiedener Transformationen in den Features aussehen. Wenn Sie ein Merkmal X haben, können Sie ein Gramm davon wie folgt darstellen. Es sieht tatsächlich so aus, als gäbe es einen schönen Grund dafür, dass ein Gramm gezischt wurde. Lassen Sie mich die Anzahl der Bins in meinem Verlaufsgramm auf 50 erhöhen. Die Bins entsprechen dort also 50. Das ist es, was ein Gramm Mülleimer zischte. Und wenn Sie die Farbe übrigens ändern möchten, können Sie dies auch wie folgt tun. Und wenn Sie eine andere Transformation ausprobieren möchten, können Sie beispielsweise versuchen, die Quadratwurzel von Und das sieht tatsächlich etwas gaußscher aus. Aber nicht perfekt, und versuchen wir es mit einem anderen Parameter. Lassen Sie es mich also mit der Potenz 4,25 versuchen. Vielleicht bin ich einfach ein bisschen zu weit gegangen. Es ist das alte 0,4, das ziemlich Gaußsch aussieht.

Sie könnten also X durch eine hervorragende Potenz von 0,4 ersetzen. Sie würden also X gleich X hoch 0,4 setzen. Und verwenden Sie stattdessen einfach den Wert von X in Ihrem Trainingsprozess. Oder lassen Sie mich Ihnen eine andere Transformation zeigen. Hier werde ich versuchen, das Protokoll von Und es stellt sich heraus, dass Sie einen Fehler erhalten, weil es so ausgezeichnet ist. In diesem Beispiel gibt es einige Werte, die gleich Null sind, und wir protokollieren, dass Null negativ ist.

Unendlich ist nicht definiert. Ein üblicher Trick besteht darin, dort nur eine sehr kleine Zahl hinzuzufügen. Der Export beträgt also 0,001 und wird nicht negativ. Und so erhalten Sie das gezischte Gramm, das so aussieht. Wenn Sie jedoch möchten, dass die Verteilung eher gaußförmig aussieht, können Sie auch mit diesem Parameter experimentieren, um herauszufinden, ob es dafür einen Wert gibt. Bewirken Sie, dass Benutzerdaten wie folgt symmetrischer und möglicherweise gaußscher aussehen. Und genau wie ich es gerade in Echtzeit mache, können Sie sehen, dass Sie diese Parameter sehr schnell ändern und das gezischte Gramm grafisch darstellen können. Um einen Blick darauf zu werfen und zu versuchen, etwas zu erhalten, das etwas gaußscher ist als die Originaldaten, die Sie oben in diesem gezischten Gramm gesehen haben. Wenn Sie die Literatur zum maschinellen Lernen lesen, gibt es einige Möglichkeiten, automatisch zu messen, wie nahe diese Verteilungen an der Gaußschen Verteilung liegen. Aber ich habe festgestellt, dass es in der Praxis keinen großen Unterschied macht, wenn man einfach ein paar Werte ausprobiert und etwas auswählt, das für einen richtig erscheint, das für alle praktischen Zwecke gut funktioniert. Wenn Sie also Dinge im Jupiter-Notizbuch ausprobieren, können Sie versuchen, eine Transformation auszuwählen, die Ihre Daten gaußscher macht. Und zur Erinnerung: Unabhängig davon, welche Transformation Sie auf den Trainingssatz anwenden, denken Sie bitte daran, dieselbe Transformation auch auf Ihre Kreuzvalidierungs- und Testsatzdaten anzuwenden. Nachdem Sie Ihren Anomalie-Erkennungsalgorithmus trainiert haben, stellen Sie nicht nur sicher, dass Ihre Daten annähernd Gaußsch sind, sondern können auch einen Fehleranalyseprozess zur Anomalie-Erkennung durchführen, wenn dieser in Ihrem Vertrauensvalidierungssatz nicht so gut funktioniert. Mit anderen Worten: Sie können versuchen, zu prüfen, wo der Algorithmus noch nicht gut abschneidet, während er Fehler macht, und dann versuchen, Verbesserungen zu erzielen. Zur Erinnerung: Wir wollen, dass P von X groß ist. Für normale Beispiele ist X, also größer als gleich Epsilon, und pf X muss klein oder kleiner als Epsilon sein, für die anomalen Beispiele X. Wenn Sie das Modell P von X aus Ihren unbeschrifteten Daten gelernt haben, ist das häufigste Problem, das Sie haben Es kann darauf stoßen, dass P von Als konkretes Beispiel: Wenn dies Ihr Datensatz ist, könnten Sie diese Galaxie darin einfügen. Und wenn Sie in Ihrem Kreuzvalidierungssatz oder Testsatz ein Beispiel haben, das hier drüben ist und anomal ist, dann ist dies mit ziemlich hoher Wahrscheinlichkeit der Fall. Und tatsächlich sieht es den anderen Beispielen in Ihrem Trainingssatz ziemlich ähnlich. Und obwohl dies eine Anomalie ist, ist P von X tatsächlich ziemlich groß. Daher wird der Algorithmus dieses spezielle Beispiel nicht als Anomalie kennzeichnen. In diesem Fall würde ich normalerweise versuchen, mir dieses Beispiel anzusehen und herauszufinden, was mich zu der Annahme veranlasst hat, dass es sich um eine Anomalie handelt, selbst wenn dieses Feature X One ähnliche Werte wie andere Trainingsbeispiele annehmen würde. Und wenn ich eine neue Funktion identifizieren kann, sagen wir X zwei, hilft das, dieses Beispiel von den normalen Beispielen zu unterscheiden. Das Hinzufügen dieser Funktion kann dann dazu beitragen, die Leistung des Algorithmus zu verbessern. Hier ist ein Bild, das zeigt, was ich meine. Wenn ich mir eine neue Funktion einfallen lassen kann, sagen wir: „Ich versuche, betrügerisches Verhalten zu erkennen “, und wenn anders. Aber wenn ich entdecke, dass dieser Benutzer eine wahnsinnig schnelle Tippgeschwindigkeit hat, und wenn ich eine neue Funktion X zwei hinzufügen würde, dann wäre das die Tippgeschwindigkeit dieses Benutzers. Und wenn sich herausstellt, dass, wenn ich diese Daten mit der alten Funktion X eins und dieser neuen Funktion X zwei zeichne, X zwei hier hervorsticht. Dann wird es für den Anomalie-Erkennungsalgorithmus viel einfacher, zu erkennen, dass es sich bei X Two um einen anomalen Benutzer handelt. Denn wenn Sie dieses neue Merkmal X zwei haben, passt die Lernanomalie möglicherweise zu einer Gaußschen Verteilung, die Punkten in dieser Region eine hohe Wahrscheinlichkeit zuweist, in dieser Region etwas niedriger und in dieser Region etwas niedriger. Und so lässt sich dieses Beispiel aufgrund des sehr anomalen Werts von X zwei leichter als Anomalie erkennen. Zusammenfassend lässt sich sagen, dass der Entwicklungsprozess oft durchlaufen wird: Das Modell trainieren und dann sehen, welche Anomalien im Kreuzvalidierungssatz der Algorithmus nicht erkennen kann.



Und dann schauen wir uns diese Beispiele an, um zu sehen, ob sie die Entwicklung neuer Funktionen inspirieren können, die es dem Algorithmus ermöglichen würden, etwas zu erkennen. Dieses Beispiel nimmt bei den neuen Funktionen ungewöhnlich große oder ungewöhnlich kleine Werte an, sodass Sie diese Beispiele nun erfolgreich als Anomalien kennzeichnen können. Nehmen wir als weiteres Beispiel an, Sie bauen ein Anomalie-Erkennungssystem zur Überwachung von Computern im Rechenzentrum auf. Versuchen Sie herauszufinden, ob sich ein Computer möglicherweise seltsam verhält und eine genauere Betrachtung verdient, möglicherweise aufgrund eines Hardwarefehlers oder weil er gehackt wurde oder so etwas. Sie möchten also Funktionen auswählen, die im Falle einer Anomalie ungewöhnlich große oder kleine Werte annehmen könnten. Sie können mit Funktionen beginnen, z. B. X eins ist die Speichernutzung, X zwei ist die Anzahl der Festplattenzugriffe pro Sekunde, dann die CPU-Auslastung und das Volumen des Netzwerkverkehrs. Und wenn Sie den Algorithmus trainieren, stellen Sie möglicherweise fest, dass er einige Anomalien erkennt, andere jedoch nicht. In diesem Fall ist es nicht ungewöhnlich, neue Funktionen durch die Kombination alter Funktionen zu erstellen. Wenn Sie beispielsweise feststellen, dass sich ein Computer sehr seltsam verhält, die CPU-Auslastung und der Netzwerkverkehr jedoch nicht so ungewöhnlich sind. Was jedoch ungewöhnlich ist, ist die sehr hohe CPU-Auslastung bei gleichzeitig sehr geringem Netzwerkverkehrsaufkommen. Wenn Sie das Rechenzentrum betreiben, das Videos streamt, können die Computer eine hohe CPU-Auslastung und hohen Netzwerkverkehr oder eine niedrige CPU-Auslastung und keinen Netzwerkverkehr aufweisen. Das Ungewöhnliche an dieser einen Maschine ist jedoch eine sehr hohe CPU-Auslastung trotz sehr geringem Verkehrsaufkommen. In diesem Fall könnten Sie ein neues Feature X five erstellen, das ein Verhältnis von CPU-Auslastung zu Netzwerkverkehr darstellt. Und diese neue Funktion mit Hoffnung: Der Anomalie-Erkennungsalgorithmus markiert zukünftige Beispiele wie die spezifische Maschine, die Sie möglicherweise als anomal ansehen. Sie können auch andere Merkmale berücksichtigen, z. B. das Quadrat der CPU-Auslastung geteilt durch das Netzwerkverkehrsvolumen. Und Sie können mit verschiedenen Optionen dieser Funktionen herumspielen. Um zu versuchen, dass P von X für die normalen Beispiele immer noch groß ist, bei den Anomalien in Ihrem Kreuzvalidierungssatz jedoch klein wird. Das war's. Vielen Dank, dass Sie bis zum Ende dieser Woche bei mir geblieben sind. Ich wünsche Ihnen viel Spaß beim Hören sowohl über Clustering-Algorithmen als auch über Anomalie-Erkennungs-Algorithmen. Und dass es Ihnen auch Spaß macht, in den Übungslaboren mit diesen Ideen zu experimentieren. Nächste Woche werden wir über Empfehlungssysteme sprechen. Wenn Sie eine Website besuchen und Ihnen Produkte, Filme oder andere Dinge empfehlen. Wie funktioniert dieser Algorithmus eigentlich? Dies ist einer der kommerziell wichtigsten Algorithmen im maschinellen Lernen, über den überraschend wenig gesprochen wird. Nächste Woche werfen wir

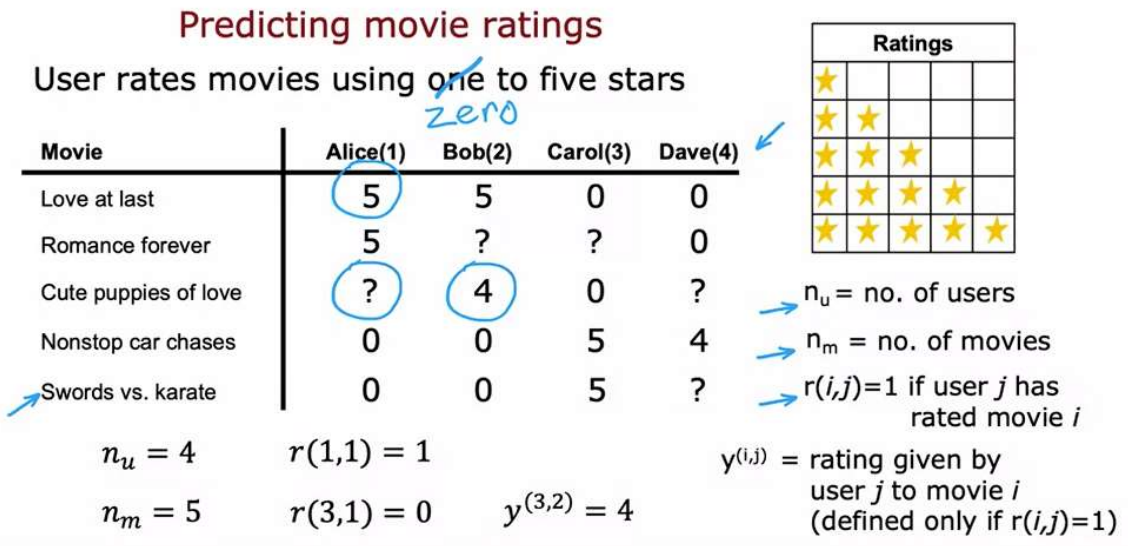
jedoch einen Blick auf die Funktionsweise dieser Algorithmen, damit Sie sie verstehen, wenn Sie das nächste Mal auf die Website gehen, und Ihnen dann etwas empfehlen können. Vielleicht, wie das zustande kam. Ebenso können Sie auch andere Algorithmen dieser Art für sich selbst erstellen. Viel Spaß mit den Laboren und sie freuen sich darauf, Sie nächste Woche zu sehen.

Empfehlungs-Systeme

Lernziele

- Implementieren Sie kollaborative Filterempfehlungssysteme in TensorFlow
- Implementieren Sie eine inhaltsbasierte Deep-Learning-Filterung mithilfe eines neuronalen Netzwerks in TensorFlow
- Verstehen Sie ethische Überlegungen beim Aufbau von Empfehlungssystemen

Willkommen zu dieser vorletzten Woche der Spezialisierung auf maschinelles Lernen. Ich bin wirklich glücklich, dass wir fast bis zur Ziellinie zusammen waren. Was wir diese Woche tun werden, ist die Diskussion empfohlener Systeme. Dies ist eines der Themen, die in der Wissenschaft große Aufmerksamkeit erhalten haben. Aber die kommerziellen Auswirkungen und die tatsächliche Anzahl praktischer Anwendungsfälle empfohlener Systeme scheinen mir noch weitaus größer zu sein als die Aufmerksamkeit, die ihnen in der Wissenschaft zu Teil wird. Jedes Mal, wenn Sie eine Online-Shopping-Website wie Amazon oder eine Film-Streaming-Website wie Netflix oder eine der Apps oder Websites besuchen, die Essenslieferungen anbieten. Viele dieser Websites empfehlen Ihnen Dinge, von denen sie glauben, dass Sie sie kaufen möchten, Filme, von denen sie glauben, dass Sie sie sich ansehen möchten, oder Restaurants, von denen sie glauben, dass Sie sie ausprobieren möchten.



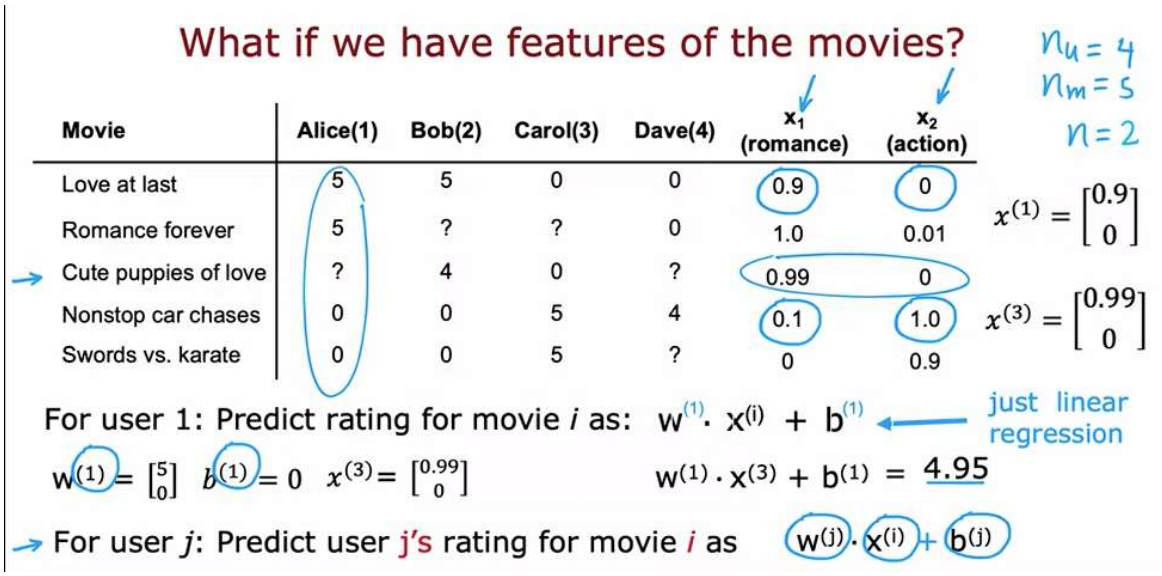
Heutzutage ist für viele Unternehmen die Wirtschaftlichkeit oder der Wert, der durch empfohlene Systeme erzielt wird, sehr groß, und deshalb werfen wir diese Woche einen Blick darauf, wie sie funktionieren. Lassen Sie uns also einen Blick darauf werfen, welches System empfohlen wird. Ich werde als laufendes Beispiel die Anwendung der Vorhersage von Filmbewertungen verwenden. Angenommen, Sie betreiben eine große Film-Streaming-Website und Ihre Benutzer haben Filme mit einem bis fünf Sternen bewertet. In einem typischen empfohlenen System gibt es also eine Gruppe von Benutzern, hier haben wir vier Benutzer: Alice, Bob Carol und Dave. Welche Benutzer nummeriert haben 1,2,3,4. Sowie eine Reihe von Filmen „Love at last“, „Romance Forever“, „ Cute

Puppies of Love“ und dann „Nonstop Car Chases“ und „Sword versus Karate“. Und was die Benutzer getan haben, ist, diese Filme mit einem bis fünf Sternen zu bewerten. Oder um einige dieser Beispiele etwas einfacher zu machen. Ich werde nicht zulassen, dass sie die Filme mit null bis fünf Sternen bewerten. Angenommen, Alice hat „Love and Last“ mit fünf Sternen bewertet, „Romance Forever“ mit fünf Sternen. Vielleicht hat sie sich die süßen Welpen der Liebe noch nicht angesehen, daher gibt es dafür keine Bewertung. Und ich werde das durch ein Fragezeichen kennzeichnen, und sie ist der Meinung, dass ununterbrochene Verfolgungsjagden und Schwert gegen Karate null Sterne verdienen. Rennen bei fünf Sternen habe ich nicht gesehen, daher gibt es bei Rennen bei vier Sternen keine Bewertung, 0,0. Carol hingegen, Dinge, die null Sterne verdienen, hat diese null Sterne nicht gesehen und sie liebt ununterbrochene Verfolgungsjagden und Schwertkämpfe gegen Karate, und Dave hebt die Filme wie folgt hervor. Im typischen empfohlenen System gibt es sowohl eine bestimmte Anzahl von Benutzern als auch eine bestimmte Anzahl von Elementen. In diesem Fall handelt es sich bei den Artikeln um Filme, die Sie den Benutzern empfehlen möchten. Und obwohl ich in diesem Beispiel Filme verwende, ist es die gleiche Logik oder das Gleiche. Arbeiten Sie daran, dem Benutzer alles zu empfehlen, von Produkten oder der Website selbst bis hin zu Restaurants, bis hin zu den Medienartikeln, den Social-Media-Artikeln, die dem Benutzer angezeigt werden sollen, was für ihn möglicherweise interessanter ist. Die Notation, die ich verwenden werde, ist, dass ich nu verwende, um die Anzahl der Benutzer anzugeben. In diesem Beispiel ist nu also gleich vier, weil Sie vier Benutzer haben, und nm, um die Anzahl der Filme oder eigentlich die Anzahl der Elemente anzugeben. In diesem Beispiel ist nm also gleich fünf, weil wir fünf Filme haben. Ich werde r(i,j)=1 setzen, wenn Benutzer j den Film i bewertet hat. Verwenden Sie also zum Beispiel eine Eins, bei der Dallas Alice den ersten Film, aber den dritten Film nicht bewertet hat, also r(1,1) = 1, weil sie den ersten Film bewertet hat, aber r(3,1)=0, weil sie dies nicht getan hat bewerteter Film Nummer drei. Dann werde ich endlich y(i,j) verwenden. J bezeichnet die Bewertung, die Benutzer j dem Film i gegeben hat. Diese Bewertung hier wäre beispielsweise, dass Film drei von Benutzer 2 mit vier bewertet wurde. Beachten Sie, dass nicht jeder Benutzer jeden Film bewertet und es für das System wichtig ist zu wissen, welche Benutzer welche Filme bewertet haben. Aus diesem Grund definieren wir r(i,j)=1, wenn Benutzer j den Film i bewertet hat, und gleich Null, wenn Benutzer j den Film i nicht bewertet hat. Mit diesem Rahmen für empfohlene Systeme besteht eine Möglichkeit, das Problem anzugehen, darin, sich die Filme anzusehen, die Benutzer nicht bewertet haben. Und wir versuchen vorherzusagen, wie Benutzer diese Filme bewerten würden, denn dann können wir versuchen, Benutzern Dinge zu empfehlen, die sie eher mit fünf Sternen bewerten würden. Und im nächsten Video beginnen wir mit der Entwicklung eines Algorithmus, um genau das zu tun. Aber ich gehe von einer ganz besonderen Annahme aus. Das heißt, wir gehen vorübergehend davon aus, dass wir Zugriff auf Funktionen oder zusätzliche Informationen zu den Filmen haben, z. B. welche Filme Liebesfilme und welche Filme Actionfilme sind. Und damit wird die Entwicklung eines Algorithmus beginnen. Aber später in dieser Woche werde ich tatsächlich zurückkommen und fragen: Wie kann man den Algorithmus trotzdem zum Laufen bringen, wenn wir diese Funktionen nicht haben? Aber fahren wir mit dem nächsten Video fort, um mit dem Aufbau dieses Algorithmus zu beginnen.

Verwendung von Eigenschaften pro Einheit

Schauen wir uns also an, wie wir ein empfohlenes System entwickeln können, wenn wir Features für jedes Element oder Features für jeden Film hätten. Hier ist also derselbe Datensatz, den wir zuvor hatten, wobei die vier Benutzer einige, aber nicht alle der fünf Filme bewertet haben. Was wäre, wenn wir zusätzlich Features der Filme hätten? Deshalb habe ich hier zwei Features X1 und X2 hinzugefügt, die uns sagen, wie sehr jeder dieser Filme ein Liebesfilm und wie sehr jeder dieser Filme ein Actionfilm ist. So ist „Love at Last“ zum Beispiel ein sehr romantischer Film, daher liegt dieser Spielfilm bei 0,9, aber es handelt sich überhaupt nicht um einen Actionfilm. Diese Funktion nimmt also den Wert 0 an. Es stellt sich jedoch heraus, dass „Verfolgungsjagden ohne Unterbrechung“ nur

ein wenig Romantik enthält. Es ist also 0,1, aber es hat jede Menge Action. Dieses Feature nimmt also den Wert 1,0 an.

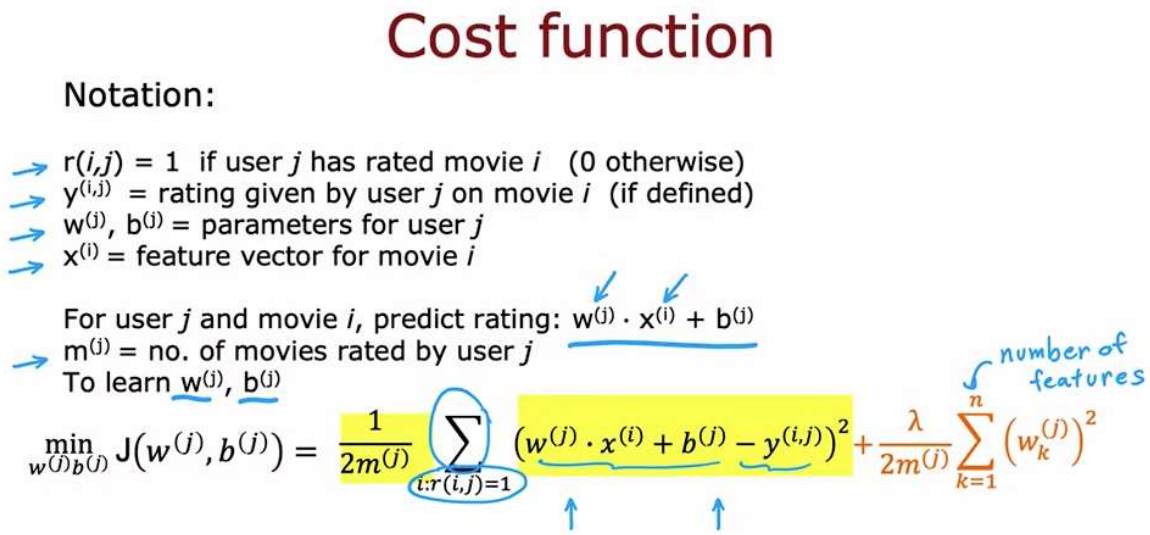


Sie erinnern sich also, dass ich die Notation nu verwendet habe, um die Anzahl der Benutzer anzugeben, also 4, und m, um die Anzahl der Filme anzugeben, also 5. Ich werde auch n einführen, um die Anzahl der Funktionen anzugeben, die wir hier haben. Also n=2, weil wir für jeden Film zwei Features X1 und X2 haben. Mit diesen Features haben wir zum Beispiel, dass die Features für Film eins, also den Film Love at Last, 0,90 betragen würden. Und die Features für den dritten Film „Cute Puppies of Love“ wären 0,99 und 0. Und werfen wir zunächst einen Blick darauf, wie wir Vorhersagen für Alices Filmbewertungen treffen könnten. Nehmen wir also an, dass wir für Benutzer eins, also Alice, die Bewertung für Film i als wX (i)+b vorhersagen. Das ähnelt also stark der linearen Regression. Wenn wir zum Beispiel am Ende den Parameter w(1)=[5,0] wählen und b(1)=0 sagen, dann wird zuerst die Vorhersage für Film drei mit den Merkmalen 0,99 und 0 erstellt, die einfach von hier kopiert wird Merkmal 0,99, zweites Merkmal 0. Unsere Vorhersage wäre wX (3)+b=0,99 mal 5 plus 0 mal Null, was sich als gleich 4,95 herausstellt. Und diese Bewertung scheint ziemlich plausibel. Es sieht so aus, als hätte Alice „Love at Last“ und „Romance Forever“, zwei sehr romantische Filme, hohe Bewertungen gegeben, den Actionfilmen „Nonstop Car Chases“ und „Swords vs. Karate“ hingegen niedrige Bewertungen. Wenn wir uns also „Cute Puppies of Love“ ansehen, erscheint die Vorhersage, dass sie diese 4,95 bewerten könnte, durchaus plausibel. Daher scheinen diese Parameter w und b für Alice ein vernünftiges Modell zur Vorhersage ihrer Filmbewertungen zu sein. Fügen Sie einfach ein wenig die Notation hinzu, denn wir haben nicht nur einen Benutzer, sondern mehrere Benutzer, oder nu entspricht 4 Benutzern. Ich füge hier eine hochgestellte 1 hinzu, um anzuzeigen, dass dies der Parameter w(1) für Benutzer 1 ist, und füge dort auch eine hochgestellte 1 hinzu. Und ähnlich auch hier und hier, so dass wir tatsächlich für jeden der 4 Benutzer im Datensatz unterschiedliche Parameter hätten. Und allgemeiner gesagt können wir in diesem Modell für Benutzer j, nicht nur Benutzer 1, die Bewertung von Benutzer j für Film i als w(j).X(i)+b(j) vorhersagen. Hier sind also die Parameter w(j) und b(j) die Parameter, die verwendet werden, um die Bewertung von Benutzer j für Film i vorherzusagen, die eine Funktion von X(i) ist, also den Merkmalen von Film i. Und das ähnelt stark der linearen Regression, außer dass wir für jeden der vier Benutzer im Datensatz ein anderes lineares Regressionsmodell anpassen.

Schauen wir uns also an, wie wir die Kostenfunktion für diesen Algorithmus formulieren können. Zur Erinnerung: r(i.,j)=1, wenn Benutzer j den Film i bewertet hat, andernfalls 0. Und y(i,j)=Bewertung

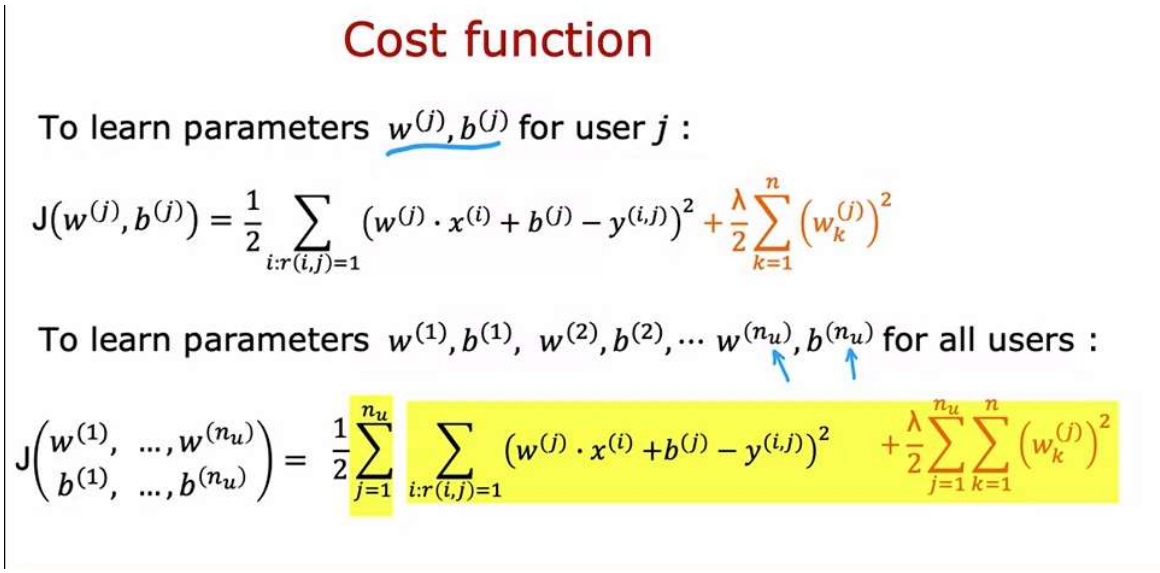
durch Benutzer j für Film i. Und auf der vorherigen Seite haben wir w(j), b(j) als Parameter für Benutzer j definiert. Und X(i) als Merkmalsvektor für Film i. Das Modell, das wir haben, gilt also für Benutzer j und Film i und sagt voraus, dass die Bewertung w(j).X(i)+b(j) ist. Ich werde nur eine neue Notation einführen: Ich werde m(j) verwenden, um die Anzahl der von Benutzer j bewerteten Filme anzugeben. Wenn der Benutzer also 4 Filme bewertet hat, wäre m(j) gleich 4. Und wenn der Benutzer 3 Filme bewertet hat, wäre m(j) gleich 3. Was wir also tun möchten, ist zu lernen die Parameter w(j) und b(j), angesichts der Daten, die wir haben. Dabei handelt es sich um die Bewertungen, die ein Benutzer für eine Reihe von Filmen abgegeben hat.

Der Durchschnitt, den wir verwenden werden, ist also der linearen Regression sehr ähnlich. Schreiben wir also die Kostenfunktion zum Lernen der Parameter w(j) und b(j) für einen gegebenen Benutzer j auf.



Konzentrieren wir uns vorerst nur auf einen Benutzer von Benutzer j. Ich werde das Kriterium des mittleren quadratischen Fehlers verwenden. Die Kosten entsprechen also der Vorhersage, also w(j).X(i)+b(j) minus der tatsächlichen Bewertung, die der Benutzer abgegeben hat. Also minus y(i,j) zum Quadrat. Und wir versuchen, die Parameter w und b so zu wählen, dass der quadratische Fehler zwischen der vorhergesagten Bewertung und der tatsächlich beobachteten Bewertung minimiert wird. Aber der Benutzer hat nicht alle Filme bewertet. Wenn wir also darüber summieren, werden wir nur über die Werte von i summieren, wobei r(i,j)=1 ist. Daher werden wir nur die Filme i zusammenfassen, die Benutzer j tatsächlich bewertet hat. Das ist also die Summe aller Werte von i, wobei r(i,j)=1 ist. Das bedeutet, dass Benutzer j diesen Film i bewertet hat. Und schließlich können wir die übliche Normalisierung 1 über m(j) vornehmen. Und das ähnelt stark der Kostenfunktion, die wir für die lineare Regression mit m oder eigentlich m(j) Trainingsbeispielen haben. Dabei summieren Sie über die m(j) Filme, für die Sie eine Bewertung haben, nehmen einen quadrierten Fehler und die Normalisierung durch diese 1 über 2m(j). Und das wird eine Kostenfunktion J von w(j), b(j) sein. Und wenn wir dies als Funktion von w(j) und b(j) minimieren, dann sollten Sie eine ziemlich gute Auswahl an Parametern w(i) und b(j) finden. Zum Erstellen von Vorhersagen für die Bewertungen von Benutzer j. Lassen Sie mich dieser Kostenfunktion noch einen Term hinzufügen, nämlich den Regularisierungsterm, um eine Überanpassung zu verhindern. Und hier ist unser üblicher Regularisierungsparameter: Lambda dividiert durch 2m(j) und dann multipliziert mit der Summe der quadrierten Werte der Parameter w. Und so ist n eine Anzahl von Zahlen in X(i) und das ist dasselbe wie eine Anzahl von Zahlen in w(j). Wenn Sie diese Kostenfunktion J als Funktion von w und b minimieren würden, sollten Sie einen ziemlich guten Parametersatz erhalten, um die

Bewertungen von Benutzer j für andere Filme vorherzusagen. Bevor wir nun fortfahren, stellt sich heraus, dass es für empfohlene Systeme zweckmäßig wäre, diese Division durch den Term m(j) tatsächlich zu eliminieren, da m(j) in diesem Ausdruck nur eine Konstante ist. Und selbst wenn Sie es herausnehmen, sollten Sie am Ende den gleichen Wert von w und b erhalten. Lassen Sie mich nun diese Kostenfunktion hier nach unten bringen und auf die nächste Folie kopieren. Das haben wir also, um die Parameter w(j), b(j) für Benutzer j zu lernen. Wir würden diese Kostenfunktion als Funktion von w(j) und b(j) minimieren. Aber anstatt uns auf einen einzelnen Benutzer zu konzentrieren, schauen wir uns an, wie wir die Parameter für alle Benutzer lernen.

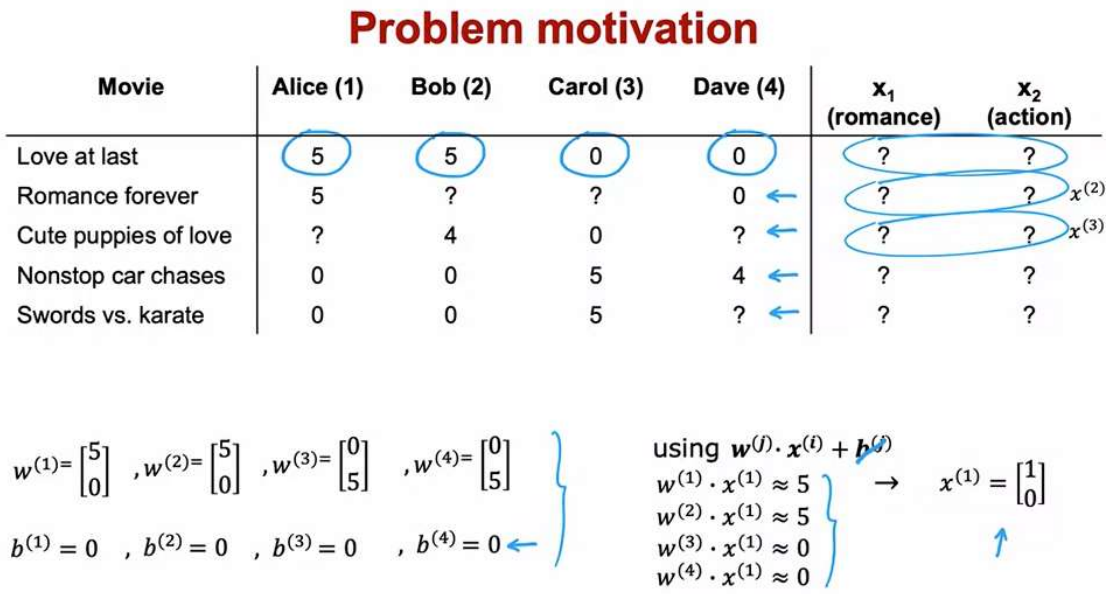


Um die Parameter w(1), b(1), w(2), b(2),...,w(nu), b(nu) zu lernen, würden wir diese Kostenfunktion oben nehmen und summieren alle Nu- Benutzer. Wir hätten also eine Summe von j=1 eins bis nu derselben Kostenfunktion, die wir oben geschrieben haben. Und das sind die Kosten für das Erlernen aller Parameter für alle Benutzer. Und wenn wir den Gradientenabstieg oder einen anderen Optimierungsalgorithmus verwenden, um dies als Funktion von w(1), b(1) bis hin zu w(nu), b(nu) zu minimieren, dann haben wir einen ziemlich guten Parametersatz zur Vorhersage von Filmbewertungen für alle Benutzer. Und Sie werden vielleicht bemerken, dass dieser Algorithmus der linearen Regression sehr ähnelt, wobei er eine ähnliche Rolle spielt wie die Ausgabe f(x) der linearen Regression. Erst jetzt trainieren wir für jeden der nu- Benutzer ein anderes lineares Regressionsmodell. Auf diese Weise können Sie Parameter lernen und Filmbewertungen vorhersagen, wenn Sie Zugriff auf diese Funktionen X1 und X2 haben. Das sagt Ihnen, wie viel kostet jeder der Filme, ein Liebesfilm, und wie viel kostet jeder der Filme, ein Actionfilm? Aber woher kommen diese Funktionen? Und was ist, wenn Sie keinen Zugriff auf solche Funktionen haben, die Ihnen genügend Details zu den Filmen liefern, mit denen Sie diese Vorhersagen treffen möchten? Im nächsten Video schauen wir uns die Modifikation dieses Algorithmus an. Sie ermöglichen es Ihnen, Vorhersagen zu treffen und Empfehlungen abzugeben. Auch wenn Sie nicht über erweiterte Funktionen verfügen, die die Elemente der Filme ausreichend detailliert beschreiben, um den Algorithmus auszuführen, den wir gerade gesehen haben. Lasst uns weitermachen und uns das im nächsten Video ansehen

Kollaborativer Filteralgorithmus

Im letzten Video haben Sie gesehen, wie Sie für jeden Film Funktionen festlegen können, z. B. die Funktionen x_1 und x_2, die Ihnen sagen, wie sehr es sich um einen Liebesfilm und wie viel um einen Actionfilm handelt. Dann können Sie grundsätzlich die lineare Regression verwenden, um zu lernen, Filmbewertungen vorherzusagen. Aber was ist, wenn Sie nicht über diese Funktionen x_1 und x_2

verfügen? Sehen wir uns an, wie Sie aus den Daten die Funktionen x_1 und x_2 lernen oder daraus entwickeln können. Hier sind die Daten, die wir zuvor hatten. Aber was wäre, wenn wir statt dieser Zahlen für x_1 und x_2 nicht im Voraus wüssten, welche Werte die Merkmale x_1 und x_2 haben? Ich werde sie hier durch Fragezeichen ersetzen. Nehmen wir nun zur Veranschaulichung an, wir hätten die Parameter für die vier Benutzer bereits gelernt. Nehmen wir an, wir haben gelernt, dass die Parameter w^1 gleich 5 und 0 und b^1 gleich 0 sind, für Benutzer eins. W^2 ist auch 5, 0 b^2, 0. W^3 ist 0, 5 b^3 ist 0, und für Benutzer vier ist W^4 auch 0, 5 und b^4 0, 0.



Machen Sie sich später Gedanken darüber, wie wir auf diese Parameter w und b gekommen sein könnten. Aber nehmen wir an, wir haben sie bereits. Zur Erinnerung: Um die Bewertung von Musik j für Film i vorherzusagen, verwenden wir das Punktprodukt w^j, die Funktionen von x_i plus b^j. Um dieses Beispiel zu vereinfachen, sind tatsächlich alle Werte von b gleich 0. Um den Schreibaufwand ein wenig zu reduzieren, werde ich b für den Rest dieses Beispiels ignorieren. Werfen wir einen Blick darauf, wie wir erraten können, welche sinnvollen Funktionen für Film eins sein könnten. Wenn dies die Parameter sind, die Sie auf der linken Seite haben, dann sollten wir angesichts der Tatsache, dass Alice den ersten Film mit 5 bewertet hat, wissen, dass w^1.x^1 ungefähr gleich 5 sein sollte und w^2.x^2 ebenfalls ungefähr gleich sein sollte gleich 5, weil Bob es mit 5 bewertet hat. W^3.x^1 sollte nahe bei 0 liegen und w^4.x^1 sollte ebenfalls nahe bei 0 liegen. Die Frage ist, welche Wahl für x_1 angesichts dieser Werte für w, die wir hier oben haben, dazu führt, dass diese Werte richtig sind? Nun, eine mögliche Wahl wäre, wenn die Features für diesen ersten Film 1, 0 wären. In diesem Fall wäre w^1.x^1 gleich 5, w^2.x^1 wäre gleich 5 und ähnlich, w^3 oder w^4 Skalarprodukt mit diesem Merkmalsvektor x_1 wäre gleich 0. Wir haben Folgendes: Wenn Sie hier die Parameter für alle vier Benutzer haben und wenn Sie in diesem Beispiel vier Bewertungen haben, die Sie möchten Versuchen Sie, eine Übereinstimmung zu finden, können Sie eine vernünftige Vermutung anstellen, was einen Feature-Vektor x_1 für Film eins auflistet, der gute Vorhersagen für diese vier Bewertungen oben machen würde. Wenn Sie über diese Parametervektoren verfügen, können Sie in ähnlicher Weise auch versuchen, einen Merkmalsvektor x_2 für den zweiten Film, einen Merkmalsvektor x_3 für den dritten Film usw. zu erstellen, um zu versuchen, die Vorhersagen des Algorithmus für diese zusätzlichen Filme annähernd zu treffen Wie lauteten eigentlich die Bewertungen der Nutzer? Lassen Sie uns eine Kostenfunktion entwickeln, um die Werte von x_1 und x_2 tatsächlich zu lernen. Beachten Sie übrigens, dass dies nur funktioniert, weil wir Parameter für vier Benutzer haben. Dadurch können wir versuchen, geeignete Merkmale x_1 zu erraten. Aus

diesem Grund verfügen Sie in einer typischen linearen Regressionsanwendung, wenn Sie nur einen einzelnen Benutzer hätten, nicht über genügend Informationen, um herauszufinden, welche Funktionen x_1 und x_2 wären, weshalb Sie in den linearen Regressionskontexten, die Sie gesehen haben, nicht über genügend Informationen verfügen Natürlich 1: Sie können die Funktionen x_1 und x_2 nicht von Grund auf neu entwickeln. Bei der kollaborativen Filterung liegt dies jedoch daran, dass Sie Bewertungen von mehreren Benutzern desselben Artikels mit demselben Film haben. Dadurch ist es möglich, die möglichen Werte für diese Merkmale zu erraten. Gegeben w^1, b^1, w^2, b^2 usw. durch w^n_u und b^n_u für die n tiefgestellten u-Benutzer. Wenn Sie die Funktionen x^i für einen bestimmten Film lernen möchten, ist i eine Kostenfunktion, die wir verwenden könnten. Ich möchte den quadratischen Fehler wie üblich minimieren. Wenn die vorhergesagte Bewertung von Benutzer j für Film i dadurch gegeben ist, nehmen wir die quadrierte Differenz zur tatsächlichen Filmbewertung y,i,j. Lassen Sie uns wie zuvor alle Benutzer j zusammenfassen. Aber das wird eine Summe über alle Werte von j sein, wobei r, i, j gleich I ist. Ich füge dort wie üblich eine 1,5 hinzu. Da ich dies als Kostenfunktion für x^i definiert habe. Wenn wir dies dann als Funktion von x^i minimieren, wählen Sie die Funktionen für Film i aus. Deshalb werden wir für alle Benutzer J, die den Film i bewertet haben, versuchen, die quadratische Differenz zwischen dem, was Ihre Wahl der Funktionen x^i ergibt, in Bezug auf die vorhergesagte Filmbewertung minus der tatsächlichen Filmbewertung, die der Benutzer ihm gegeben hat, zu minimieren. Wenn wir schließlich einen Regularisierungsterm hinzufügen möchten, fügen wir das übliche Plus-Lambda über 2 hinzu, K ist gleich 1 bis n, wobei n wie üblich die Anzahl der Merkmale von x^i im Quadrat ist. Um schließlich alle Funktionen x1 bis x^n_m kennenzulernen, da wir n_m Filme haben, können wir diese Kostenfunktion zusätzlich verwenden und sie über alle Filme summieren. Die Summe von i entspricht 1 bis zur Anzahl der Filme. Nehmen Sie dann einfach diesen Term von oben und dies wird zu einer Kostenfunktion zum Erlernen der Funktionen für alle Filme im Datensatz. Wenn Sie also die Parameter w und b für alle Benutzer haben und diese Kostenfunktion als Funktion von x1 bis x^n_m mithilfe eines Gradientenabstiegs oder eines zellularen Algorithmus minimieren, können Sie tatsächlich ziemlich gut erraten, wie man gute Funktionen für lernt die Filme. Dies ist für die meisten Anwendungen des maschinellen Lernens ziemlich bemerkenswert. Die Funktionen mussten extern bereitgestellt werden, aber mit diesem Algorithmus können wir tatsächlich die Funktionen für einen bestimmten Film lernen. Aber was wir bisher in diesem Video gemacht haben, ist davon ausgegangen, dass Sie die Parameter w und b für die verschiedenen Benutzer haben. Woher bekommen Sie diese Parameter? Nun, lasst uns den Algorithmus aus dem letzten Video zum Erlernen von w und b und dem, worüber wir gerade in diesem Video zum Erlernen von x gesprochen haben, zusammenstellen, und das wird uns unseren kollaborativen Filteralgorithmus geben.

Cost function

Given $w^{(1)}, b^{(1)}, w^{(2)}, b^{(2)}, \cdots, w^{(n_u)}, b^{(n_u)}$

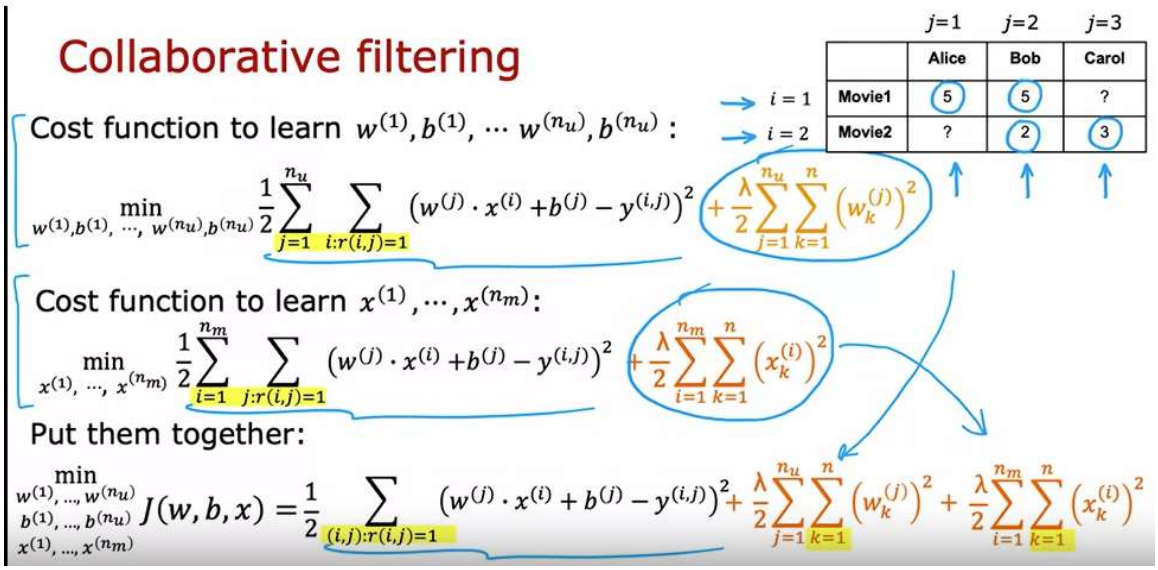
to learn $x^{(i)}$:

$$J(x^{(i)}) = \frac{1}{2} \sum_{j:r(i,j)=1} \left(w^{(j)} \cdot x^{(i)} + b^{(j)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(x_k^{(i)}\right)^2$$

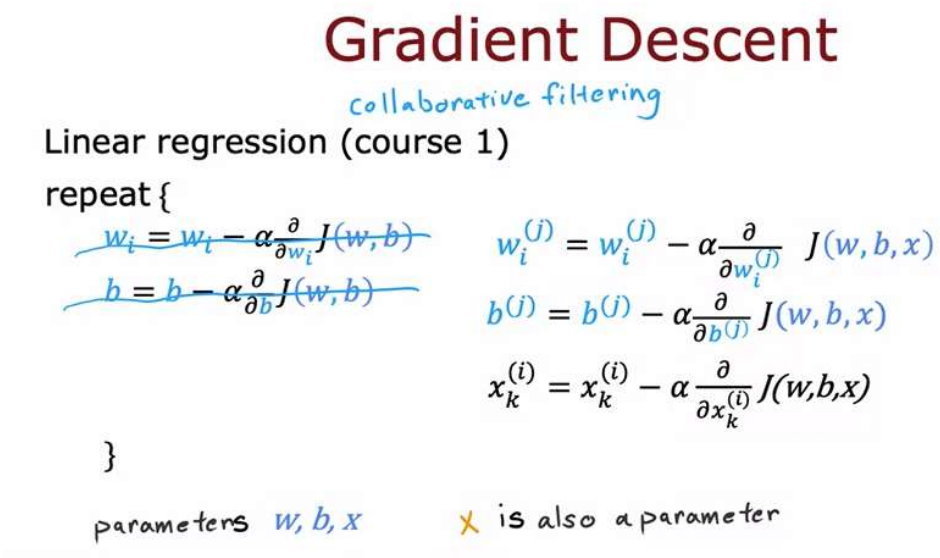
→ To learn $x^{(1)}, x^{(2)}, \cdots, x^{(n_m)}$:

$$J(x^{(1)}, x^{(2)}, \ldots, x^{(n_m)}) = \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} \left(w^{(j)} \cdot x^{(i)} + b^{(j)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} \left(x_k^{(i)}\right)^2$$

Hier ist die Kostenfunktion zum Erlernen der Funktionen. Das haben wir auf der letzten Folie abgeleitet. Nun stellt sich heraus, dass, wenn wir diese beiden zusammenfügen, dieser Begriff hier genau derselbe ist wie dieser Begriff hier. Beachten Sie, dass die Summe über j aller Werte von i, bei der r,i,j gleich 1 ist, dasselbe ist wie die Summierung über alle Werte von i mit allen j, wobei r,i,j gleich 1 ist. Diese Summierung ist nur eine Summierung über alle Benutzerfilmpaare, für die eine Bewertung vorliegt. Was ich tun werde, ist, diese beiden Kostenfunktionen zusammenzusetzen und dies zu erhalten, wobei ich die Summierung einfach expliziter als Summierung über alle Paare i und j ausschreibe, wo wir eine Bewertung der üblichen quadrierten Kostenfunktion und haben Lassen Sie mich dann den Regularisierungsterm aus dem Lernen der Parameter w und b nehmen und ihn hier einfügen. Nehmen wir den Regularisierungsterm aus dem Lernen der Merkmale x und setzen ihn hier ein. Das Ergebnis ist unsere Gesamtkostenfunktion für das Lernen von w, b und x. Es stellt sich heraus, dass dieser Algorithmus tatsächlich funktioniert, wenn Sie diese Kostenfunktion als Funktion von w und b als y als x minimieren. Hier ist, was ich meine. Wenn wir drei Benutzer und zwei Filme hätten und Sie Bewertungen für diese vier Filme haben, aber nicht für diese beiden, dann wird hier die Summe aller Benutzer ermittelt. Denn Benutzer 1 hat die Kostenfunktion dafür bestimmt, denn Benutzer 2 hat die Kostenfunktion dafür bestimmt, denn Benutzer 3 hat die Kostenfunktion dafür bestimmt. Wir summieren zuerst die Benutzer und erstellen dann einen Begriff für jeden Film, für den es eine Bewertung gibt. Aber eine alternative Möglichkeit, die Summierung durchzuführen, besteht darin, sich zuerst Film 1 anzusehen, das ist es, was diese Summierung hier bewirkt, und dann alle Benutzer einzubeziehen, die Film 1 bewertet haben, und sich dann Film 2 anzusehen und einen Begriff für alle Benutzer festzulegen der Film 2 bewertet hat. Sie sehen, dass wir in beiden Fällen nur die vier Bereiche zusammenfassen, in denen der Benutzer den entsprechenden Film bewertet hat. Deshalb sind diese Summierung oben und diese Summierung hier die beiden Möglichkeiten, alle Paare zu summieren, bei denen der Benutzer diesen Film bewertet hat. Wie minimiert man diese Kostenfunktion als Funktion von w, b und x?



Eine Möglichkeit wäre, den Gradientenabstieg zu verwenden. Als wir in Kurs 1 etwas über die lineare Regression lernten, war dies der Gradientenabstiegsalgorithmus, den Sie gesehen hatten, wobei wir die Kostenfunktion J hatten, die eine Funktion der Parameter w und b ist, und wir den Gradientenabstieg wie folgt anwendeten.

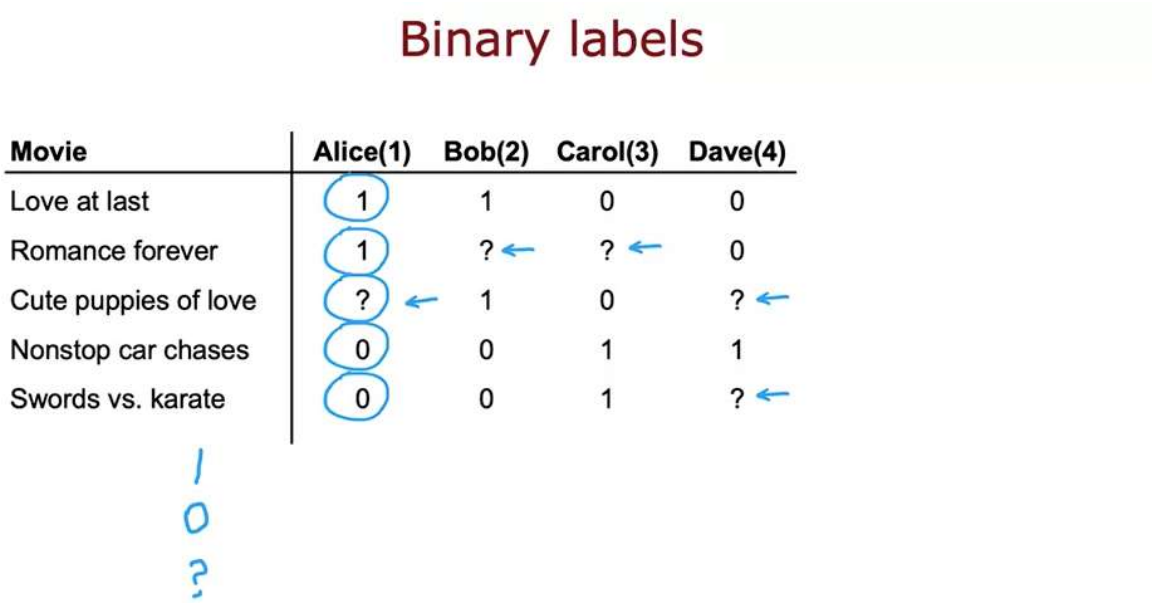


Bei der kollaborativen Filterung ist die Kostenfunktion nur eine Funktion von w und b ist nun eine Funktion von w, b und x. Ich verwende w und b hier, um die Parameter für alle Benutzer zu bezeichnen, und x hier nur informell, um die Funktionen aller Filme zu bezeichnen. Wenn Sie jedoch in der Lage sind, partielle Ableitungen in Bezug auf die verschiedenen Parameter vorzunehmen, können Sie die Parameter wie folgt weiter aktualisieren. Aber jetzt müssen wir dies auch in Bezug auf x optimieren. Wir möchten auch jeden dieser Parameter x mithilfe des Gradientenabstiegs wie folgt aktualisieren. Es stellt sich heraus, dass Sie, wenn Sie dies tun, tatsächlich ziemlich gute Werte für w und b sowie x finden. In dieser Problemformulierung sind die Parameter w und b, und x ist ebenfalls ein Parameter. Um schließlich die Werte von x zu erfahren, aktualisieren wir x auch als x minus der partiellen Ableitung der Kosten w, b, x nach x. Ich verwende die Notation hier ein wenig informell und achte nicht besonders sorgfältig auf die hochgestellten und tiefgestellten Zeichen, aber die wichtigste Erkenntnis, die Sie daraus hoffentlich mitnehmen können, ist, dass die Parameter dieses Modells w und b sind, und jetzt ist es auch x ein Parameter, weshalb wir die Kostenfunktion als Funktion aller drei dieser Parametersätze w und b sowie x minimieren. Der von uns abgeleitete Durchschnitt wird als kollaboratives Filtern bezeichnet. Der Name kollaboratives Filtern bezieht sich auf den Sinn, dass mehrere Benutzer denselben Film gemeinsam bewertet haben und Sie dadurch ein Gefühl dafür bekommen, wie dieser Film aussehen könnte, sodass Sie erraten können, wofür geeignete Funktionen geeignet sind Dadurch können Sie vorhersagen, wie andere Benutzer, die denselben Film noch nicht bewertet haben, ihn in Zukunft bewerten werden. Bei dieser kollaborativen Filterung handelt es sich um das Sammeln von Daten von mehreren Benutzern. Diese Zusammenarbeit zwischen Benutzern hilft Ihnen, Bewertungen auch für andere Benutzer in der Zukunft vorherzusagen. Bisher wurden in unserer Problemformulierung Filmbewertungen von 1 bis 5 Sternen oder von 0 bis 5 Sternen verwendet. Ein sehr häufiger Anwendungsfall empfohlener Systeme ist, wenn Sie binäre Bezeichnungen haben, z. B. dass der Benutzer ein Element bevorzugt, mag oder mit ihm interagiert. Schauen wir uns im nächsten Video eine Verallgemeinerung des Modells, das Sie bisher gesehen haben, auf binäre Labels an. Schauen wir uns das im nächsten Video an.

Binäre Labels: Favoriten, Likes und Klicks

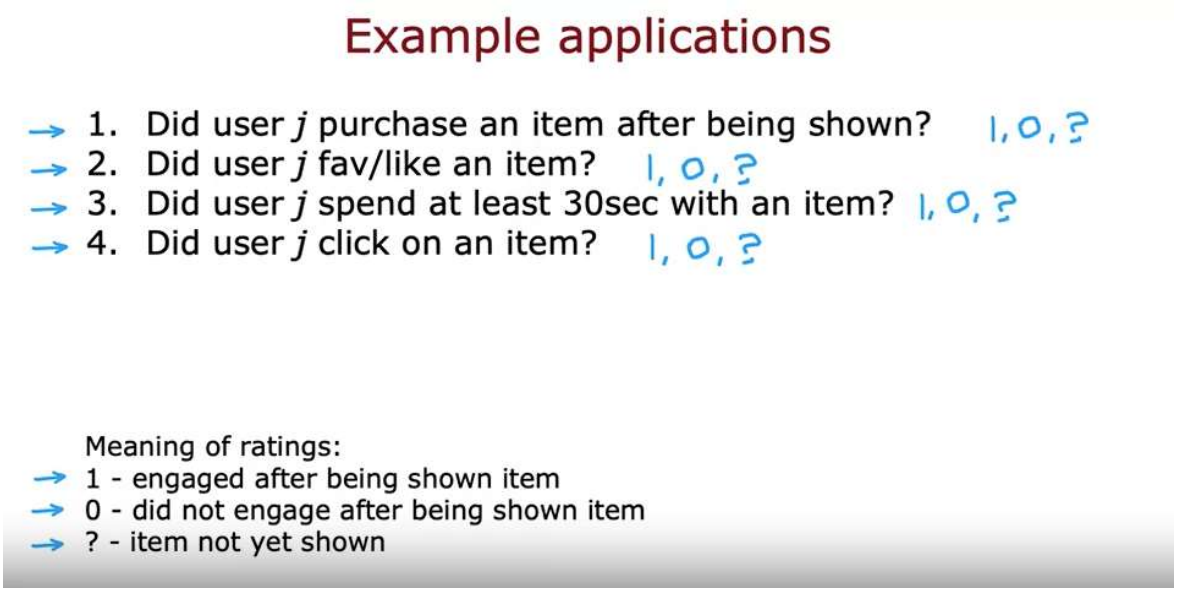
Bei vielen wichtigen Anwendungen empfohlener Systeme oder kollektiver Filteralgorithmen handelte es sich um binäre Etiketten, bei denen der Benutzer Ihnen nicht eine Bewertung von eins bis fünf oder null bis fünf Sternen gibt, sondern Ihnen irgendwie das Gefühl vermittelt, dass ihm dieser Artikel gefällt oder dass ihm dieser nicht gefällt Artikel. Schauen wir uns an, wie Sie den Algorithmus, den Sie gesehen haben, auf diese Einstellung verallgemeinern können. Der Prozess, den wir zur

Verallgemeinerung des Algorithmus verwenden werden, wird sehr daran erinnern, wie wir im ersten Kurs von der linearen Regression zur logistischen Regression und von der Vorhersage von Zahlen zur Vorhersage einer binären Bezeichnung übergegangen sind. Werfen wir einen Blick darauf. Hier ist ein Beispiel für einen kollaborativen Filterdatensatz mit binären Bezeichnungen.



Eine davon ist die Notiz, dass der Benutzer einen bestimmten Film mochte oder sich mit ihm beschäftigte. Das Etikett eins könnte also bedeuten, dass Alice den Film „Love at last“ bis zum Ende und „Romance Forever“ bis zum Ende gesehen hat. Aber nachdem ich ein paar Minuten ununterbrochener Verfolgungsjagden abgespielt hatte, beschloss ich, das Video zu stoppen und weiterzumachen. Oder es könnte bedeuten, dass sie in einer App explizit auf „Gefällt mir“ oder „ Favorit“ geklickt hat, um anzuzeigen, dass ihr diese Filme gefallen. Aber nach dem Ausprobieren von ununterbrochenen Wolkenjägern und Schwertkämpfen gegen Karate hat es nicht so gut geklappt. Und das Fragezeichen bedeutet normalerweise, dass der Benutzer den Artikel noch nicht gesehen hat und daher nicht entscheiden konnte, ob er bei diesem bestimmten Artikel auf „ Gefällt mir“ oder „ Favorit“ klicken soll oder nicht. Die Frage ist also, wie wir den kollaborativen Filterdurchschnitt, den Sie im letzten Video gesehen haben, in diesem Daaset zum Laufen bringen können. Und indem wir vorhersagen, wie wahrscheinlich es ist, dass Alice, Bob Carol und Dave die Artikel mögen, die sie noch nicht bewertet haben, können wir dann entscheiden, wie sehr wir ihnen diese Artikel empfehlen sollten. Es gibt viele Möglichkeiten zu definieren, was das Label Eins und was das Label Null ist und was das Label-Fragezeichen bei der kollaborativen Filterung mit binären Labels ist. Schauen wir uns ein paar Beispiele an. Auf einer Online-Shopping-Website könnte das Etikett angeben, ob sie sich für den Kauf eines Artikels entschieden haben oder nicht, nachdem sie damit in Kontakt gekommen sind, nachdem ihnen der Artikel gezeigt wurde. Eins würde also bedeuten, dass sie es gekauft haben, null würde bedeuten, dass sie es nicht gekauft haben. Und das Fragezeichen würde bedeuten, dass ihnen der Gegenstand nicht einmal gezeigt wurde und sie ihm nicht einmal ausgesetzt waren. Oder in einer Social-Media-Umgebung könnten die Bezeichnungen „Eins“ oder „Null“ bedeuten, dass der Benutzer einen Artikel als Favorit oder „Gefällt mir“ markiert hat, nachdem er ihm gezeigt wurde. Und ein Fragezeichen wäre, wenn ihnen der Artikel noch nicht angezeigt wurde oder viele Websites, anstatt nach einer expliziten Benutzerbewertung zu fragen, das Benutzerverhalten nutzen, um zu erraten, ob dem Benutzer der Artikel gefällt. So können Sie beispielsweise messen, ob ein Benutzer mindestens 30 Sekunden mit einem Artikel verbringt. Und wenn dies der Fall ist, weisen Sie ihm die Bezeichnung „Eins“ zu, weil der Benutzer das Element ansprechend fand, oder wenn einem Benutzer ein Element gezeigt wurde, sich aber nicht mindestens 30 Sekunden damit beschäftigt hat, ist dies ein Zeichen

dafür, dass die Bezeichnung „Null“ lautet. Oder wenn dem Benutzer das Element noch nicht angezeigt wurde, weisen Sie ihm ein Fragezeichen zu.



Eine weitere Möglichkeit, eine Bewertung implizit als Funktion des Nutzerverhaltens zu generieren, besteht darin, zu sehen, dass der Nutzer auf einen Artikel klickt. Dies geschieht häufig in der Online-Werbung, bei der, wenn dem Nutzer eine Anzeige angezeigt wurde, wenn er darauf geklickt hat, dem Arbeitspunkt eins zugeordnet wurde, wenn er nicht geklickt hat, dem Label Null zugeordnet wurde und auf das Fragezeichen verwiesen wurde, wenn der Nutzer dies nicht getan hat wurde diese Anzeige überhaupt erst angezeigt. So oft haben diese binären Bezeichnungen eine grobe, bedeutungslose Folge. Ein Arbeitsaufwand von eins bedeutet, dass der Benutzer interagiert hat, nachdem ihm ein Artikel angezeigt wurde. Und interagiert könnte bedeuten, dass er geklickt hat oder 30 Sekunden damit verbracht hat oder den Artikel explizit favorisiert hat oder kaufen möchte. Eine Null bedeutet, dass der Benutzer nicht interagiert, nachdem ihm das Element angezeigt wurde. Das Fragezeichen zeigt an, dass das Element dem Benutzer noch nicht angezeigt wurde. Schauen wir uns also angesichts dieser binären Bezeichnungen an, wie wir unseren Algorithmus, der der linearen Regression aus den vorherigen Videos sehr ähnlich ist, auf die Vorhersage dieser binären Ausgaben verallgemeinern können. Zuvor haben wir die Bezeichnung yij als wj.xi+b vorhergesagt. Das ähnelte also stark einem linearen Regressionsmodell. Für binäre Beschriftungen werden wir vorhersagen, dass die Wahrscheinlichkeit von yijb =1 nicht durch wj.xi+b gegeben ist. Aber nach g dieser Formel heißt es, wir sind jetzt g(z) 1/1 +e zum -z. Das ist also die logistische Funktion, genau wie wir sie bei der logistischen Regression gesehen haben. Und was wir tun würden, wäre, etwas zu nehmen, das einem linearen Regressionsmodell ähnelte, und es in etwas umzuwandeln, das einem logistischen Regressionsmodell sehr ähnlich wäre, bei dem nun die Wahrscheinlichkeit vorhergesagt wird, dass yij 1 ist, was bedeutet, dass der Benutzer mit oder interagiert hat Ich mag den Artikel, der dieses Modell verwendet. Um diesen Algorithmus zu erstellen, müssen wir auch die Kostenfunktion von der Kostenfunktion mit quadratischem Fehler in die Kostenfunktion ändern, die für binäre Bezeichnungen für ein logistisches Regressionsmodell besser geeignet ist.

From regression to binary classification

- Previously:
  - Predict $y^{(i,j)}$ as $w^{(j)} \cdot x^{(i)} + b^{(j)}$
- For binary labels:
  - Predict that the probability of $y^{(i,j)} = 1$ is given by $g(w^{(j)} \cdot x^{(i)} + b^{(j)})$

where $g(z) = \frac{1}{1+e^{-z}}$

Bisher war dies also die Kostenfunktion, die wir hatten, wobei dieser Term eine ähnliche Rolle wie f(x) spielte, die Vorhersage des Algorithmus. Wenn Sie jetzt binäre Beschriftungen haben, yij, wenn die Beschriftungen eins oder null oder ein Fragezeichen sind, dann wird die Vorhersage f(x) anstelle von wj.xi+b j zu g davon, wobei g die logistische Funktion ist. Und ähnlich wie bei der Ableitung der logistischen Regression hatten wir die folgende Verlustfunktion für ein einzelnes Beispiel geschrieben, das den Verlust hatte, wenn der Algorithmus f(x) vorhersagt und die wahre Bezeichnung y war, der Verlust war dieser. Es war -y log fy log 1-f. Dies wird manchmal auch als binäre Kreuzentropiekostenfunktion bezeichnet. Dies ist jedoch eine Standardkostenfunktion, die wir für die logistische Regression verwendet haben, ebenso wie für die binären Klassifizierungsprobleme beim Training neuronaler Netze. Und um dies an die Einstellung der kollaborativen Filterung anzupassen, möchte ich die Kostenfunktion aufschreiben, die nun eine Funktion aller Parameter w und b sowie aller Parameter x ist, die die Merkmale der einzelnen Filme oder Elemente von sind. Wir müssen jetzt einige über alle Paare ij berechnen, wobei riij = 1 ist.

Cost function for binary application

Previous cost function:

$$\frac{1}{2}\sum_{(i,j):r(i,j)=1}\underbrace{\left(w^{(j)} \cdot x^{(i)} + b^{(j)}\right)}_{f(x)} - y^{(i,j)}\Big)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(w_k^{(j)}\right)^2$$

Loss for binary labels $y^{(i,j)}$: $f_{(w,b,x)}(x) = g(w^{(j)} \cdot x^{(i)} + b^{(j)})$

$$L\left(f_{(w,b,x)}(x), y^{(i,j)}\right) = -y^{(i,j)}\log\left(f_{(w,b,x)}(x)\right) - \left(1 - y^{(i,j)}\right)\log\left(1 - f_{(w,b,x)}(x)\right)$$

Loss for single example

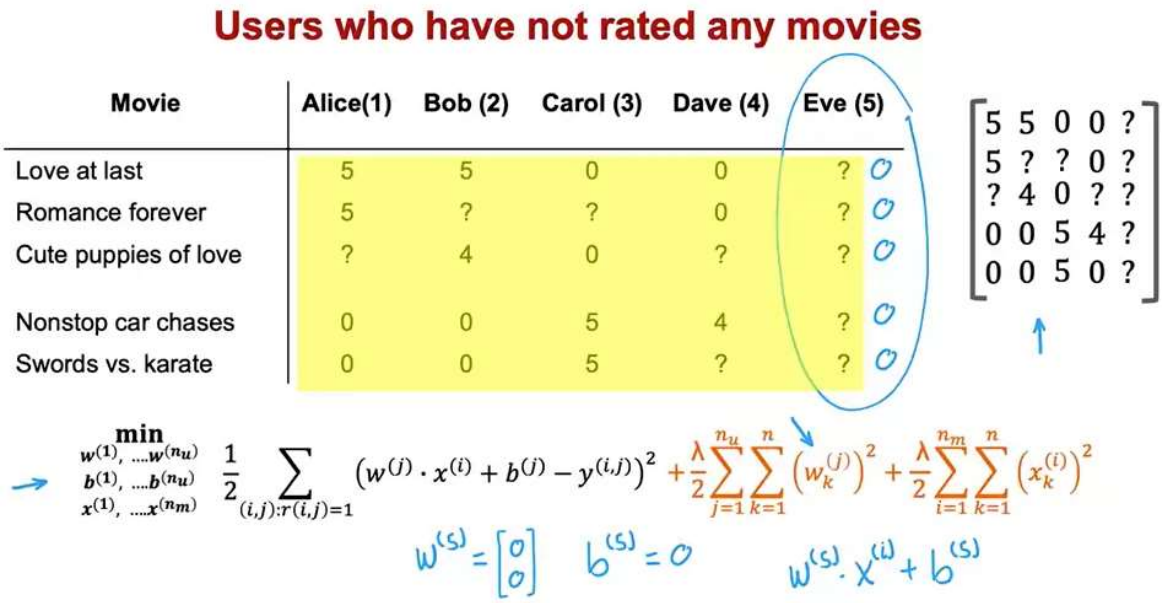
$$J(w,b,x) = \sum_{(i,j):r(i,j)=1} L\left(f_{(w,b,x)}(x), y^{(i,j)}\right)$$

Beachten Sie, dass dies einfach der Summierung oben ähnelt. Und jetzt verwenden wir anstelle dieser quadratischen Fehlerkostenfunktion diese Verlustfunktion. Es gibt eine Funktion von f(x), yij. Wo f(x) hier? Das ist meine Abkürzung. Meine Abkürzung für g(w) j.x1+ej. Wenn wir dies hier einfügen, erhalten Sie die Kostenfunktion, die sie für die gemeinsame Filterung binärer Labels

verwenden könnten. Das war's. Auf diese Weise können Sie die lineare Regression, wie den kollaborativen Filteralgorithmus, so verallgemeinern, dass sie mit binären Bezeichnungen funktioniert. Und dies erweitert tatsächlich ganz erheblich die Anzahl der Anwendungen, die Sie mit diesem Algorithmus ansprechen können. Obwohl Sie nun die Schlüsselstruktur und Kostenfunktion des Algorithmus kennengelernt haben, gibt es auch einige Implementierungstipps, die dazu beitragen, dass Ihr Algorithmus viel besser funktioniert. Fahren wir mit dem nächsten Video fort, um einen Blick auf einige Details zu werfen, wie Sie es implementieren, und auf einige kleine Modifikationen, die dafür sorgen, dass das Album viel schneller läuft. Kommen wir zum nächsten Video.

Mittelwert Normalisierung

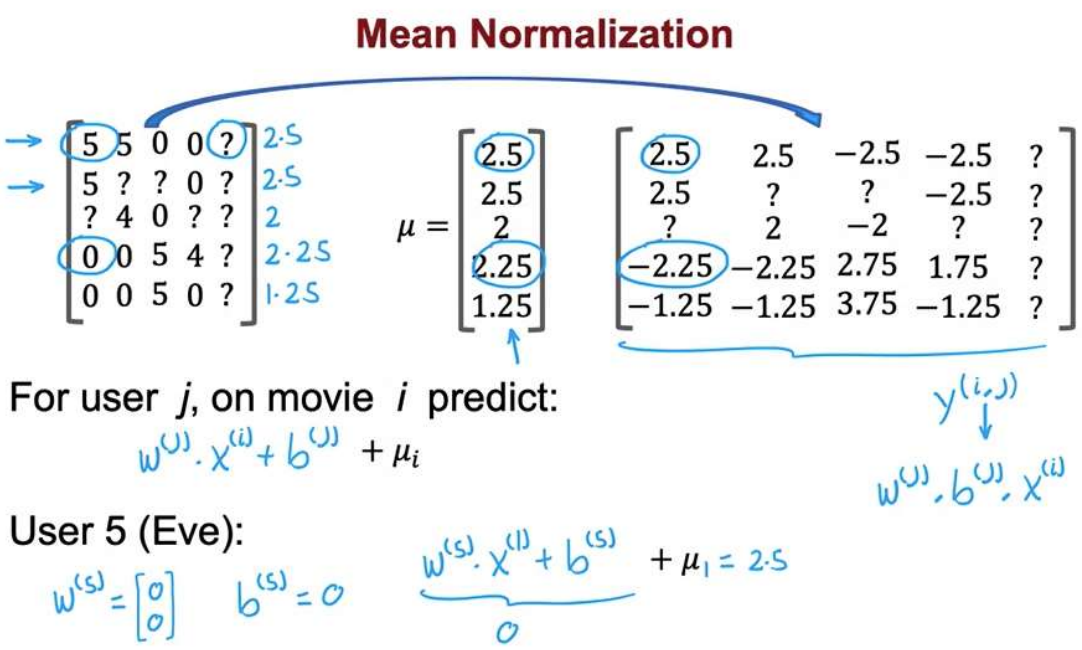
Bereits im ersten Kurs haben Sie gesehen, wie bei der linearen Regression eine zukünftige Normalisierung dazu beitragen kann, dass der Algorithmus schneller läuft. Wenn Sie ein empfohlenes System mit weitreichenden Zahlen wie Filmbewertungen von eins bis fünf oder null bis fünf Sternen erstellen, stellt sich heraus, dass Ihr Algorithmus effizienter läuft. Und Sie erzielen auch eine etwas bessere Leistung, wenn Sie zunächst eine Mittelwertnormalisierung durchführen. Wenn Sie also die Filmbewertungen normalisieren, um einen konsistenten Durchschnittswert zu erhalten, schauen wir uns an, was das bedeutet. Hier ist also der Datensatz, den wir verwendet haben. Unten sehen Sie die Kostenfunktion, mit der Sie die Parameter für das Modell gelernt haben. Um die Mittelwertnormalisierung zu erklären, Im Ich werde tatsächlich die fünfte Benutzerin Eve hinzufügen, die noch keine Filme bewertet hat. Und Sie sehen gleich, dass das Hinzufügen einer Mittelwertnormalisierung dem Algorithmus dabei helfen wird, bessere Vorhersagen über den Benutzer Eve zu treffen.



Wenn Sie tatsächlich einen kollaborativen Filteralgorithmus für diese Daten trainieren würden, dann weil wir aufgrund dieses Regularisierungsterms versuchen, die Parameter w klein zu machen. Wenn Sie den Algorithmus für diesen Datensatz ausführen würden, erhalten Sie tatsächlich die Parameter w für den fünften Benutzer, wobei der Benutzer Eve gleich [0 0] und höchstwahrscheinlich auch b(5) = 0 ist. Weil Eve hat noch keine Filme bewertet, die Parameter w und b haben keinen Einfluss auf diesen ersten Term in der Kostenfunktion, da keine der Bewertungen von Eves Film in dieser Kostenfunktion mit quadratischem Fehler eine Rolle spielt. Dies zu minimieren bedeutet also, die

Parameter so klein wie möglich zu machen. Wir haben b nicht wirklich reguliert. Wenn Sie b jedoch standardmäßig auf 0 initialisieren, erhalten Sie am Ende auch b(%) = 0. Wenn dies jedoch die Parameter für Benutzer 5 sind, also für Eve, dann wird der Durchschnitt am Ende vorhersagen, dass alle Filmbewertungen von Eve w(%).x für Film i + b(5) sind. Und dies ist gleich 0, wenn w und b oben gleich 0 sind. Und so sagt dieser Algorithmus voraus, dass, wenn Sie einen neuen Benutzer haben, der noch nichts bewertet hat, wir glauben, dass er alle Filme mit null Sternen bewerten wird, und das ist nicht besonders hoffnungsvoll.

In diesem Video werden wir sehen, dass die Mittelwertnormalisierung diesem Algorithmus dabei hilft, bessere Vorhersagen über die Filmbewertungen für einen neuen Benutzer zu treffen, der noch keine Filme bewertet hat. Um die Mittelwertnormalisierung zu beschreiben, möchte ich alle Werte hier, einschließlich aller Fragezeichen für Eve, in eine zweidimensionale Matrix wie diese einfügen. Nur um alle Bewertungen inklusive der Fragezeichen nachhaltiger und kompakter aufzuschreiben. Um eine Mittelwertnormalisierung durchzuführen, nehmen wir alle diese Bewertungen und berechnen für jeden Film die durchschnittliche Bewertung, die gegeben wurde. Film eins hatte also zwei 5er und zwei 0er, die durchschnittliche Bewertung liegt also bei 2,5. Film zwei hatte eine 5 und eine 0, das ergibt also einen Durchschnitt von 2,5. Film drei: 4 und 0 ergeben im Durchschnitt eine Bewertung von 2. Film vier ergibt im Durchschnitt eine Bewertung von 2,25. Und Film fünf ist nicht so beliebt und hat eine durchschnittliche Bewertung von 1,25. Also werde ich alle diese fünf Zahlen nehmen und sie in einem Vektor zusammenfassen, den ich μ nennen werde, weil dies der Vektor der durchschnittlichen Bewertungen ist, die jeder der Filme hatte. Durchschnittlich der Benutzer, die diesen bestimmten Film gelesen haben. Anstatt diese ursprünglichen 0- bis 5-Sterne-Bewertungen hier zu verwenden, werde ich diese nehmen und von jeder Bewertung die Durchschnittsbewertung abziehen, die sie gegeben hat. Die Bewertung dieses Films betrug zum Beispiel 5. Ich ziehe 2,5 davon ab und erhalte hier 2,5. Dieser Film hatte eine 0-Sterne-Bewertung. Ich ziehe 2,25 ab, was mir eine Bewertung von -2,25 ergibt und so weiter für alle jetzt fünf Benutzer, einschließlich der neuen Benutzerin Eve, sowie für alle fünf Filme. Dann werden diese neuen Werte auf der rechten Seite zu Ihren neuen Werten von Y(i,j). Wir gehen davon aus, dass Benutzer 1 Film eins mit 2,5 und Film vier mit -2,25 bewertet hat. Und damit können Sie dann w(j), b(j) und x(i) lernen, genauso wie zuvor für Benutzer j im Film i, Sie würden w(j).x(i) + b(j) vorhersagen. Da wir jedoch während dieses Mittelwertnormalisierungsschritts μ i für Film i abgezogen hatten, um keine negative Sternebewertung vorherzusagen, ist dies für Benutzerbewertungen von 0 bis 5 Sternen unmöglich.



Wir müssen diesen μ i wieder addieren, der genau der Wert ist, den wir subtrahiert haben. Wenn wir uns also als konkretes Beispiel ansehen, was mit Benutzer 5 mit der neuen Benutzerin Eve passiert, weil sie noch keine Filme bewertet hat, lernt der Durchschnitt möglicherweise die Parameter w(5) = [0 0] und sagt b(5) = 0. Wenn wir uns also die vorhergesagte Bewertung für Film eins ansehen, werden wir vorhersagen, dass Eve ihn mit w(5).x1 + b(5) bewerten wird, aber das ist 0 und dann + μ1, was 2,5 entspricht. Daher erscheint es vernünftiger anzunehmen, dass Eve diesen Film wahrscheinlich mit 2,5 bewertet, als dass Eve allen Filmen null Sterne geben wird, nur weil sie noch keine Filme bewertet hat. Tatsächlich führt dieser Algorithmus dazu, dass die anfänglichen Schätzungen für die neue Benutzerin Eve genau dem Mittelwert aller anderen Benutzer entsprechen, die diese fünf Filme bewertet haben. Und es erscheint sinnvoller, die durchschnittliche Bewertung der Filme zu nehmen, als zu vermuten, dass alle Bewertungen von Eve bei Null liegen werden. Es stellt sich heraus, dass durch die Normalisierung des Mittelwerts der verschiedenen Filmbewertungen auf Null der Optimierungsalgorithmus für das empfohlene System auch etwas schneller läuft. Allerdings verhält sich der Algorithmus dadurch viel besser für Benutzer, die keine Filme oder nur eine sehr kleine Anzahl von Filmen bewertet haben. Und die Vorhersagen werden vernünftiger. In diesem Beispiel haben wir jede Zeile dieser Matrix so normalisiert, dass sie einen Mittelwert von Null hat. Wir haben festgestellt, dass dies hilfreich ist, wenn es einen neuen Benutzer gibt, der noch nicht viele Filme bewertet hat. Es gibt noch eine andere Alternative, die Sie verwenden könnten: Stattdessen können Sie die Spalten dieser Matrix so normalisieren, dass sie einen Mittelwert von Null haben. Und das wäre auch vernünftig. Aber ich denke, in dieser Anwendung scheint es wichtiger zu sein, die Zeilen zu normalisieren, damit Sie einem neuen Benutzer angemessene Bewertungen geben können, als die Spalten zu normalisieren. Eine Normalisierung der Spalten würde hoffen, wenn es einen brandneuen Film gäbe, den noch niemand bewertet hat. Aber wenn es einen brandneuen Film gibt, den noch niemand bewertet hat, sollten Sie diesen Film zunächst wahrscheinlich nicht zu vielen Benutzern zeigen, da Sie nicht viel über diesen Film wissen. Daher erscheint mir die Normalisierung der Spalten „Hoffnung“ im Fall eines Films ohne Bewertungen weniger wichtig als die Normalisierung der Regeln „Hoffnung“ im Fall eines neuen Benutzers, der noch kaum Filme bewertet hat. Und wenn Sie im Übungslabor dieser Woche Ihr eigenes empfohlenes System erstellen, sollte die Normalisierung nur der Rollen problemlos funktionieren. Das ist also eine mittlere Normalisierung. Dadurch läuft der Algorithmus etwas schneller. Noch wichtiger ist jedoch, dass der Algorithmus dadurch viel bessere und vernünftigere

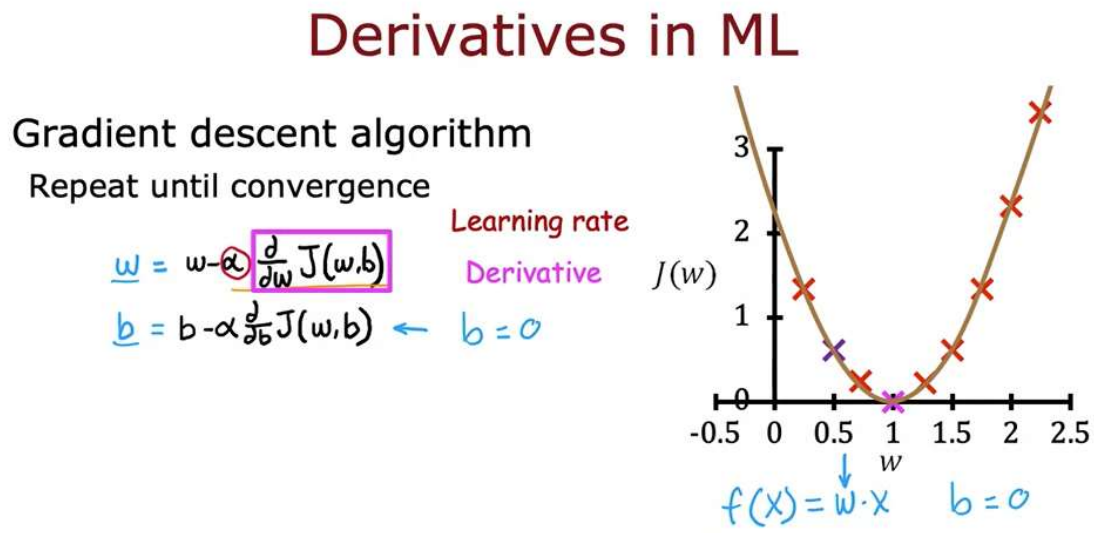
Vorhersagen liefert, wenn es Benutzer gibt, die nur sehr wenige oder gar keine Filme bewertet haben. Dieses Implementierungsdetail der Mittelwertnormalisierung sorgt dafür, dass Ihr empfohlenes System viel besser funktioniert. Lassen Sie uns als Nächstes im nächsten Video darüber sprechen, wie Sie diesen intensiven Flow für sich umsetzen können.

TensorFlow- Implementierung des „collaborative Filtering“

In diesem Video werfen wir einen Blick darauf, wie Sie TensorFlow verwenden können, um den collaborativen Filteralgorithmus zu implementieren. Sie sind es vielleicht gewohnt, sich TensorFlow als Werkzeug zum Aufbau neuronaler Netze vorzustellen. Und es ist. Es ist auch ein großartiges Werkzeug zum Aufbau neuronaler Netze.

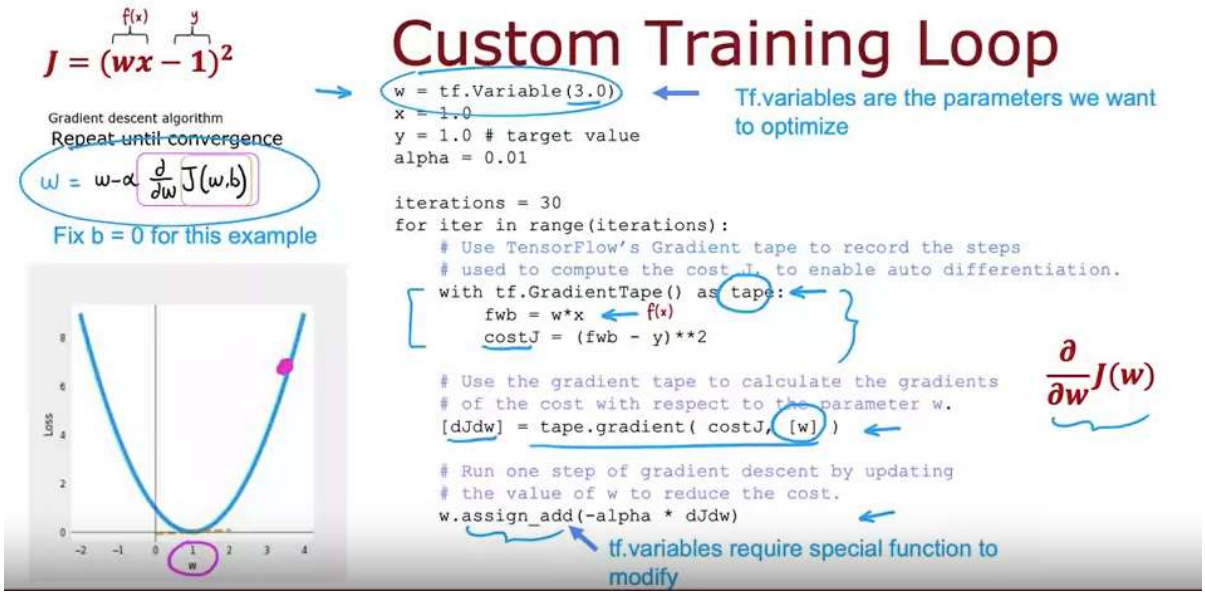
Und es stellt sich heraus, dass TensorFlow auch große Hoffnungen auf die Entwicklung anderer Arten von Lernalgorithmen machen kann. Wie der kollaborative Filteralgorithmus. Einer der Gründe, warum ich TensorFlow gerne für Vorträge wie diese verwende, ist, dass man für viele Anwendungen zur Implementierung des Gradientenabstiegs die Ableitungen der Kostenfunktion finden muss, TensorFlow jedoch automatisch für Sie herausfinden kann, welche Kosten-Ableitungen.

Alles, was Sie tun müssen, ist, die Kostenfunktion zu implementieren, und ohne irgendwelche Berechnungen beherrschen zu müssen, ohne selbst Ableitungen vornehmen zu müssen, können Sie TensorFlow mit nur wenigen Codezeilen dazu bringen, diesen Ableitungsterm zu berechnen, der zur Optimierung der Kosten verwendet werden kann Funktion. Werfen wir einen Blick darauf, wie das alles funktioniert.



Vielleicht erinnern Sie sich an dieses Diagramm hier rechts aus dem ersten Kurs. Dies ist genau das Diagramm, das wir uns angesehen hatten, als wir über die Optimierung von w sprachen. Als wir unser erstes lineares Regressionsbeispiel durcharbeiteten. Und damals hatten wir b=0 gesetzt. Und so sagte das Modell nur f(x)= wx voraus. Und wir wollten den Wert von w finden, der die Kostenfunktion J minimiert. Das haben wir also über eine Gradientenabstiegsaktualisierung gemacht, die so aussah, wobei w wiederholt aktualisiert wird als w minus der Lernrate Alpha multipliziert mit der Ableitung Begriff. Wenn Sie auch b aktualisieren, verwenden Sie diesen Ausdruck. Aber wenn Sie b=0 sagen, verzichten Sie einfach auf die zweite Aktualisierung und führen diese Gradientenabstiegsaktualisierung bis zur Konvergenz weiter durch. Manchmal kann es schwierig sein, diesen Ableitungs- oder partiellen Ableitungsterm zu berechnen. Und es stellt sich heraus, dass TensorFlow dabei helfen kann. Mal sehen, wie. Ich werde eine sehr einfache Kostenfunktion J =(wx-1) im Quadrat verwenden. Also ist wx unsere vereinfachte fw von x und y ist gleich 1. Und das wäre

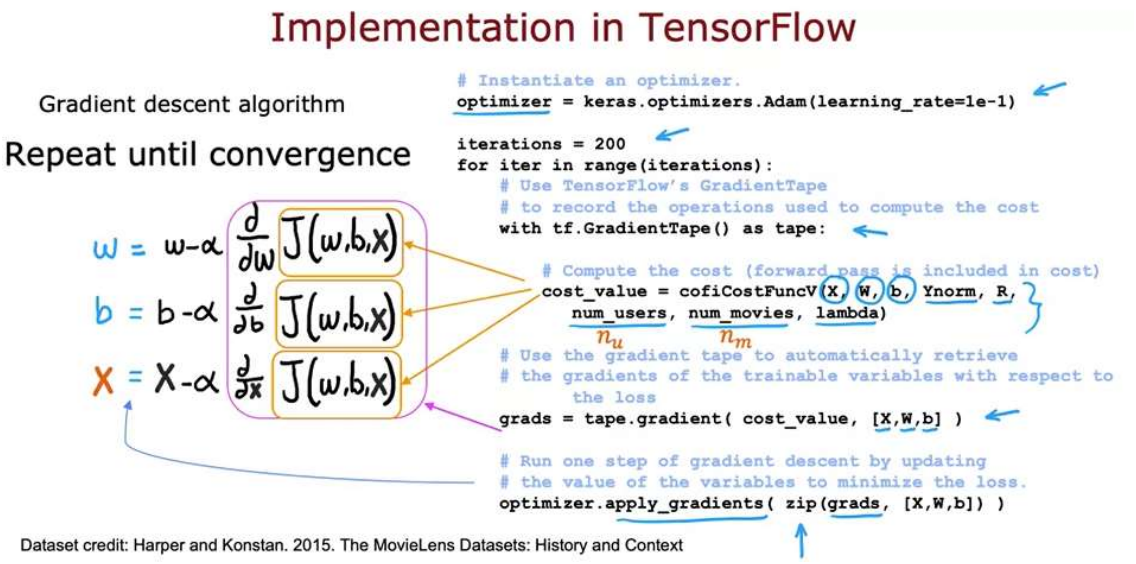
die Kostenfunktion, wenn f(x) gleich wx, y gleich 1 für das eine Trainingsbeispiel wäre, das wir haben, und wenn das nicht der Fall wäre Optimierung dieser Hinsicht auf b. Daher wird der Gradientenabstiegsalgorithmus diese Aktualisierung hier bis zur Konvergenz wiederholen. Es stellt sich heraus, dass TensorFlow diesen Ableitungsterm automatisch für Sie berechnen kann, wenn Sie hier die Kostenfunktion J implementieren, und dadurch den Gradientenabstieg zum Laufen bringen kann. Ich gebe Ihnen einen allgemeinen Überblick über die Funktionsweise dieses Codes, w= tf.variable (3.0). Nimmt den Parameter w und initialisiert ihn auf den Wert 3,0. Indem wir TensorFlow mitteilen, dass w eine Variable ist, teilen wir ihm mit, dass w ein Parameter ist, den wir optimieren möchten. Ich werde x=1,0, y=1,0 und die Lernrate Alpha auf 0,01 setzen. Und lassen Sie uns 30 Iterationen lang einen Gradientendissens ausführen. In diesem Code reicht es also immer noch für Iterationen in Bereichsiterationen, also für 30 Iterationen. Und das ist die Syntax, mit der TensorFlow die Rotoren automatisch für Sie berechnet.



TensorFlow verfügt über eine Funktion namens „Gradient Tape“. Und wenn Sie dies mit unserem „Gradient Tape“ als Tape f schreiben. Dies bedeutet, f(x) als w*x zu berechnen und J als f(x)-y im Quadrat zu berechnen. Indem Sie TensorFlow dann mitteilen, wie costJ berechnet werden soll, und dies mit der Gradient-Taped-Syntax wie folgt tun, zeichnet TensorFlow automatisch die Abfolge der Schritte auf. Die Abfolge der Operationen, die zur Berechnung der KostenJ erforderlich sind. Und dies ist notwendig, um eine automatische Differenzierung zu ermöglichen. Als nächstes speichert TensorFlow die Abfolge der Vorgänge auf Tape, im „Gradient Tape“. Und mit dieser Syntax berechnet TensorFlow automatisch diesen abgeleiteten Term, den ich dJdw nennen werde. Und TensorFlow weiß, dass Sie die entsprechende Ableitung w nehmen möchten. Das w ist der Parameter, den Sie optimieren möchten, weil Sie es hier oben angegeben haben. Und weil wir es hier unten auch präzisieren. Nun zu den Computerableitungen. Schließlich können Sie dieses Update durchführen, indem Sie w nehmen und davon die Lernrate Alpha multipliziert mit dem Ableitungsterm subtrahieren, den wir gerade oben erhalten haben.

TensorFlow- Variablen und Tier-Variablen erfordern eine besondere Behandlung. Aus diesem Grund verwenden wir diese zugewiesene Additionsfunktion, anstatt w wie üblich auf w minus Alpha mal Ableitung zu setzen. Aber wenn Sie im Übungslabor ankommen, machen Sie sich darüber keine Sorgen. Wir stellen Ihnen die gesamte Syntax zur Verfügung, die Sie zur korrekten Implementierung des Sicherheitenfilteralgorithmus benötigen. Beachten Sie also, dass die Hauptarbeit, die Sie mit der „Gradient Tape“ Funktion von TensorFlow leisten müssen, darin besteht, ihm mitzuteilen, wie die

Kostenfunktion J berechnet werden soll. Und der Rest der Syntax bewirkt, dass TensorFlow automatisch für Sie herausfindet, was diese Ableitung ist? Und mit diesem TensorFlow beginnen wir damit, dessen Steigung bei 3 zu ermitteln, die durch diese gestrichelte Linie dargestellt ist. Machen Sie einen Gradientenschritt und aktualisieren Sie w, berechnen Sie die Ableitung erneut und aktualisieren Sie w immer wieder, bis schließlich der optimale Wert von w erreicht wird, der bei w gleich 1 liegt. Mit diesem Verfahren können Sie also einen Gradientenabstieg implementieren, ohne jemals eine Berechnung vornehmen zu müssen Finden Sie selbst heraus, wie Sie diesen Ableitungsterm berechnen. Dies ist eine sehr leistungsstarke Funktion von TensorFlow namens Auto Diff. Und einige andere Pakete für maschinelles Lernen wie Pytorch unterstützen ebenfalls Auto Diff. Manchmal hört man dies Auto Grad nennen. Der technisch korrekte Begriff ist Auto Diff, und Auto Grad ist eigentlich der Name des spezifischen Softwarepakets zur automatischen Differenzierung, zur automatischen Bildung von Ableitungen. Aber manchmal, wenn jemand von Auto Grad spricht, bezieht er sich einfach auf dasselbe Konzept der automatischen Übernahme von Derivaten. Schauen wir uns also an, wie Sie mit Auto Diff einen kollaborativen Filteralgorithmus implementieren können. Und tatsächlich sind Sie, sobald Sie Ableitungen automatisch berechnen können, nicht nur auf den Gradientenabstieg beschränkt. Sie können auch einen leistungsfähigeren Optimierungsalgorithmus verwenden, z. B. den Adam- Optimierungsalgorithmus. Um den kollaborativen Filteralgorithmus TensorFlow zu implementieren, können Sie diese Syntax verwenden. Beginnen wir mit der Angabe, dass der Optimierer Keras Optimizers Adam mit der hier angegebenen Lernrate ist. Und dann, sagen wir mal, 200 Iterationen, hier ist die Syntax wie zuvor mit tf- Gradientenband, s-Band, Sie müssen Code bereitstellen, um den Wert der Kostenfunktion J zu berechnen. Denken Sie also daran, dass bei der kollaborativen Filterung die Kostenfunktion J als Eingabe verwendet wird Parameter x, w und b sowie die Bewertungen bedeuten normalisiert. Deshalb schreibe ich y norm, r(i,j) und gebe an, welche Werte eine Bewertung, die Anzahl der Benutzer oder nu in unserer Notation, die Anzahl der Filme oder nm in unserer Notation oder einfach nur num sowie den Regularisierungsparameter Lambda haben. Und wenn Sie diese Kostenfunktion J implementieren können, dann wird diese Syntax dazu führen, dass TensorFlow die Ableitungen für Sie ermittelt. Diese Syntax veranlasst TensorFlow dann, die Abfolge der Operationen aufzuzeichnen, die zur Berechnung der Kosten verwendet werden. Wenn Sie es dann bitten, Ihnen grads equal tape.gradient zu geben, erhalten Sie die Ableitung der Kostenfunktion nach x, w und b. Und schließlich mit dem Optimierer, den wir oben angegeben hatten, als Adam- Optimierer. Sie können den Optimierer mit den gerade berechneten Verläufen verwenden. Und funktioniert es in Python? Es handelt sich lediglich um eine Funktion, die die Zahlen in eine geeignete Reihenfolge für die angewendete Verlaufsfunktion umordnet. Wenn Sie den Gradientenabstieg zur Sicherheitenfilterung verwenden, denken Sie daran, dass die Kostenfunktion J eine Funktion von w, b und x wäre. Und wenn Sie einen Gradientenabstieg anwenden, nehmen Sie die partielle Ableitung unter Berücksichtigung des w. Und dann war das Update wie folgt. Und Sie würden auch die partielle Ableitung dieser Beziehung nach b nehmen. Und aktualisieren Sie b wie folgt.



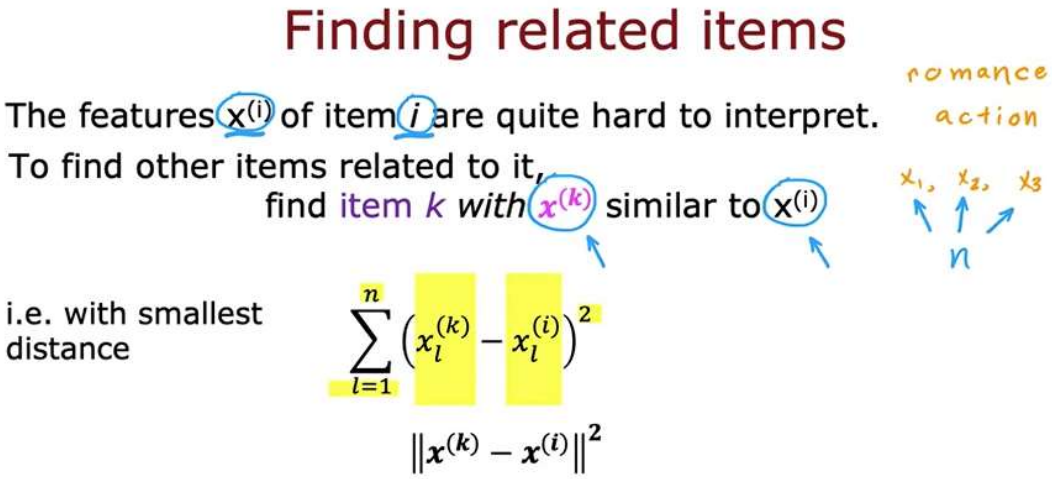
Und aktualisieren Sie auf ähnliche Weise die Funktionen x wie folgt. Und Sie wiederholen bis zur Konvergenz. Aber wie ich bereits erwähnt habe, sind Sie mit TensorFlow und Auto Diff nicht nur auf den Gradientenabstieg beschränkt. Sie können auch einen leistungsfähigeren Optimierungsalgorithmus wie den Adam- Optimierer verwenden. Der Datensatz, den Sie im Übungslabor verwenden, ist ein realer Datensatz, der echte Filme umfasst, die von echten Menschen bewertet wurden.

Dies ist der Filmobjektiv-Datensatz und er stammt von Harper und Konstan. Und ich wünsche Ihnen viel Spaß beim Ausführen dieses Algorithmus anhand eines echten Datensatzes von Filmen und Bewertungen und überzeugen Sie sich selbst von den Ergebnissen, die dieser Algorithmus erzielen kann. Das war's. So können Sie den kollaborativen Filteralgorithmus in TensorFlow implementieren. Wenn Sie sich fragen, warum wir das so machen müssen? Warum konnten wir nicht eine dichte Schicht verwenden und dann den Modell-Compiler und das Modell anpassen? Der Grund, warum wir dieses alte Rezept nicht verwenden konnten, ist, dass der Sicherheitenfilteralgorithmus und die Kostenfunktion nicht genau in die dichte Schicht oder die anderen standardmäßigen neuronalen Netzwerkschichttypen von TensorFlow passen. Deshalb mussten wir es auf eine andere Art und Weise implementieren, bei der wir die Kostenfunktion selbst implementieren würden. Nutzen Sie dann aber die Tools von TensorFlow zur automatischen Differenzierung, auch Auto Diff genannt. Und nutzen Sie TensorFlows Implementierung des Adam- Optimierungsalgorithmus, damit dieser einen Großteil der Arbeit zur Optimierung der Kostenfunktion für uns erledigt. Wenn das Modell, über das Sie verfügen, eine Folge dichter neuronaler Netzwerkschichten oder anderer von TensorFlow unterstützter Schichttypen ist und das alte Implementierungsrezept der Modell-Compound-Model-Anpassung funktioniert. Aber selbst wenn dies nicht der Fall ist, bieten Ihnen diese TensoFlow- Tools eine sehr effektive Möglichkeit, auch andere Lernalgorithmen zu implementieren. Ich wünsche Ihnen daher viel Spaß beim Spielen mit der Sicherheitenfilterungsübung im Übungslabor dieser Woche. Und es sieht so aus, als gäbe es viel Code und viel Syntax, machen Sie sich darüber keine Sorgen. Stellen Sie sicher, dass Sie alles haben, was Sie brauchen, um diese Übung erfolgreich abzuschließen. Und im nächsten Video möchte ich auch näher auf die Nuancen der Sicherheitenfilterung eingehen und insbesondere auf die Frage, wie man bei einem bestimmten Film verwandte Elemente findet oder ob es andere Filme gibt, die diesem ähnlich sind. Kommen wir zum nächsten Video

Verwandte Artikel finden

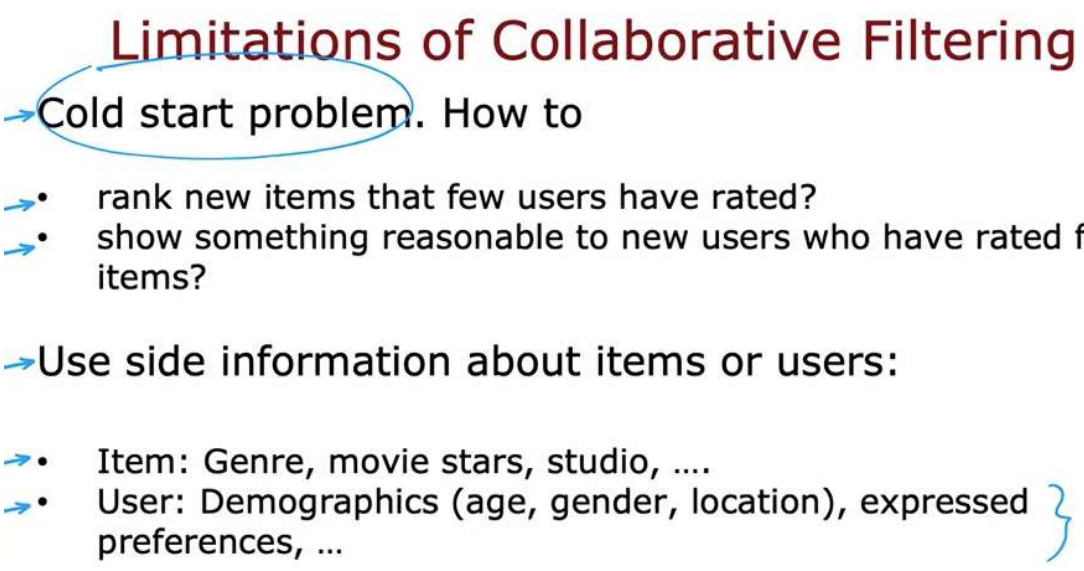
Wenn Sie auf eine Online-Shopping-Website kommen und sich einen bestimmten Artikel ansehen, zum Beispiel ein bestimmtes Buch, zeigt Ihnen die Website möglicherweise Dinge wie „Hier sind

einige andere Bücher, die diesem ähneln“ oder wenn Sie in einem stöbern In einem bestimmten Film heißt es möglicherweise: „Hier sind einige andere Filme, die diesem ähnlich sind.“ Wie machen die Websites das ? Wenn Sie sich also einen Artikel ansehen, werden Ihnen andere ähnliche oder verwandte Artikel angezeigt, die Sie in Betracht ziehen sollten. Es stellt sich heraus, dass der kollaborative Filteralgorithmus, über den wir gesprochen haben, Ihnen eine gute Möglichkeit bietet, verwandte Elemente zu finden. Lass uns einen Blick darauf werfen.



Im Rahmen der von uns besprochenen kollaborativen Filterung haben Sie Funktionen x ^(i) für jedes Element i, für jeden Film i oder eine andere Art von Element kennengelernt, die sie Benutzern empfehlen. Anfang dieser Woche hatte ich ein hypothetisches Beispiel für die Merkmale verwendet, die darstellen, wie sehr ein Film ein Liebesfilm und nicht ein Actionfilm ist. Wenn Sie in der Praxis diesen Algorithmus zum automatischen Lernen der Merkmale x ^(i) verwenden und sich die einzelnen Merkmale x_1, x_2, x_3 ansehen, stellen Sie fest, dass diese recht schwer zu interpretieren sind. Es ist ziemlich schwierig, Funktionen zu lernen und zu sagen, dass x_1 ein Actionfilm und x_2 ein ausländischer Film ist und so weiter. Aber nichtsdestotrotz vermitteln diese erlernten Funktionen, zusammen x_1, x_2, x_3 und viele andere Funktionen, und Sie haben zusammengenommen, etwas darüber, wie dieser Film ist. Es stellt sich heraus, dass bei gegebenen Merkmalen x^(i) von Element i, wenn Sie andere Elemente finden möchten, beispielsweise andere Filme, die sich auf Film i beziehen, Sie versuchen können, das Element k mit den Merkmalen x^(k) zu finden. das ist ähnlich wie x^(i). Insbesondere bei gegebenem Merkmalsvektor x ^(k) bestimmen wir, was dem Merkmal x^(i) als ähnlich bezeichnet wird, wie folgt: Ist die Summe von l gleich 1 bis n mit n Merkmalen von x^(k)_l minus x^(i)_l Quadrat. Es stellt sich heraus, dass dies der quadrierte Abstand zwischen x ^(k) und x^(i) ist, und in der Mathematik wird dieser quadrierte Abstand zwischen diesen beiden Vektoren x^(k) und x^(i) manchmal wie folgt geschrieben Also. Wenn Sie nicht nur den einen Film mit dem kleinsten Abstand zwischen x^(k) und x^(i) finden, sondern beispielsweise die fünf oder zehn Elemente mit den ähnlichsten Merkmalsvektoren, dann finden Sie am Ende fünf oder zehn verwandte Elemente zum Element x^(i). Wenn Sie eine Website erstellen und Benutzern helfen möchten, verwandte Produkte zu einem bestimmten Produkt zu finden, das sie sich ansehen, wäre dies eine gute Möglichkeit, da die Funktionen x^(i) einen Eindruck davon vermitteln, um welchen Artikel es sich bei i handelt, werden andere Elemente x^(k) mit ähnlichen Merkmalen dem Element i ähnlich sein. Später in dieser Woche stellt sich heraus, dass diese Idee, verwandte Elemente zu finden, ein kleiner Baustein sein wird, den wir nutzen werden, um auch zu einem noch leistungsfähigeren empfohlenen System zu gelangen. Bevor ich diesen Abschnitt abschließe, möchte ich einige Einschränkungen der kollaborativen Filterung erwähnen. Bei der kollaborativen Filterung

haben Sie eine Reihe von Elementen und daher haben die Benutzer und Benutzer eine Teilmenge der Elemente bewertet. Eine dieser Schwächen besteht darin, dass das Kaltstartproblem nicht sehr gut gelöst ist. Wenn es beispielsweise einen neuen Artikel in Ihrem Katalog gibt, sagen wir, jemand hat gerade einen neuen Film veröffentlicht und kaum jemand hat diesen Film bisher bewertet, wie ordnen Sie dann den neuen Artikel ein, wenn nur sehr wenige Benutzer ihn zuvor bewertet haben? Und wie können wir neuen Benutzern, die nur wenige Artikel bewertet haben, sicherstellen, dass wir ihnen etwas Vernünftiges zeigen? Wir konnten in einem früheren Video sehen, wie die Mittelwertnormalisierung dabei helfen kann, und sie hilft wirklich sehr. Aber vielleicht gibt es noch bessere Möglichkeiten, Benutzern, die nur sehr wenige Artikel bewertet haben, Dinge zu zeigen, die sie wahrscheinlich interessieren. Dies wird als Kaltstartproblem bezeichnet, denn wenn Sie einen neuen Artikel haben und nur wenige Benutzer ihn bewertet haben, oder wenn wir einen neuen Benutzer haben, der nur sehr wenige Artikel bewertet hat, sind die Ergebnisse der gemeinsamen Filterung für diesen Artikel oder für diesen Benutzer möglicherweise nicht stimmend sehr akkurat. Die zweite Einschränkung der kollaborativen Filterung besteht darin, dass Sie keine natürliche Möglichkeit haben, Nebeninformationen oder zusätzliche Informationen über Elemente oder Benutzer zu verwenden. Beispielsweise wissen Sie für einen bestimmten Film in Ihrem Katalog möglicherweise, welches Genre der Film hat, wer die Hauptrollen in dem Film hatte, ob es sich um ein Studio handelt, wie hoch das Budget ist und so weiter. Möglicherweise haben Sie viele Features zu einem bestimmten Film. Für einen einzelnen Benutzer wissen Sie möglicherweise etwas über seine demografischen Merkmale, z. B. Alter, Geschlecht und Standort. Sie drücken Präferenzen aus, beispielsweise wenn sie Ihnen mitteilen, dass sie bestimmte Filmgenres mögen, andere jedoch nicht, oder wenn Sie die IP-Adresse des Benutzers kennen, kann Ihnen das viel über den Standort eines Benutzers verraten, und wenn Sie den Standort des Benutzers kennen, kann dies möglicherweise der Fall sein Sie helfen Ihnen auch dabei, zu erraten, woran der Benutzer interessiert sein könnte, oder ob Sie wissen, ob der Benutzer über ein Mobilgerät oder einen Desktop auf Ihre Website zugreift, oder ob Sie wissen, welchen Webbrowser er verwendet. Es stellt sich heraus, dass dies alles kleine Hinweise sind, die man bekommen kann.

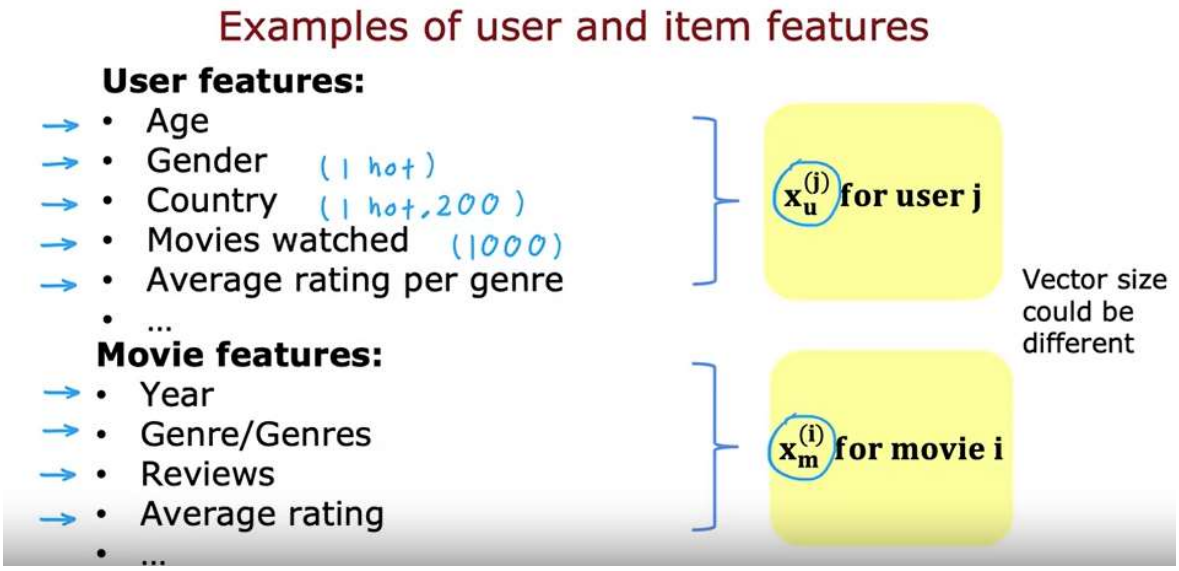


Sie können überraschenderweise mit den Vorlieben eines Benutzers korrelieren. Es stellt sich übrigens heraus, dass bekannt ist, dass sich Benutzer, die Chrome, Firefox, Safari oder Microsoft Edge verwenden, tatsächlich sehr unterschiedlich verhalten. Selbst wenn Sie den Webbrowser des Benutzers kennen, können Sie einen Hinweis darauf erhalten, ob Sie genügend Daten darüber gesammelt haben, was diesem bestimmten Benutzer gefallen könnte. Obwohl es sich bei der

kollaborativen Filterung, bei der mehrere Benutzer mehrere Artikel bewerten, um einen sehr leistungsfähigen Satz von Algorithmen handelt, gibt es auch einige Einschränkungen. Im nächsten Video entwickeln wir inhaltsbasierte Filteralgorithmen, mit denen viele dieser Einschränkungen behoben werden können. Inhaltsbasierte Filteralgorithmen sind eine gängige Technik, die heutzutage in vielen kommerziellen Anwendungen eingesetzt wird. Werfen wir einen Blick darauf, wie sie funktionieren.

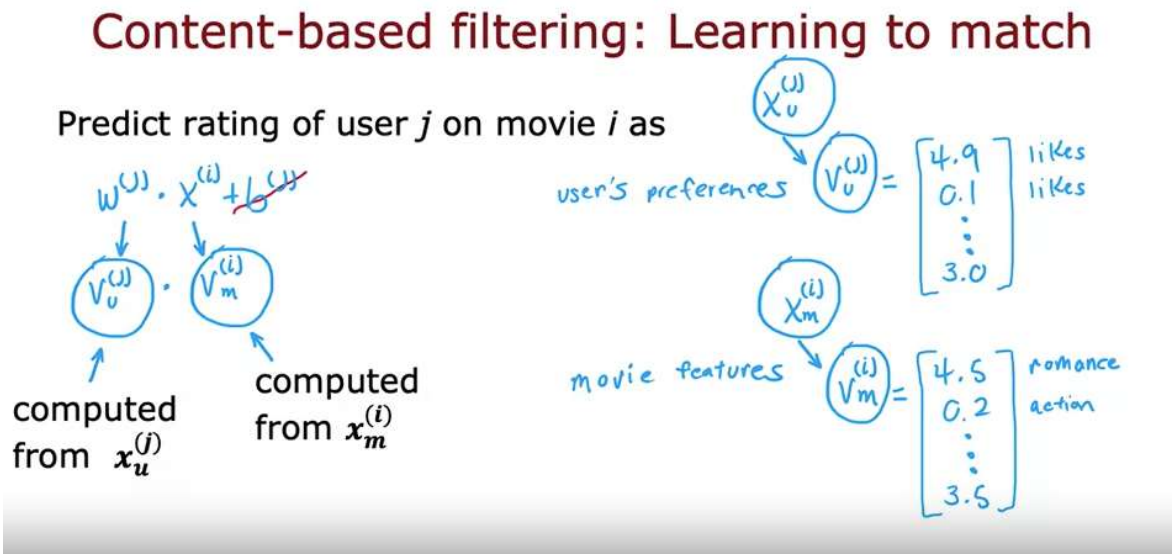
Kollaboratives Filtern vs. inhaltsbasiertes Filtern

In diesem Video beginnen wir mit der Entwicklung einer zweiten Art von Empfehlungssystem, einem sogenannten inhaltsbasierten Filteralgorithmus. Vergleichen und kontrastieren wir zunächst den kollaborativen Filteransatz, den wir bisher betrachten, mit diesem neuen inhaltsbasierten Filteransatz. Lass uns einen Blick darauf werfen. Bei der kollaborativen Filterung besteht der allgemeine Ansatz darin, dass wir Ihnen Artikel basierend auf den Bewertungen von Benutzern empfehlen, die ähnliche Bewertungen wie Sie abgegeben haben. Wir haben einige Benutzer, die einige Bewertungen für einige Artikel abgeben, und der Algorithmus findet heraus, wie er diese nutzen kann, um Ihnen neue Artikel zu empfehlen. Im Gegensatz dazu verfolgt die inhaltsbasierte Filterung einen anderen Ansatz bei der Entscheidung, was Ihnen empfohlen werden soll. Ein inhaltsbasierter Filteralgorithmus empfiehlt Ihnen Artikel basierend auf den Merkmalen der Benutzer und den Merkmalen der Artikel, um eine gute Übereinstimmung zu finden. Mit anderen Worten: Es erfordert einige Merkmale jedes Benutzers sowie einige Merkmale jedes Elements und verwendet diese Funktionen, um zu entscheiden, welche Elemente und Benutzer gut zueinander passen könnten.



Mit einem inhaltsbasierten Filteralgorithmus verfügen Sie immer noch über Daten, in denen Benutzer einige Artikel bewertet haben. Nun, die inhaltsbasierte Filterung verwendet weiterhin r, i, j, um anzugeben, ob Benutzer j den Artikel i bewertet hat oder nicht, und verwendet weiterhin y i, j, um die Bewertung anzugeben, die Benutzer j dem Artikel i gegeben hat, sofern diese definiert ist. Der Schlüssel zur inhaltsbasierten Filterung liegt jedoch darin, dass wir Funktionen des Benutzers und der Elemente sinnvoll nutzen können, um bessere Übereinstimmungen zu finden, als dies möglicherweise mit einem rein kollaborativen Filteransatz möglich wäre. Werfen wir einen Blick darauf, wie das funktioniert. Im Falle von Filmempfehlungen finden Sie hier einige Beispiele für Funktionen. Möglicherweise kennen Sie das Alter des Benutzers oder das Geschlecht des Benutzers. Dies könnte ein One-Hot-Feature sein, ähnlich dem, was Sie gesehen haben, als wir über

Entscheidungsbäume gesprochen haben, wobei Sie ein One-Hot-Feature mit Werten haben könnten, die darauf basieren, ob das selbstidentifizierte Geschlecht des Benutzers männlich oder weiblich oder unbekannt ist, und Sie kennt möglicherweise das Land des Benutzers. Wenn es etwa 200 Länder auf der Welt gibt, dann muss es sich auch um ein One-Hot-Feature mit etwa 200 möglichen Werten handeln. Sie können sich auch frühere Verhaltensweisen des Benutzers ansehen, um diesen Merkmalsvektor zu erstellen. Wenn Sie sich beispielsweise die tausend besten Filme in Ihrem Katalog ansehen, können Sie tausend Funktionen erstellen, die Ihnen sagen, welche der tausend beliebtesten Filme der Welt der Benutzer ansieht. Tatsächlich können Sie auch Bewertungen verwenden, die der Benutzer möglicherweise bereits abgegeben hat, um neue Funktionen zu konstruieren. Es stellt sich heraus, dass, wenn Sie eine Reihe von Filmen haben und wissen, zu welchem Genre jeder Film gehört, die durchschnittliche Bewertung pro Genre ist, die der Benutzer gegeben hat. Wie hoch war die durchschnittliche Bewertung aller Liebesfilme, die der Benutzer bewertet hat? Wie hoch war die durchschnittliche Bewertung aller Actionfilme, die der Benutzer bewertet hat? Und so weiter für alle anderen Genres. Auch dies kann eine leistungsstarke Funktion zur Beschreibung des Benutzers sein. Das Interessante an dieser Funktion ist, dass sie tatsächlich von den Bewertungen abhängt, die der Benutzer abgegeben hat. Aber daran ist nichts auszusetzen. Die Konstruktion eines Merkmalsvektors, der von den Bewertungen des Benutzers abhängt, ist eine völlig gute Möglichkeit, einen Merkmalsvektor zur Beschreibung dieses Benutzers zu entwickeln. Mit Features wie diesen können Sie dann einen Feature-Vektor x mit dem Index u erstellen und diesen als User-Highscript j für den Benutzer j verwenden.



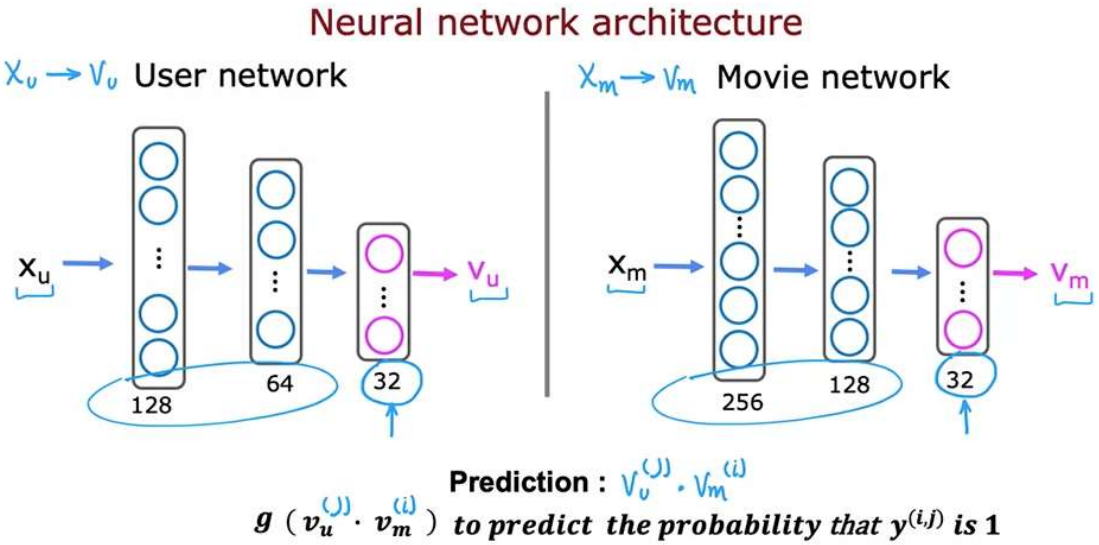
Ebenso können Sie für jeden Film und jedes Element eine Reihe von Merkmalen erstellen, z. B. „Welches war das Jahr des Films?“ Welches Genre oder welche Genres ist der Film bekannt? Wenn es Kritiken zu dem Film gibt, können Sie eine oder mehrere Funktionen erstellen, um etwas darüber festzuhalten, was die Kritiker über den Film sagen. Oder noch einmal: Sie können tatsächlich die Benutzerbewertungen des Films heranziehen, um beispielsweise eine Funktion für die durchschnittliche Bewertung dieses Films zu erstellen. Auch diese Funktion hängt von den Bewertungen ab, die die Benutzer erhalten, aber auch hier ist nichts falsch. Sie können für einen bestimmten Film eine Funktion erstellen, die von den Bewertungen abhängt, die der Film erhalten hat, beispielsweise von der durchschnittlichen Bewertung des Films. Wenn Sie möchten, können Sie auch eine durchschnittliche Bewertung pro Land oder eine durchschnittliche Bewertung pro Benutzergruppe festlegen, da diese auch andere Arten von Filmfunktionen erstellen möchten. Damit können Sie dann für jeden Film einen Merkmalsvektor konstruieren, den ich mit x subscript m bezeichnen werde, m steht für movie und superscript i für movie i. Angesichts solcher Funktionen

besteht die Aufgabe darin, herauszufinden, ob ein bestimmter Film i gut zu Benutzer j passt. Beachten Sie, dass die Benutzerfunktionen und Filmfunktionen sehr unterschiedlich groß sein können. Beispielsweise könnten die Benutzerfunktionen möglicherweise 1500 Nummern umfassen und die Filmfunktionen könnten nur 50 Nummern umfassen. Das ist auch okay. Bei der inhaltsbasierten Filterung werden wir einen Algorithmus entwickeln, der lernt, Benutzer und Filme abzugleichen. Zuvor haben wir die Bewertung von Benutzer j für Film i als wj -Skalarprodukt von xi plus bj vorhergesagt. Um eine inhaltsbasierte Filterung zu entwickeln, werde ich bj loswerden. Es stellt sich heraus, dass dies die Leistung der inhaltsbasierten Filterung überhaupt nicht beeinträchtigt. Anstatt wj für einen Benutzer j und xi für einen Film i zu schreiben, werde ich diese Notation stattdessen einfach durch vj_u ersetzen. Dieses v steht hier für einen Vektor. Es wird eine Liste mit Zahlen angezeigt, die für Benutzer j berechnet wurden, und der Index „u“ steht hier für „Benutzer“. Anstelle von xi werde ich einen separaten Vektor-Index m berechnen, der für den Film steht, und für „Film“ steht ein hochgestellter Index. Vj_u als Vektor als Liste von Zahlen, die aus den Merkmalen des Benutzers j berechnet wurden, und vi_m ist eine Liste von Zahlen, die aus den Merkmalen berechnet wurden, wie Sie sie auf der vorherigen Folie von Film i gesehen haben. Wenn es uns gelingt, eine geeignete Auswahl dieser Vektoren vj_u und vi_m zu treffen, dann ist das Skalarprodukt zwischen diesen beiden Vektoren hoffentlich eine gute Vorhersage der Bewertung, die Benutzer j dem Film i gibt. Veranschaulichen Sie einfach, wozu ein Lernalgorithmus führen könnte. Wenn v, u, das ist ein Benutzervektor, die Präferenzen des Benutzers erfasst, beispielsweise 4,9, 0,1 usw. Solche Zahlenlisten. Die erste Zahl gibt an, wie sehr sie Liebesfilme mögen. Dann erfasst die zweite Zahl, wie sehr sie Actionfilme mögen und so weiter. Dann beträgt v_m, der Filmvektor 4,5, 0,2 usw. Diese Zahlen geben an, wie viel dieser Film ein Liebesfilm, wie viel dieser ein Actionfilm usw. ist. Dann gibt das Skalarprodukt, das diese Zahlenlisten elementweise multipliziert und dann eine Summe bildet, hoffentlich einen Eindruck davon, wie sehr diesem bestimmten Benutzer dieser bestimmte Film gefallen wird. Die Herausforderungen gegebener Merkmale eines Benutzers, sagen wir xj_u, wie können wir diesen Vektor vj_u berechnen, der die Präferenzen des Benutzers prägnant oder kompakt darstellt? Wie können wir bei gegebenen Merkmalen eines Films vi_m berechnen ? Beachten Sie, dass x_u und x_m zwar unterschiedlich groß sein können, eine Liste jedoch sehr lange Zahlenlisten sein kann, eine Liste viel kürzer sein kann und v hier die gleiche Größe haben muss. Denn wenn Sie ein Skalarprodukt zwischen v_u und v_m bilden möchten, müssen beide die gleichen Abmessungen haben, z. B. sind beide beispielsweise 32 Zahlen. Zusammenfassend lässt sich sagen, dass wir bei der kollaborativen Filterung eine Reihe von Benutzern hatten, die verschiedene Artikel bewerteten. Im Gegensatz dazu haben wir bei der inhaltsbasierten Filterung Merkmale von Benutzern und Merkmalen von Elementen und möchten einen Weg finden, gute Übereinstimmungen zwischen den Benutzern und den Elementen zu finden. Wir werden dies tun, indem wir diese Vektoren berechnen, v_u für die Benutzer und v_m für die Elemente über den Filmen, und dann Skalarprodukte zwischen ihnen bilden, um zu versuchen, gute Übereinstimmungen zu finden. Wie berechnen wir v_u und v_m ? Schauen wir uns das im nächsten Video an.

Deep Learning für inhaltsbasierte Filterung

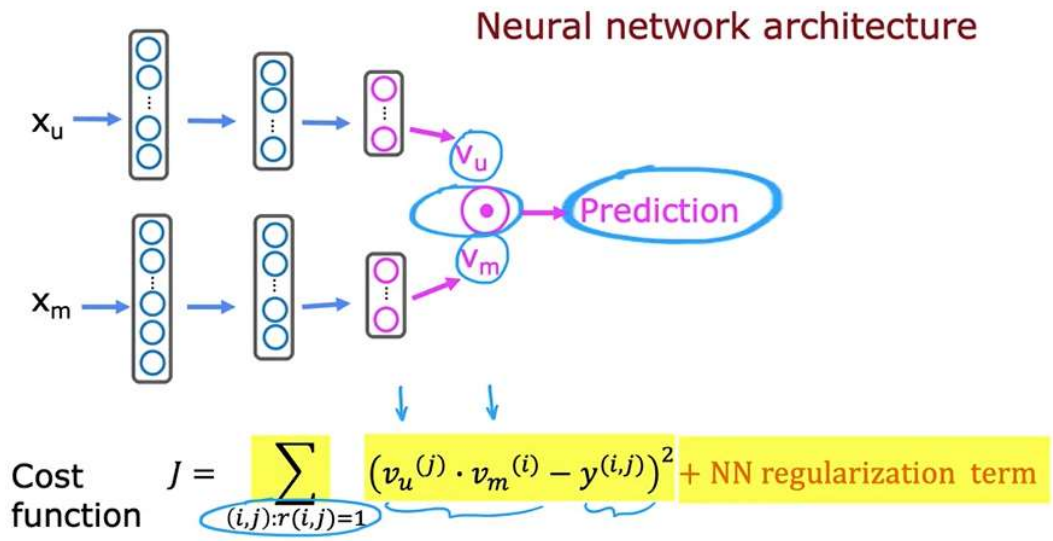
Eine gute Möglichkeit, einen inhaltsbasierten Filteralgorithmus zu entwickeln, ist der Einsatz von Deep Learning. Der Ansatz, den Sie in diesem Video sehen, entspricht der Art und Weise, wie heute viele wichtige kommerzielle, hochmoderne inhaltsbasierte Filteralgorithmen erstellt werden. Lass uns einen Blick darauf werfen. Denken Sie daran, dass wir in unserem Ansatz bei einem gegebenen Merkmalsvektor, der einen Benutzer beschreibt, wie z. B. Alter, Geschlecht, Land usw., den Vektor v_u berechnen müssen, und in ähnlicher Weise müssen wir bei einem gegebenen Vektor, der einen Film beschreibt, wie z. B. das Erscheinungsjahr, Für die Sterne im Film usw. müssen wir einen Vektor v_m berechnen. Um Ersteres zu erreichen, verwenden wir ein neuronales Netzwerk. Das erste neuronale Netzwerk wird das sein, was wir das Benutzernetzwerk nennen. Hier ist ein Beispiel für ein

Benutzernetzwerk, das als Eingabe die Liste der Funktionen des Benutzers x_u verwendet, also das Alter, das Geschlecht, das Land des Benutzers usw. Dann wird unter Verwendung einiger Schichten, beispielsweise Schichten eines dichten neuronalen Netzwerks, dieser Vektor v_u ausgegeben, der den Benutzer beschreibt.

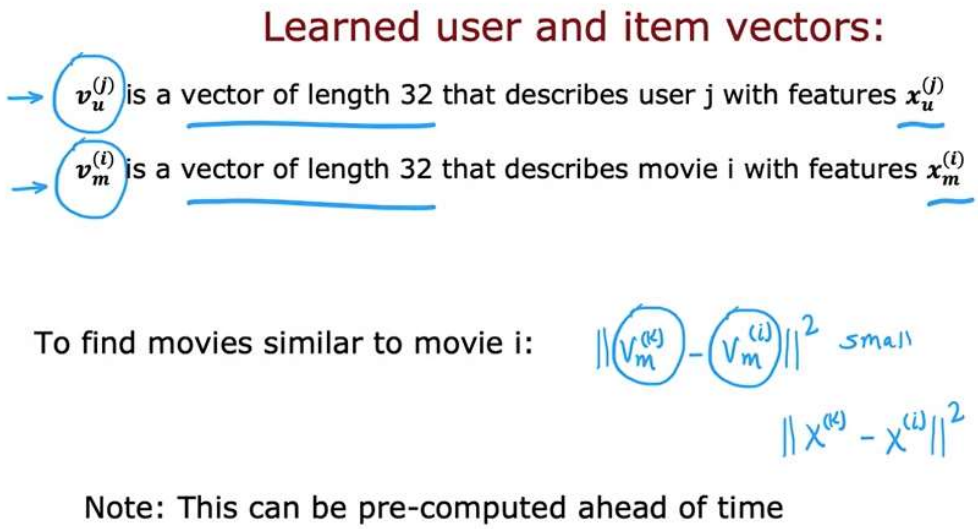


Beachten Sie, dass in diesem neuronalen Netzwerk die Ausgabeschicht 32 Einheiten hat und v_u also tatsächlich eine Liste mit 32 Zahlen ist. Im Gegensatz zu den meisten neuronalen Netzen, die wir zuvor verwendet haben, ist die letzte Schicht keine Schicht mit einer Einheit, sondern eine Schicht mit 32 Einheiten. Um v_m für einen Film zu berechnen, können wir in ähnlicher Weise ein Filmnetzwerk wie folgt verwenden, das Merkmale des Films als Eingabe verwendet und über einige Schichten eines neuronalen Netzwerks v_m ausgibt, den Vektor, der den Film beschreibt. Schließlich werden wir die Bewertung dieses Benutzers für diesen Film als v_ u Punktprodukt mit v_m vorhersagen. Beachten Sie, dass das Benutzernetzwerk und das Filmnetzwerk hypothetisch eine unterschiedliche Anzahl verborgener Schichten und eine unterschiedliche Anzahl von Einheiten pro verborgener Schicht haben können. Alle Ausgabeebenen müssen die gleiche Größe und Abmessung haben. In der Beschreibung, die Sie bisher gesehen haben, haben wir eine Filmbewertung von 1–5 oder 0–5 Sternen vorhergesagt. Wenn wir binäre Beschriftungen hätten, wenn y dem Benutzer ein Element gefallen oder bevorzugen würde, dann können Sie diesen Algorithmus auch so ändern, dass er ausgegeben wird. Anstelle von v_u.v_m können Sie die Sigmoidfunktion darauf anwenden und damit die Wahrscheinlichkeit vorhersagen, dass y^i,j 1 ist. Um diese Notation zu konkretisieren, können wir hier auch hochgestellte Zeichen i und j hinzufügen, wenn wir sie hervorheben möchten dass dies die Vorhersage von Benutzer j zum Film i ist. Ich habe hier das Benutzernetzwerk und das Filmnetzwerk als zwei separate neuronale Netzwerke gezeichnet. Aber es stellt sich heraus, dass wir sie tatsächlich in einem einzigen Diagramm zusammenfassen können, als wäre es ein einzelnes neuronales Netzwerk. So sieht es aus. Im oberen Teil dieses Diagramms sehen wir das Benutzernetzwerk, das x_u eingibt und schließlich v_u berechnet. Im unteren Teil dieses Diagramms sehen wir das, was einmal das Filmnetzwerk war, die Eingabe ist x_m und wird schließlich v_m berechnet, und diese beiden Vektoren werden dann als Skalarprodukt zusammengefügt. Dieser Punkt hier stellt das Skalarprodukt dar, und das gibt uns unsere Vorhersage. Nun, dieses Modell hat viele Parameter. Jede dieser Schichten eines neuronalen Netzwerks verfügt über einen üblichen Satz von Parametern des neuronalen Netzwerks. Wie trainiert man alle Parameter sowohl des Benutzernetzwerks als auch des Filmnetzwerks? Was wir tun werden, ist, eine Kostenfunktion J zu konstruieren, die der Kostenfunktion, die Sie bei der kollaborativen Filterung gesehen haben, sehr ähnlich sein wird. Dabei wird davon ausgegangen, dass Sie einige Daten von einigen Benutzern

haben, die einige Filme bewertet haben, wir Wir werden über alle Paare i und j summieren, in denen Sie Beschriftungen haben, wobei i, j gleich 1 der Differenz zwischen der Vorhersage ist. Das wäre v_u^j Skalarprodukt mit v_m^i minus y^ij im Quadrat.



Die Art und Weise, wie wir dieses Modell trainieren würden, hängt von den Parametern des neuronalen Netzwerks ab. Am Ende ergeben sich hier unterschiedliche Vektoren für die Benutzer und für die Filme. Was wir tun möchten, ist, die Parameter des neuronalen Netzwerks zu trainieren, sodass Sie am Ende Vektoren für die Benutzer und für die Filme erhalten, die zu kleinen quadratischen Fehlern in den Vorhersagen führen, die Sie hier veröffentlichen. Um es klarzustellen: Es gibt kein separates Schulungsverfahren für die Benutzer- und Filmnetzwerke. Dieser Ausdruck hier unten ist die Kostenfunktion, mit der alle Parameter des Benutzers und des Filmnetzwerks trainiert werden. Wir werden die beiden Netzwerke danach beurteilen, wie gut v_u und v_m y^ij vorhersagen, und mit dieser Kostenfunktion werden wir den Gradientenabstieg oder einen anderen Optimierungsalgorithmus verwenden, um die Parameter des neuronalen Netzwerks zu optimieren die Kostenfunktion J soll möglichst klein sein.

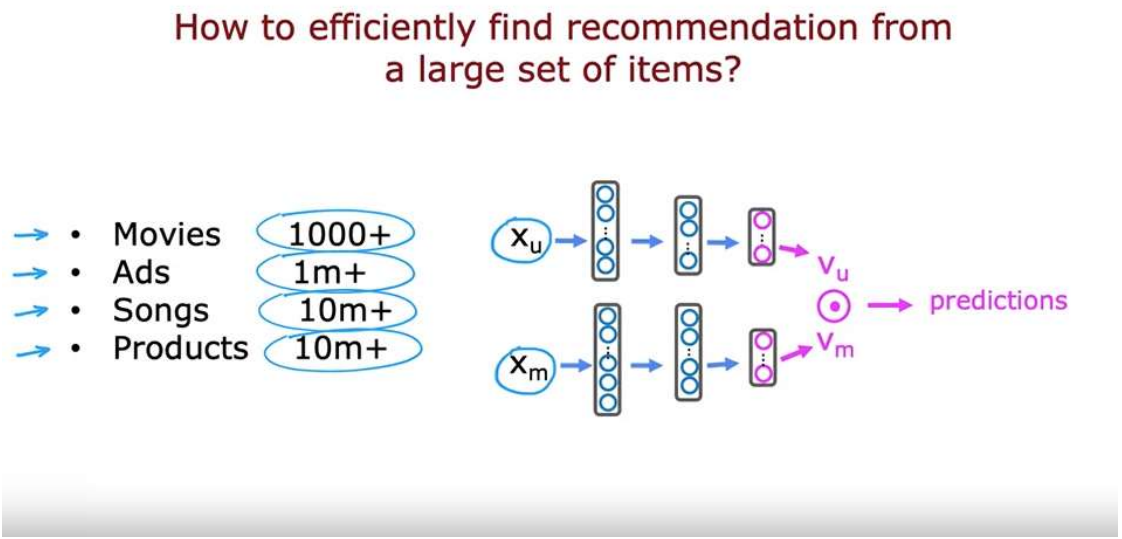


Wenn Sie dieses Modell regulieren möchten, können wir auch den üblichen Regularisierungsterm für neuronale Netze hinzufügen, um die neuronalen Netze dazu zu ermutigen, die Werte ihrer Parameter klein zu halten. Es stellt sich heraus, dass Sie, nachdem Sie dieses Modell trainiert haben, es auch verwenden können, um ähnliche Gegenstände zu finden. Dies ähnelt dem, was wir bei den kollaborativen Filterfunktionen gesehen haben und Ihnen dabei helfen, auch ähnliche Artikel zu finden. Lass uns einen Blick darauf werfen. V_u^j ist ein Vektor der Länge 32, der einen Benutzer j beschreibt, der über Merkmale x_ u^j verfügt. Ebenso ist v^i_m ein Vektor der Länge 32, der hier einen Film mit diesen Merkmalen beschreibt. Was ist, wenn Sie angesichts eines bestimmten Films andere ähnliche Filme finden möchten? Nun, dieser Vektor v^i_m beschreibt den Film i. Wenn Sie andere ähnliche Filme finden möchten, können Sie nach anderen Filmen k suchen, sodass der Abstand zwischen dem Vektor, der den Film k beschreibt, und dem Vektor, der den Film i beschreibt, also der quadrierte Abstand klein ist. Dieser Ausdruck spielt eine ähnliche Rolle wie zuvor bei der kollaborativen Filterung, bei der es darum ging, einen Film mit den Merkmalen x^k zu finden, der den Merkmalen x^i ähnelt. Somit können Sie mit diesem Ansatz auch Artikel finden, die einem bestimmten Artikel ähneln. Ein letzter Hinweis: Dies kann im Voraus berechnet werden. Damit meine ich, dass Sie über Nacht einen Rechenserver betreiben können, um die Liste aller Ihrer Filme zu durchsuchen und für jeden Film ähnliche Filme zu finden, sodass morgen, wenn ein Benutzer auf die Website kommt und sich einen bestimmten Film ansieht Für einen Film können Sie bereits 10 oder 20 ähnlichste Filme vorberechnet haben, um sie dem Benutzer zu diesem Zeitpunkt anzuzeigen. Die Tatsache, dass Sie vorab berechnen können, was einem bestimmten Film ähnelt, wird sich später als wichtig erweisen, wenn wir über die Ausweitung dieses Ansatzes auf einen sehr großen Filmkatalog sprechen. So können Sie mithilfe von Deep Learning einen inhaltsbasierten Filteralgorithmus erstellen. Sie erinnern sich vielleicht, als wir über Entscheidungsbäume und die Vor- und Nachteile von Entscheidungsbäumen im Vergleich zu neuronalen Netzen sprachen. Ich habe erwähnt, dass einer der Vorteile neuronaler Netze darin besteht, dass es einfacher ist, mehrere neuronale Netze zusammenzusetzen, damit sie in der Konsole funktionieren und ein größeres System aufbauen können. Was Sie gerade gesehen haben, war tatsächlich ein Beispiel dafür, bei dem wir ein Benutzernetzwerk und das Filmnetzwerk nehmen und sie zusammenfügen und dann das Innenprodukt der Ausgaben ermitteln konnten. Durch die Fähigkeit, zwei neuronale Netze zusammenzufügen, ist es uns gelungen, eine komplexere Architektur zu entwickeln, die sich als recht leistungsfähig erweist. Wenn man diese Algorithmen in die Praxis umsetzt, stelle ich fest, dass Entwickler oft viel Zeit damit verbringen, sorgfältig die Funktionen zu entwerfen, die für die Einspeisung in diese inhaltsbasierten Filteralgorithmen erforderlich sind. Wenn wir am Ende eines dieser Systeme kommerziell bauen, könnte es sich lohnen, etwas Zeit in die Entwicklung guter Funktionen für diese Anwendung zu investieren. Im Hinblick auf diese Anwendungen besteht eine Einschränkung des von uns beschriebenen Algorithmus darin, dass die Ausführung sehr rechenintensiv sein kann, wenn Sie über einen großen Katalog mit vielen verschiedenen Filmen verfügen, die Sie möglicherweise empfehlen möchten. Im nächsten Video werfen wir einen Blick auf einige praktische Probleme und wie Sie diesen Algorithmus ändern können, um eine Skala zu erstellen, die auch bei sehr großen Artikelkatalogen funktioniert. Schauen wir uns das im nächsten Video an.

Empfehlung aus einem großen Katalog

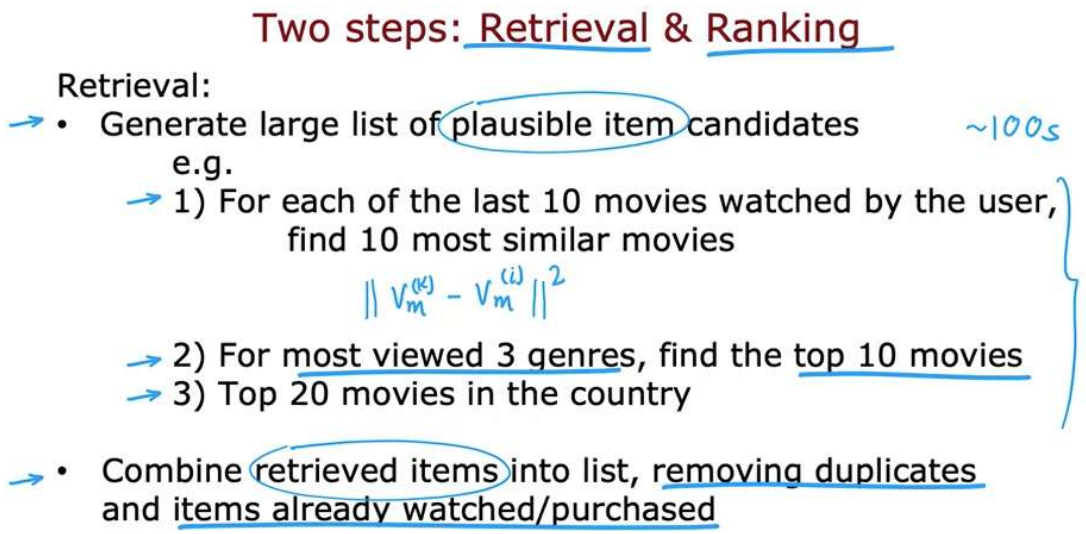
Heutige empfohlene Systeme müssen manchmal eine Handvoll zu empfehlender Elemente auswählen. Aus einem Katalog mit Tausenden oder Millionen oder Zehnmillionen oder sogar mehr Artikeln. Wie man das rechnerisch effizient macht, schauen wir uns mal an. Hier haben wir Ihr Netzwerk genutzt, um Vorhersagen darüber zu treffen, wie ein Benutzer einen Artikel bewerten könnte. Heutzutage verfügt eine große Film-Streaming-Site möglicherweise über Tausende von Filmen oder ein System, das versucht zu entscheiden, welche Werbung geschaltet werden soll.

Möglicherweise steht ein Katalog mit Millionen von Anzeigen zur Auswahl. Oder eine Musik-Streaming-Site bietet eine Auswahl von mehreren zehn Millionen Songs. Und auf großen Online-Shoppingseiten können Millionen oder sogar Zehnmillionen von Produkten zur Auswahl stehen.

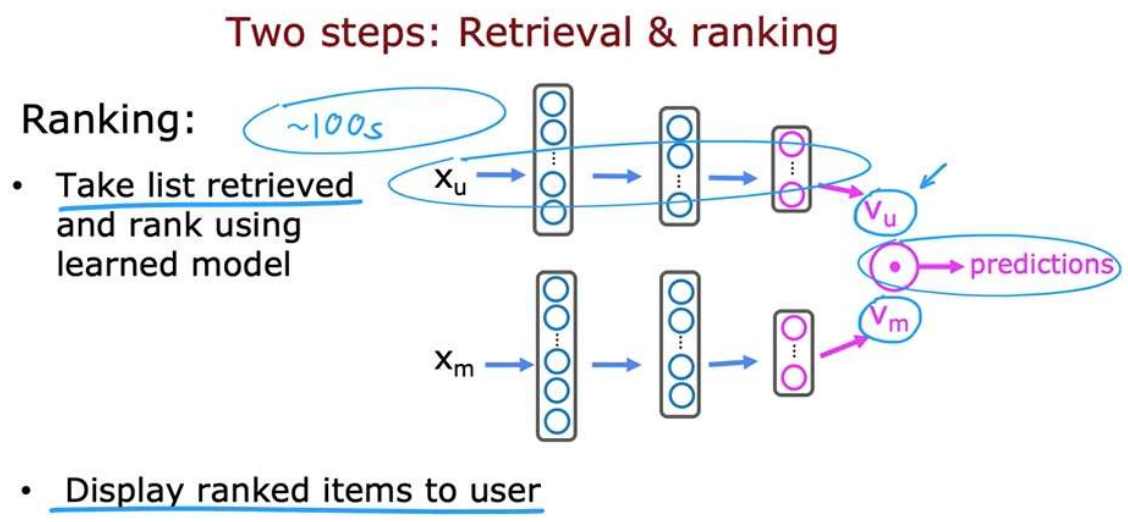


Wenn ein Benutzer auf Ihrer Website auftaucht, verfügt er über einige Funktionen. Aber wenn Sie Tausende von Millionen Elementen durch dieses neuronale Netzwerk laufen lassen müssen, um das Produkt zu berechnen. Um herauszufinden, welche Produkte Sie empfehlen sollten, müssen Sie eine neuronale Netzwerkinferenz durchführen. Tausende Millionen Male jedes Mal, wenn ein Benutzer auf Ihrer Website erscheint, ist rechnerisch nicht durchführbar. Viele von Gesetzen empfohlene Systeme werden in zwei Schritten implementiert, die als Abruf- und Ranking-Schritte bezeichnet werden. Die Idee besteht darin, während des Abrufschritts eine große Liste plausibler Artikelkandidaten zu erstellen. Dadurch wird versucht, viele mögliche Dinge abzudecken, die Sie dem Benutzer empfehlen könnten, und das ist während des Abrufschritts in Ordnung. Wenn Sie viele Artikel einbeziehen, die dem Benutzer wahrscheinlich nicht gefallen, nehmen Sie dann während des Ranking-Schritts eine Feinabstimmung vor und wählen die besten Artikel aus, die Sie dem Benutzer empfehlen möchten. Hier ist ein Beispiel: Während des Abrufschritts könnten wir so etwas tun: Finden Sie für jeden der letzten 10 Filme, die der Benutzer gesehen hat, die 10 ähnlichsten Filme. Das heißt also, wenn ein Benutzer beispielsweise den Film I mit Vector VIM angesehen hat, können Sie mit Vector VKM die Filme finden, die diesem ähneln. Und wie Sie im letzten Video zum Finden ähnlicher Filme gesehen haben, kann der gegebene Film vorab berechnet werden. Nachdem Sie also die ähnlichsten Filme für einen Film vorab berechnet haben, können Sie die Ergebnisse einfach mithilfe einer Nachschlagetabelle abrufen. Dadurch erhalten Sie einen ersten Satz vielleicht einigermaßen plausibler Filme, die Sie Benutzern empfehlen können und die gerade auf Ihrer Website aufgetaucht sind. Darüber hinaus können Sie entscheiden, welche der drei am häufigsten angesehenen Genres des Benutzers hinzugefügt werden sollen. Angenommen, der Benutzer hat viele Liebesfilme, viele Komödien und viele historische Dramen gesehen. Dann würden wir die Top-10-Filme in jedem dieser drei Genres zur Liste möglicher Artikelkandidaten hinzufügen. Und dann fügen wir dieser Liste vielleicht auch die Top-20-Filme im Land des Nutzers hinzu. Dieser Abrufschritt kann also sehr schnell durchgeführt werden und Sie erhalten möglicherweise eine Liste mit 100 oder vielleicht Hunderten plausibler Filme. Dem Benutzer zu empfehlen und hoffentlich wird diese Liste einige gute Optionen empfehlen. Es ist aber auch in Ordnung, wenn es einige Optionen enthält, die dem Benutzer überhaupt nicht gefallen. Das Ziel des Abrufschritts besteht darin, eine breite Abdeckung sicherzustellen, sodass genügend Filme vorhanden sind, zumindest viele gute. Abschließend würden wir dann alle Elemente, die wir während des Abrufschritts abrufen, nehmen

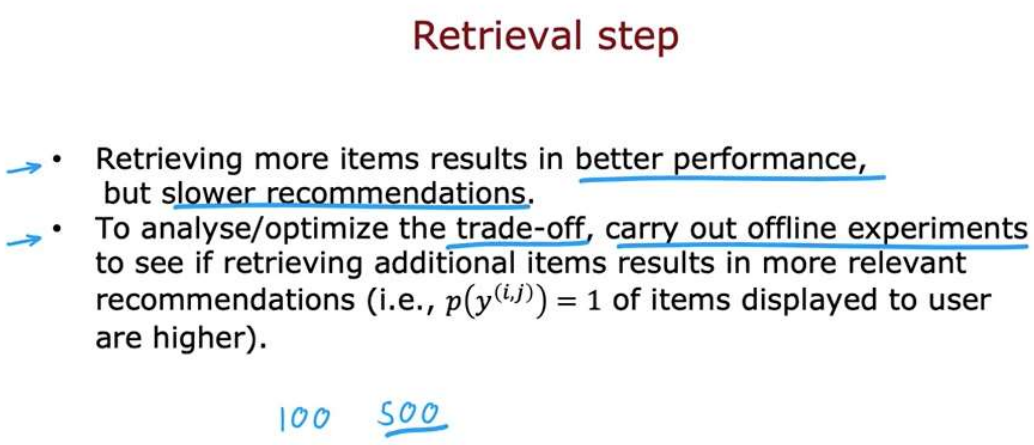
und sie in einer Liste zusammenfassen. Entfernen Sie zwei Herde und entfernen Sie Gegenstände, die der Benutzer bereits gespült hat oder die der Benutzer bereits gekauft hat und die Sie ihm möglicherweise nicht noch einmal empfehlen möchten.



Der zweite Schritt hiervon ist dann der Ranking-Schritt. Beim Ranking-Schritt verwenden Sie die im Abrufschritt abgerufene Liste. Das können also nur Hunderte von möglichen Filmen sein und sie anhand des erlernten Modells einordnen. Und das bedeutet, dass Sie den Benutzer-Feature-Vektor und den Film-Feature-Darsteller in dieses neuronale Netzwerk einspeisen. Berechnen Sie für jedes Benutzerfilmpaar die vorhergesagte Bewertung. Und auf dieser Grundlage haben Sie jetzt alle über 100 Filme, also diejenigen, denen der Benutzer am wahrscheinlichsten eine hohe Bewertung gibt. Und dann können Sie dem Benutzer einfach die Rangliste der Elemente anzeigen, je nachdem, was der Benutzer Ihrer Meinung nach geben wird. Die höchste Bewertung für eine zusätzliche Optimierung ist die, wenn Sie VM berechnet haben. Für alle Filme im Voraus müssen Sie dann nur noch einmal Rückschlüsse auf diesen Teil des neuronalen Netzwerks ziehen, um die VU zu berechnen. Und dann nehmen Sie die VU, die gerade für den Benutzer Ihrer Website berechnet wurde. Und nehmen Sie das innere Produkt zwischen VU und VM. Für die Filme, die Sie während des Abrufschritts abgerufen haben. Daher kann diese Berechnung relativ schnell durchgeführt werden. Wenn der Abrufschritt beispielsweise nur Hunderte von Filmen anzeigt, müssen Sie für dieses Album unter anderem entscheiden, wie viele Elemente Sie während des Abrufschritts abrufen möchten. Zur Einspeisung in den genaueren Ranking-Schritt.



Während des Abrufschritts führt das Abrufen von mehr Elementen tendenziell zu einer besseren Leistung. Aber der Algorithmus wird letztendlich langsamer sein, den Kompromiss zwischen der Anzahl der abzurufenden Elemente zu analysieren oder zu optimieren, um 100, 500 oder 1000 Elemente abzurufen. Ich würde empfehlen, Offline-Experimente durchzuführen, um zu sehen, inwieweit das Abrufen zusätzlicher Elemente zu relevanteren Empfehlungen führt. Und insbesondere, wenn die geschätzte Wahrscheinlichkeit, dass YIJ. Entspricht gemäß Ihrem neuronalen Netzwerkmodell eins. Oder wenn die geschätzte Bewertung Y für die Abrufelemente gemäß der Vorhersage Ihres Modells am Ende viel höher ist. Wenn Sie nur beispielsweise 500 statt nur 100 Artikel abrufen würden, dann würde das dafür sprechen, vielleicht mehr Artikel abzurufen. Auch wenn es das Album etwas verlangsamt. Aber mit dem separaten Abrufschritt und dem Ranking-Schritt können viele heute empfohlene Systeme sowohl schnelle als auch genaue Ergebnisse liefern. Denn der Retrieval-Schritt versucht, viele Elemente herauszuschneiden, die es einfach nicht wert sind, die detailliertere Beeinflussung und das innere Produkt vorzunehmen. Und dann macht der Ranking-Schritt eine genauere Vorhersage darüber, welche Artikel dem Benutzer wahrscheinlich tatsächlich gefallen werden.



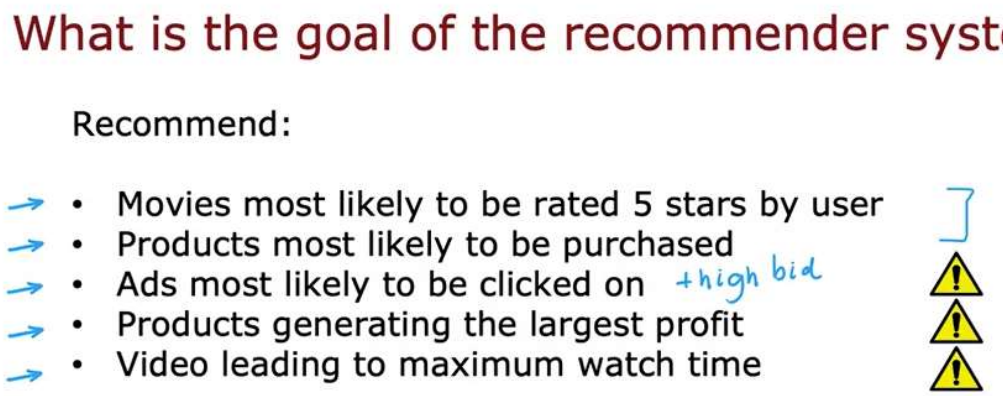
Das war's also. Auf diese Weise sorgen Sie dafür, dass Ihr empfohlenes System auch bei sehr großen Film- oder Produktkatalogen oder was auch immer effizient funktioniert. Nun stellt sich heraus, dass die von uns empfohlenen Systeme nicht nur kommerziell wichtig sind, sondern auch einige

erhebliche ethische Probleme mit ihnen verbunden sind. Und leider gibt es empfohlene Systeme, die Schaden angerichtet haben. Wenn Sie also Ihr eigenes empfohlenes System erstellen, hoffe ich, dass Sie einen ethischen Ansatz verfolgen und es zum Nutzen Ihrer Benutzer nutzen. Und die Gesellschaft ebenso wie Sie selbst und das Unternehmen, für das Sie arbeiten könnten. Werfen wir im nächsten Video einen Blick auf die ethischen Probleme im Zusammenhang mit empfohlenen Systemen

Ethischer Einsatz von Empfehlungssystemen

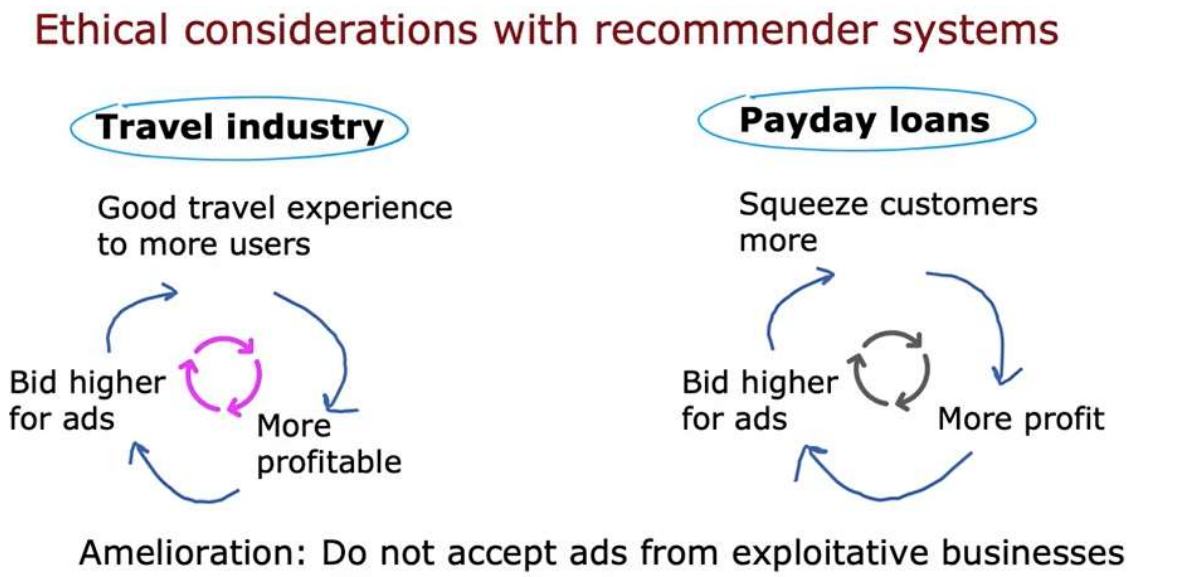
Auch wenn Empfehlungssysteme für einige Unternehmen sehr profitabel waren, kommt es dennoch vor, dass es in einigen Anwendungsfällen zu einer Verschlechterung der Situation der Menschen und der Gesellschaft insgesamt kommt. Wenn Sie jedoch Empfehlungssysteme oder auch andere Lernalgorithmen verwenden, hoffe ich, dass Sie nur Dinge tun, die der Gesellschaft als Ganzes und den Menschen zu einem besseren Leben verhelfen.

Werfen wir einen Blick auf einige der problematischen Anwendungsfälle von Empfehlungssystemen sowie auf Verbesserungen, um den Schaden zu verringern oder den Nutzen zu erhöhen, den sie bewirken können. Wie Sie in den letzten Videos gesehen haben, gibt es viele Möglichkeiten, ein Empfehlungssystem zu konfigurieren. Als wir binäre Labels sahen, könnte das Label y lauten: „Interagiert ein Benutzer oder hat er darauf geklickt oder hat ihm ein Artikel explizit gefallen?“ Beim Entwerfen eines Empfehlungssystems gibt es viele Möglichkeiten, das Ziel des Empfehlungssystems festzulegen und zu entscheiden, was den Benutzern empfohlen werden soll. Sie können beispielsweise entscheiden, Benutzern Filme zu empfehlen, die von diesem Benutzer am wahrscheinlichsten mit fünf Sternen bewertet wurden. Das scheint in Ordnung zu sein. Das scheint eine gute Möglichkeit zu sein, Benutzern Filme zu zeigen, die sie möchten.



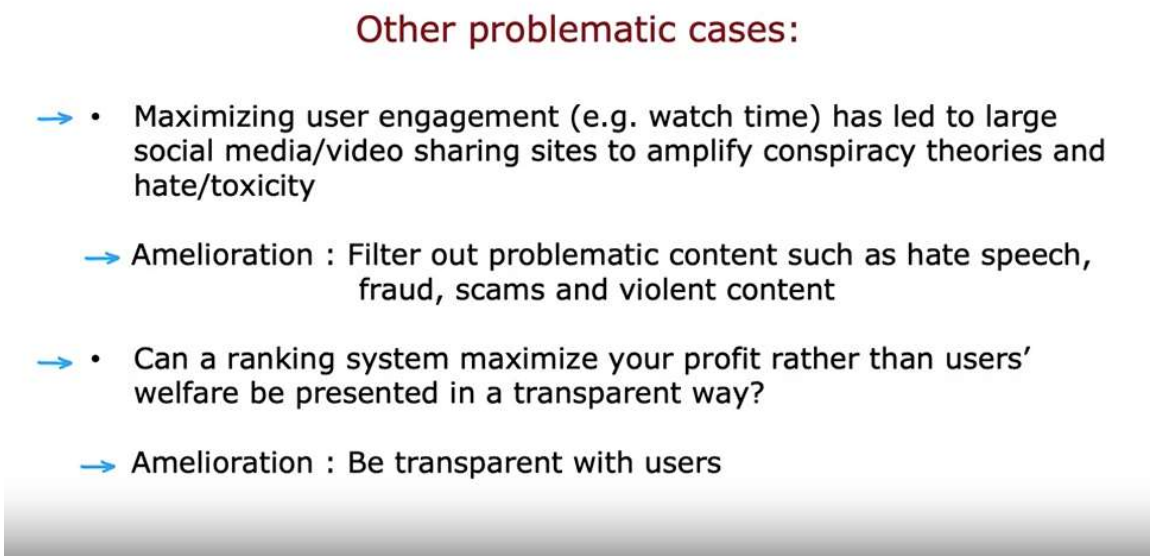
Oder vielleicht können Sie dem Benutzer Produkte empfehlen, die er am wahrscheinlichsten kaufen wird. Das scheint auch eine sehr sinnvolle Verwendung eines Empfehlungssystems zu sein. Versionen von Empfehlungssystemen können auch verwendet werden, um zu entscheiden, welche Anzeigen einem Nutzer gezeigt werden sollen. Eine Sache, die Sie tun könnten, wäre, dem Benutzer eine Empfehlung zu geben oder sie tatsächlich als diejenige anzuzeigen, die am wahrscheinlichsten angeklickt wird. Tatsächlich versuchen viele Unternehmen anzuzeigen, wie wahrscheinlich die Anzeige ist und wo der Werbetreibende ein hohes Gebot abgegeben hat, denn bei vielen Anzeigenmodellen hängt der Umsatz, den das Unternehmen erzielt, davon ab, ob auf die Anzeige geklickt wurde und was der Werbetreibende getan hat hatte ein Gebot pro Klick. Obwohl es sich dabei um eine gewinnmaximierende Strategie handelt, kann diese Art der Werbung auch einige

negative Auswirkungen haben. Auf der nächsten Folie gebe ich ein konkretes Beispiel. Viele Unternehmen versuchen außerdem, Produkte zu empfehlen, die den größten Gewinn erwirtschaften. Wenn Sie heute auf eine Website gehen und nach einem Produkt suchen, gibt es viele Websites, auf denen Ihnen nicht das relevanteste Produkt oder das Produkt angezeigt wird, das Sie am wahrscheinlichsten kaufen werden. Stattdessen wird versucht, Ihnen die Produkte zu zeigen, die dem Unternehmen den größten Gewinn bringen. Wenn ein bestimmtes Produkt für sie profitabler ist, weil sie es günstiger kaufen und zu einem höheren Preis verkaufen können, wird es in den Empfehlungen höher gerankt. Mittlerweile sehen viele Unternehmen einen Druck zur Gewinnmaximierung. Dies scheint keine unvernünftige Vorgehensweise zu sein, aber aus der Sicht des Benutzers empfinden wir es manchmal als hilfreich, wenn eine Website Ihnen ein Produkt empfiehlt, wenn Ihnen die Kriterien transparent gegenüberstehen Es ist die Entscheidung, was man Ihnen zeigt. Versucht es, seine Gewinne zu maximieren, oder versucht es, Ihnen Dinge zu zeigen, die für Sie am nützlichsten sind? Auf Video-Websites oder Social-Media-Websites kann ein Empfehlungssystem auch so modifiziert werden, dass es versucht, Ihnen den Inhalt anzuzeigen, der zur maximalen Wiedergabezeit führt.



Insbesondere Websites, die Werbeeinnahmen erzielen, haben in der Regel einen Anreiz, Sie für lange Zeit auf der Website zu halten. Der Versuch, die Zeit, die Sie auf der Website verbringen, zu maximieren, ist eine Möglichkeit für die Website, mehr von Ihrer Zeit zu gewinnen, damit sie Ihnen mehr Anzeigen zeigen kann. Heutzutage werden Empfehlungssysteme verwendet, um zu versuchen, die Benutzerinteraktion zu maximieren oder die Zeit zu maximieren, die jemand auf einer Website oder einer bestimmten App verbringt. Während die ersten beiden davon recht harmlos erscheinen, sind die dritte, vierte und fünfte möglicherweise völlig in Ordnung. Sie dürfen überhaupt keinen Schaden anrichten. Oder es könnte sich auch um problematische Anwendungsfälle für Empfehlungssysteme handeln. Werfen wir einen genaueren Blick auf einige dieser potenziell problematischen Anwendungsfälle. Lassen Sie mich mit dem Werbebeispiel beginnen. Es stellt sich heraus, dass die Werbebranche manchmal ein Verstärker einiger der schädlichsten Geschäfte sein kann. Sie können auch ein Verstärker für einige der besten und fruchtbarsten Unternehmen sein. Lassen Sie mich das anhand eines guten und eines schlechten Beispiels veranschaulichen. Nehmen wir die Reisebranche. Ich denke, in der Reisebranche besteht der Weg zum Erfolg darin, den Nutzern gute Reiseerlebnisse zu bieten und den Nutzern wirklich zu dienen. Nun stellt sich heraus: Wenn es ein wirklich gutes Reiseunternehmen gibt, kann es Ihnen eine Reise zu fantastischen Zielen verkaufen und dafür sorgen, dass Sie und Ihre Freunde und Familie viel Spaß haben. Dann ist ein gutes

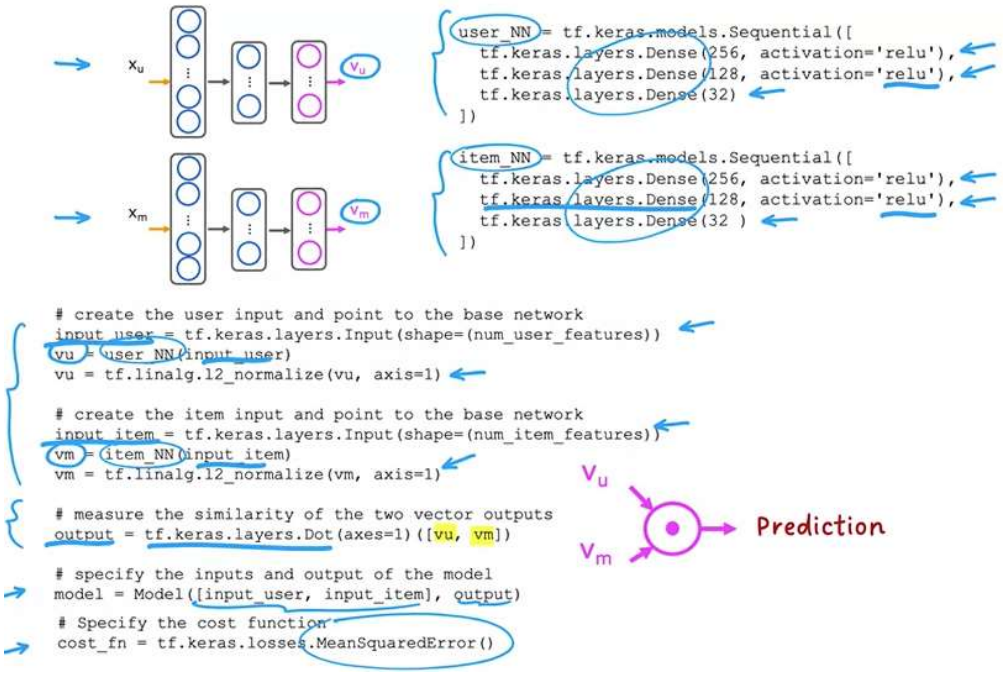
Reisegeschäft meiner Meinung nach oft profitabler. Das andere Geschäft ist profitabler. Sie können dann höhere Gebote für Anzeigen abgeben. Es kann es sich leisten, mehr zu zahlen, um Benutzer zu gewinnen. Da es sich eine Online-Werbeseite leisten kann, höhere Gebote für Anzeigen abzugeben, wird sie ihre Anzeigen häufiger schalten und mehr Nutzer zu diesem guten Unternehmen führen. Dies ist ein positiver Kreislauf: Je mehr Nutzer Sie gut bedienen, desto profitabler ist das Geschäft, desto mehr können Sie für Anzeigen bieten, desto mehr Traffic erhalten Sie und so weiter. Nur ein positiver Kreislauf wird vielleicht sogar dazu neigen, zu hoffen, dass die guten Reiseunternehmen statistisch gesehen noch besser abschneiden. Schauen wir uns das problematische Beispiel an. Die Zahltagdarlehensbranche verlangt in der Regel extrem hohe Zinssätze, oft von Personen mit niedrigem Einkommen. Eine Möglichkeit, im Geschäft mit Kurzzeitkrediten erfolgreich zu sein, besteht darin, wirklich effizient zu sein und jeden einzelnen Dollar, den man aus ihnen herausholen kann, von den Kunden abzuquetschen. Wenn es ein Unternehmen für Kurzzeitkredite gibt, das sehr gut darin ist, Kunden auszubeuten und Kunden wirklich für jeden einzelnen Dollar zu erpressen, dann wird dieses Unternehmen profitabler sein. Daher können sie für Anzeigen höher sein. Da sie höhere Gebote für Anzeigen erhalten können, wird ihnen mehr Traffic zugesandt. Dies ermöglicht es ihnen, noch mehr Kunden zu gewinnen und noch mehr Leute zu gewinnen, um Gewinn zu erzielen. Dies wiederum erhöht auch eine positive Rückkopplungsschleife. Außerdem kann es zu einer positiven Rückkopplungsschleife kommen, die dazu führen kann, dass die ausbeuterischsten und schädlichsten Zahltagdarlehensunternehmen mehr Traffic erhalten. Dies scheint der gegenteilige Effekt zu sein als das, was wir für gut für die Gesellschaft halten. Ich weiß nicht, dass es dafür eine einfache Lösung gibt. Dies sind sehr schwierige Probleme, die die Systemphase empfehlen. Eine Verbesserung könnte darin bestehen, die Schaltung von Anzeigen ausbeuterischer Unternehmen zu verweigern. Das lässt sich natürlich leicht sagen. Aber wie definiert man, was ein ausbeuterisches Geschäft ist und was nicht, ist eine sehr schwierige Frage. Aber wenn wir Empfehlungssysteme für Werbung oder andere Dinge entwickeln, denke ich, dass dies Fragen sind, die sich jeder von uns, der an diesen Technologien arbeitet, stellen sollte, damit wir hoffentlich zu offenen Diskussionen und Debatten einladen, mehrere Meinungen von mehreren Leuten einholen und es versuchen können Wir müssen Designentscheidungen treffen, die es unseren Systemen ermöglichen, viel mehr zu nützen als potenziell zu schaden. Schauen wir uns einige andere Beispiele an. In den Nachrichten wurde häufig darüber berichtet, dass die Maximierung des Benutzerengagements, beispielsweise der Zeit, die jemand Videos auf einer Website ansieht, oder der Zeit, die jemand in sozialen Medien verbringt. Dies hat dazu geführt, dass große Social-Media- und Video-Sharing-Seiten Verschwörungstheorien oder Hass und Toxizität verbreiten, weil Verschwörungstheorien und bestimmte Arten hasserfüllter Inhalte sehr ansprechend sind und dazu führen, dass Menschen viel Zeit damit verbringen. Selbst wenn die Wirkung der Verstärkung von Verschwörungstheorien die versteckte Toxizität verstärkt, erweist sich dies als schädlich für den Einzelnen und die Gesellschaft insgesamt. Eine Verbesserung für dieses unvollständige und unvollkommene Verhalten besteht darin, problematische Inhalte wie Hassreden, Betrug, Betrügereien und möglicherweise bestimmte Arten von gewalttätigen Inhalten herauszufiltern. Auch hier ist es überraschend schwierig, Definitionen dafür zu entwickeln, was genau wir herausfiltern sollten. Das bringt eine Reihe von Problemen mit sich, mit denen sich meiner Meinung nach Unternehmen, Einzelpersonen und sogar Regierungen weiterhin auseinandersetzen müssen. Nur ein letztes Beispiel.



Wenn ein Benutzer viele absolute Websites besucht, denken die Benutzer meiner Meinung nach, dass die Apple-Website, die ich dem Benutzer empfehlen wollte, ihnen gefallen wird. Ich denke, dass vielen Benutzern nicht bewusst ist, dass viele Apps und Websites versuchen, ihren Gewinn zu maximieren und nicht unbedingt die Freude des Benutzers an den empfohlenen Medienelementen. Ich möchte Sie und andere Unternehmen ermutigen, wenn möglich, den Benutzern gegenüber transparent zu machen, anhand welcher Kriterien Sie entscheiden, was Sie ihnen empfehlen möchten. Ich weiß, dass das nicht immer einfach ist, aber letztendlich hoffe ich, dass eine größere Transparenz gegenüber den Benutzern darüber, warum wir sie ihnen zeigen und warum, das Vertrauen stärkt und auch dazu führt, dass unsere Systeme der Gesellschaft mehr Gutes tun. Empfehlungssysteme sind eine sehr leistungsstarke Technologie, eine sehr profitable und sehr lukrative Technologie. Es gibt auch einige problematische Anwendungsfälle. Wenn Sie eines dieser Systeme mit Empfehlungstechnologie oder tatsächlich einer anderen maschinellen Lern- oder anderen Technologie erstellen. Ich hoffe, Sie denken nicht nur über die Vorteile nach, die Sie schaffen können, sondern auch über den möglichen Schaden und laden unterschiedliche Perspektiven ein, um zu diskutieren und zu debattieren. Bitte bauen und tun Sie nur Dinge, von denen Sie wirklich glauben, dass es der Gesellschaft besser gehen kann. Ich hoffe, dass wir alle gemeinsam nur Arbeit leisten können, die den Menschen besser geht. Danke fürs Zuhören. Wir haben noch ein weiteres Video zum Thema Empfehlungssysteme, in dem wir uns einige praktische Tipps zur Implementierung eines inhaltsbasierten Filteralgorithmus in TensorFlow ansehen. Fahren wir mit dem letzten Video über Empfehlungssysteme fort.

TensorFlow- Implementierung der inhaltsbasierten Filterung

Im Übungslabor erfahren Sie, wie Sie inhaltsbasierte Filterung in TensorFlow implementieren. In diesem Video möchte ich Ihnen lediglich einige der Schlüsselkonzepte im Code vorstellen, mit denen Sie experimentieren können. Lass uns einen Blick darauf werfen.

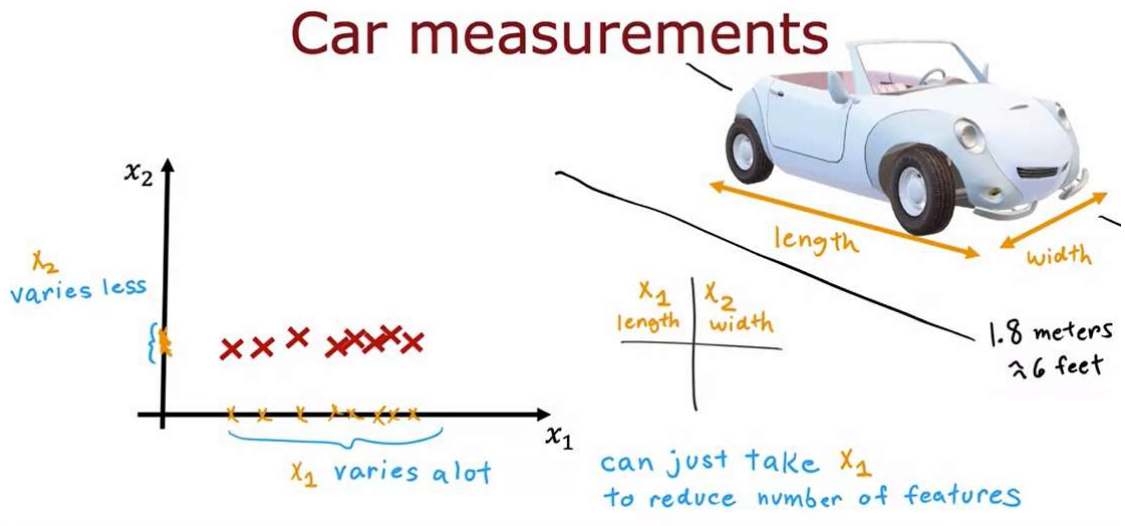


Denken Sie daran, dass unser Code mit einem Benutzernetzwerk und einem funktionierenden Film begonnen hat. Die Art und Weise, wie Sie dies in TensorFlow implementieren können, ist der Art und Weise sehr ähnlich, wie wir zuvor ein neuronales Netzwerk mit einer Reihe dichter Schichten implementiert haben. Wir werden ein sequentielles Modell verwenden. In diesem Beispiel haben wir dann zwei dichte Schichten mit der hier angegebenen Anzahl versteckter Einheiten, und die letzte Schicht hat 32 Einheiten und gibt 32 Zahlen aus. Dann nenne ich das Filmnetzwerk das Artikelnetzwerk, da die Filme hier die Artikel sind. So sieht der Code aus. Wieder einmal haben wir dichte verborgene Schichten gekoppelt, gefolgt von dieser Schicht, die 32 Zahlen ausgibt. Für die ausgeblendeten Ebenen verwenden wir unsere standardmäßige Aktivierungsfunktion, nämlich die Relu- Aktivierungsfunktion. Als nächstes müssen wir TensorFlow mitteilen Keras, wie die Benutzerfunktionen oder die Artikelfunktionen, also die Filmfunktionen, den beiden neuronalen Netzen zugeführt werden. Dies ist die Syntax dafür. Das extrahiert die Eingabemerkmale für den Benutzer und gibt sie dann an den Benutzer weiter. Das haben wir hier oben definiert, um vu, den Vektor für den Benutzer, zu berechnen. Ein zusätzlicher Schritt, der diesen Algorithmus etwas besser funktionieren lässt, ist diese Zeile hier, die den Vektor vu so normalisiert, dass er die Länge eins hat. Dies normalisiert die Länge, auch l2-Norm genannt, aber im Grunde ist die Länge des Vektors vu gleich eins. Dann machen wir dasselbe für das Artikelnetzwerk, für das Filmnetzwerk. Dadurch werden die Artikelmerkmale extrahiert und in das neuronale Netzwerk des Artikels eingespeist, das wir dort oben definiert haben. Dadurch wird der Filmvektor vm berechnet. Schließlich normalisiert der Schritt diesen Vektor auch so, dass er die Länge eins hat. Nachdem wir vu und vm berechnet haben, müssen wir das Skalarprodukt zwischen diesen beiden Vektoren bilden. Dies ist die Syntax dafür. Keras hat einen speziellen Ebenentyp, beachten Sie, dass wir hier tf hatten Keras -Schichten dicht, hier ist das tf Keras- Schichten punktieren. Es stellt sich heraus, dass es eine spezielle Keras-Schicht gibt, die einfach ein Skalarprodukt zwischen zwei Zahlen bildet. Damit werden wir das Skalarprodukt zwischen den Vektoren vu und vm ermitteln. Dies ergibt die Ausgabe des neuronalen Netzwerks. Dies ergibt die endgültige Vorhersage. Um Keras abschließend mitzuteilen, welche Eingaben und Ausgaben das Modell hat, teilt ihm diese Zeile mit, dass das Gesamtmodell ein Modell ist, dessen Eingaben die Benutzerfunktionen und der Film oder die Elementfunktionen und die Ausgabe sind. Dies ist die Ausgabe, die wir gerade definiert haben hoch oben. Die Kostenfunktion, die wir zum Trainieren dieses Modells verwenden, wird die mittlere quadratische Fehlerkostenfunktion sein. Dies sind die wichtigsten Codeausschnitte für die Implementierung der

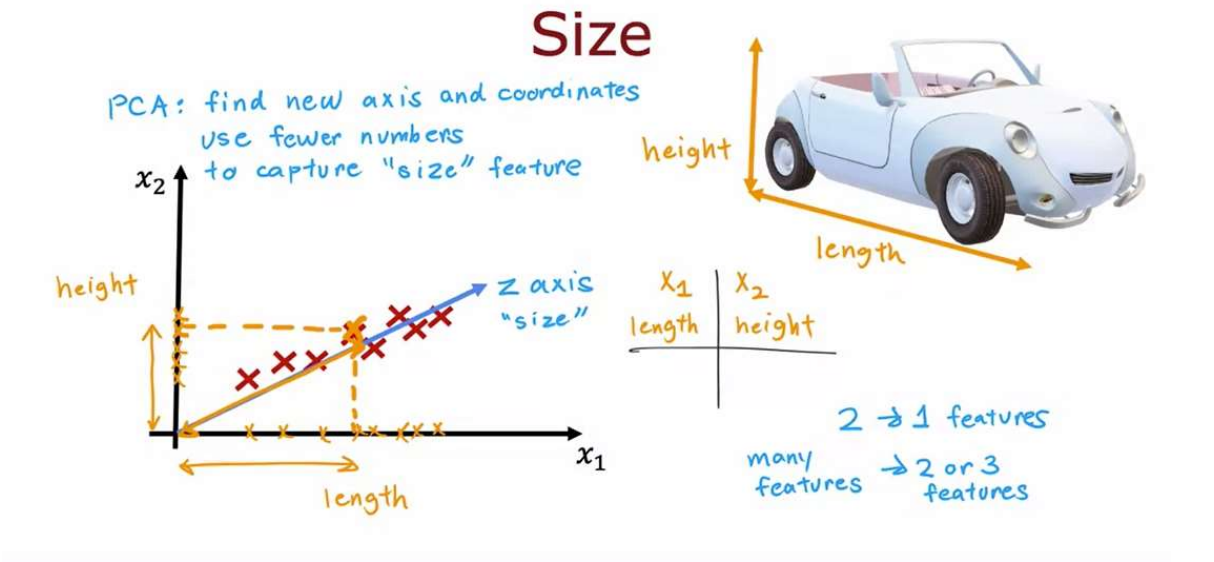
inhaltsbasierten Filterung als neuronales Netzwerk. Den Rest des Codes sehen Sie im Übungslabor, aber hoffentlich können Sie damit spielen und sehen, wie alle diese Codeschnipsel in eine funktionierende TensorFlow- Implementierung eines inhaltsbasierten Filteralgorithmus passen. Es stellt sich heraus, dass es noch einen weiteren Schritt gibt, über den ich vorher nicht gesprochen habe, aber wenn Sie diesen tun, nämlich die Länge des Vektor-vu zu normalisieren, funktioniert der Algorithmus etwas besser. TensorFlows verfügt über diese l2-normalisierte Bewegung, die den Vektor normalisiert. Sie wird auch als Normalisierung der l2-Norm des Vektors bezeichnet, daher der Name der Funktion. Das ist es. Vielen Dank, dass Sie mich durch all dieses Material über Empfehlungssysteme begleitet haben. Es handelt sich um eine spannende Technologie. Ich wünsche Ihnen viel Spaß beim Experimentieren mit diesen Ideen und Codes in den Übungslaboren dieser Woche. Das bringt uns zu den vielen dieser Videos über Empfehlungssysteme und zum Ende der nächsten bis letzten Woche für diese Spezialisierung. Ich freue mich auch darauf, Sie nächste Woche wiederzusehen. Wir sprechen über die spannende Technologie des Reinforcement Learning. Ich wünsche Ihnen viel Spaß bei den Tests und den Übungseinheiten und freue mich darauf, Sie nächste Woche wiederzusehen.

Reduzierung der Anzahl der Features (optional)

Ich hoffe, Ihnen haben die Videos gefallen, die Ihnen zeigen, wie Sie Ihr eigenes Empfehlungssystem aufbauen können. Bevor wir diese Woche mit diesem und einigen anderen optionalen Videos abschließen, möchte ich Ihnen einen unbeaufsichtigten Lernalgorithmus namens Hauptkomponentenanalyse vorstellen. Dies ist ein Algorithmus, der häufig zur Visualisierung verwendet wird. Insbesondere wenn Sie über einen Datensatz mit vielen Features verfügen, beispielsweise 10 Features oder 50 Features oder sogar Tausenden von Features, können Sie keine 1.000-dimensionalen Daten darstellen. PCA oder Hauptkomponentenanalyse ist ein Algorithmus, mit dem Sie Daten mit vielen Features, 50, 1.000 oder sogar mehr, aufnehmen und die Anzahl der Features auf zwei Features, vielleicht drei Features, reduzieren können, damit Sie sie grafisch darstellen und visualisieren können. Wird häufig von Datenwissenschaftlern verwendet, um die Daten zu visualisieren und herauszufinden, was möglicherweise vor sich geht. Werfen wir einen Blick darauf, wie PCA, die Hauptkomponentenanalyse, funktioniert. Um PCA zu beschreiben, werde ich als laufendes Beispiel verwenden, wenn Sie Daten aus einer Sammlung von Personenkraftwagen haben und Personenkraftwagen viele Funktionen haben können. Möglicherweise kennen Sie die Länge des Autos oder die Breite des Autos, vielleicht den Durchmesser des Rades oder vielleicht die Höhe des Autos und viele andere Merkmale von Autos.

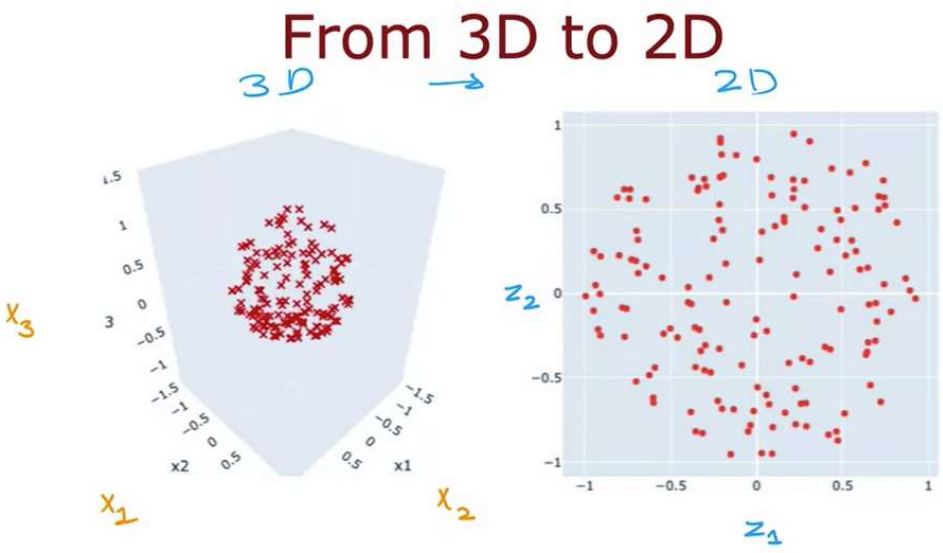


Wenn Sie die Anzahl der Funktionen reduzieren möchten, damit Sie sie visualisieren können, wie können Sie PCA dazu verwenden? Nehmen wir für das erste Beispiel an, Sie erhalten einen Datensatz mit zwei Funktionen. Das Merkmal x_1 ist auf diese Weise die Länge des Autos und das zweite Merkmal x_2 ist die Breite des Autos, die auf diese Weise gemessen wird. Es stellt sich heraus, dass in den meisten Ländern die Breite des Autos, die in die Breite der Straße einer einzelnen Spur passen muss, aufgrund von Einschränkungen hinsichtlich der Breite der Straße, auf der die Autos fahren, tendenziell nicht so stark variiert. In den Vereinigten Zustanden beispielsweise sind die meisten Autos, sagen wir mal, etwa 1,8 Meter breit, also knapp 1,80 Meter. Wenn Sie eine Sammlung von Autos und den Datensatz über die Länge und Breite der Autos hätten, würden Sie feststellen, dass der Datensatz so aussehen könnte, wobei x_1 ziemlich stark variiert, weil einige Autos wirklich lang sind, und x_2 relativ wenig variiert. Wenn Sie die Anzahl der Funktionen reduzieren möchten, können Sie uns beispielsweise x_1 überlassen, da x_2 von Auto zu Auto relativ wenig variiert. Es stellt sich heraus, dass PCA ein Algorithmus ist, der bei Anwendung auf diesen Datensatz mehr oder weniger automatisch entscheidet, nur x_1 zu verwenden, aber er kann noch viel mehr. Schauen wir uns ein zweites Beispiel an, bei dem x_1 wieder die Länge des Autos ist und wir sagen, dass in diesem Datensatz x_2 der Durchmesser des Rads ist. Der Durchmesser des Rades variiert ein wenig. Wenn Sie die Daten grafisch darstellen würden, könnte das so aussehen.



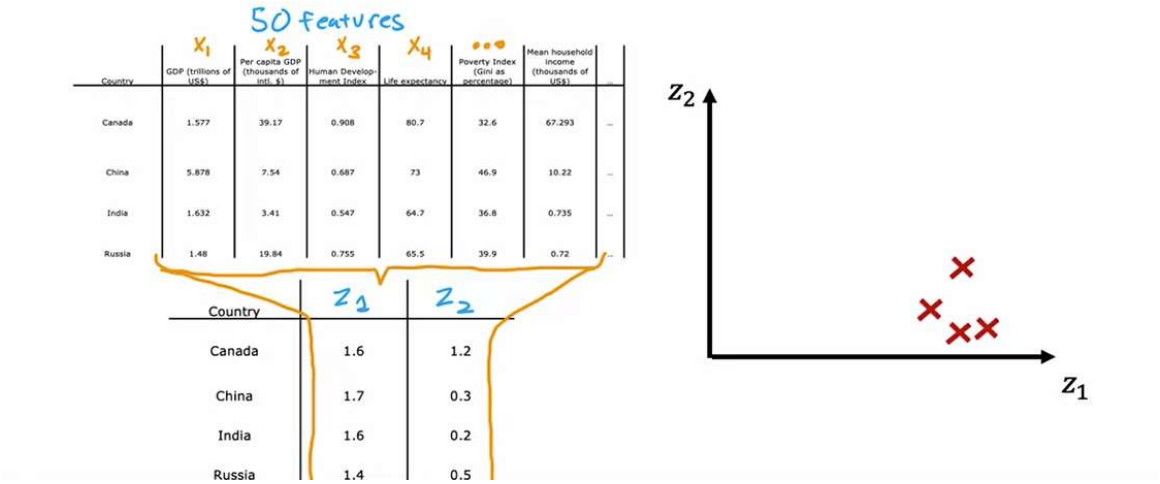
Aber auch hier: Wenn Sie diesen Datensatz auf nur eine Funktion vereinfachen möchten, könnten Sie sich entscheiden, einfach x_1 zu nehmen und x_2 und PCA bei der Anwendung auf diesen Datensatz zu vergessen. Nun, mehr oder weniger, veranlassen Sie einfach, die Funktion x_1 zu überprüfen. In beiden Beispielen, die wir gesehen haben, schien nur eines der beiden Merkmale einen sinnvollen Variationsgrad aufzuweisen. Hier ist ein komplexeres Beispiel. Angenommen, das Merkmal x_1 ist die Länge des Autos, das variiert also ziemlich stark, und das Merkmal x_2 ist hier die Höhe des Autos, die ebenfalls ziemlich stark variiert. Manche Autos sind viel höher als andere Autos. Wenn Sie die Daten grafisch darstellen, erhalten Sie möglicherweise einen Datensatz, der so aussieht: Einige Autos sind größer und tendenziell länger und höher, während andere etwas kleiner sind. Sie neigen dazu, nicht so lang und nicht so hoch zu sein. Was sollten Sie wählen, wenn Sie die Anzahl der Funktionen reduzieren möchten? Sie möchten nicht nur x_1, die Länge, auswählen und x_2, die Höhe, ignorieren, und Sie möchten auch nicht nur x_2, die Höhe, auswählen und x_1, die Länge, ignorieren. Es scheint, als hätten sowohl x_1 als auch x_2 nützliche Informationen. In diesem Diagramm sind x_1 und x_2 die beiden Achsen dieses Diagramms. Anstatt sich darauf zu beschränken, entweder die x_1-Achse oder die x_2-Achse zu verwenden, was wäre, wenn wir eine dritte Achse hätten? Ich werde

diese neue Achse Z-Achse nennen. Um es klar zu sagen: Dies steht in diesem Diagramm nicht im Vordergrund. Dies ist eine Kombination aus x_1 und x_2. Dies ist keine Z-Achse, die in der dritten Dimension hervorsteht. Diese z-Achse liegt flach innerhalb dieses Diagramms. Aber warum haben wir die Z-Achse, die etwa der Größe des Autos entspricht? Wenn wir ein Auto wie dieses hier haben, sagen uns seine Koordinaten, d. h. der Wert auf der In ähnlicher Weise sagt uns seine Koordinate, das heißt, wie groß dieser Abstand auf der x_2-Achse ist, wie hoch das Auto ist. Wenn wir jetzt stattdessen die Z-Achse als ein Merkmal verwenden, um zu erfassen, was wir über dieses Auto wissen, dann ist die Koordinate auf der Z-Achse, also dieser Abstand. Das sagt uns ungefähr, wie groß das Auto ist. Wir werden dies in den nächsten Videos formalisieren. Die Idee von PCA besteht jedoch darin, eine oder mehrere neue Achsen zu finden, z. B. die Z-Achse, sodass Sie beim Messen der Koordinaten Ihrer Daten auf der neuen Achse immer noch sehr nützliche Informationen über das Auto erhalten. Aber vielleicht brauchen wir jetzt nicht mehr zwei Zahlen, die den Koordinaten auf den Achsen X_1 und X_2 entsprechen, sondern die Länge und die Höhe. Sie benötigen nun ein paar Zahlen, in diesem Fall nur eine statt zwei, um ungefähr die Größe des Autos zu erfassen. In dem Beispiel, das wir bisher verwendet haben, haben wir versucht, die Daten von zwei Zahlen, X_1 und X_2, auf eine Zahl, die Koordinate auf der Z-Achse, zu reduzieren. In der Praxis wird PCA normalerweise verwendet, um eine sehr große Anzahl von Merkmalen, beispielsweise 10, 20, 50 oder sogar Tausende von Merkmalen, auf vielleicht zwei oder drei Merkmale zu reduzieren, sodass Sie die Daten zweidimensional oder in einem Bild visualisieren können dreidimensionale Handlung. Aber für dieses Video, denn ich konnte nur auf einem zweidimensionalen Bildschirm zeichnen.

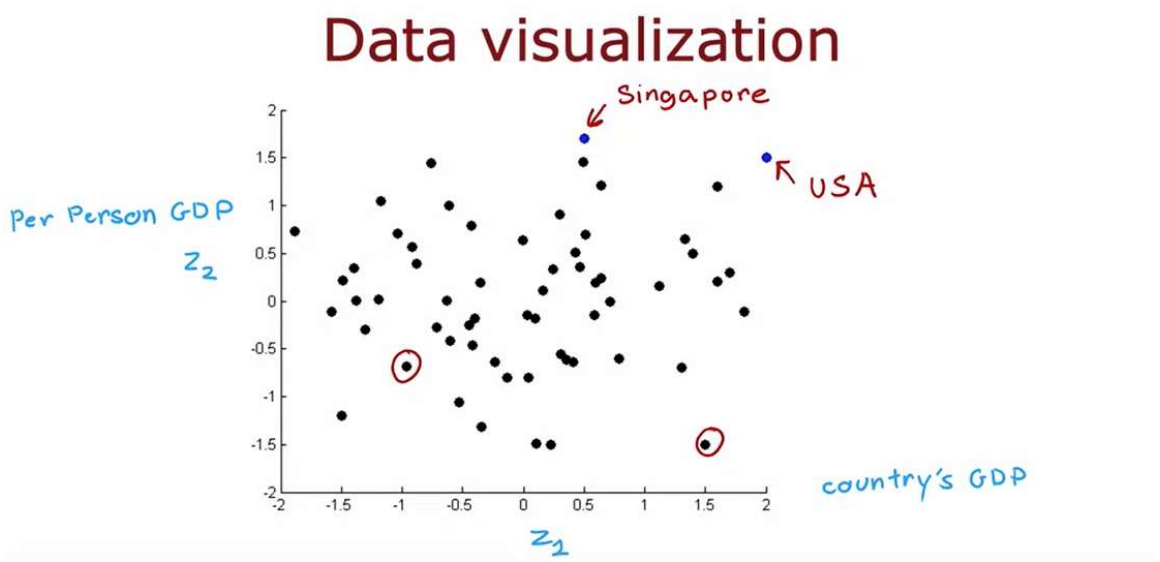


Als Beispiele verwende ich hauptsächlich zwei- oder dreidimensionale Datensätze. Schauen wir uns noch ein Beispiel an. In dieser Visualisierung haben wir einen dreidimensionalen Datensatz und beachten, dass ich den Datensatz hier drehen kann, sodass Sie ihn in 3D sehen können. Aber beachten Sie, wenn ich den Datensatz auf diese Weise drehe, werden die meisten dieser Daten, obwohl sie in 3D vorliegen, tatsächlich auf einer sehr dünnen Oberfläche gespeichert. Es ist fast so, als ob alle Daten auf einem zweidimensionalen Pfannkuchen liegen, obwohl der Pfannkuchen in diesem dreidimensionalen Raum lebt. Eine PCA können Sie statt der drei Merkmale x_1, x_2, x_3 auf zwei Zahlen reduzieren, die wir Z_1 und Z_2 nennen. Wenn Sie das tun, können Sie die Daten auf dieser Z_1-, Z_2-Achse visualisieren und dies wird zu einer bequemen Möglichkeit, diese Daten zu visualisieren, wenn Sie sie beispielsweise auf ein Blatt Papier drucken müssten. Ich konnte es nicht dynamisch drehen, wie Sie es auf dem Bildschirm sehen werden. Hier ist noch ein Beispiel. Wenn Sie Daten über den Entwicklungsstatus vieler verschiedener Länder haben, könnten Sie beispielsweise Daten über verschiedene Länder, das BIP und das Merkmal x_1 haben. Nehmen wir außerdem an,

wir haben auch das Pro-Kopf-BIP und ein Maß für ihren Human Development Index. Der Human Development Index wurde entwickelt, um den allgemeinen Fortschritt zu messen, wie gut es den Menschen in einem Land geht, basierend auf Dingen wie Lebenserwartung, Bildung usw., oder Sie können separat ein Merkmal haben, das der Lebenserwartung in verschiedenen Ländern usw. entspricht und so weiter. Wenn Sie für jedes Land 50 Merkmale haben, wie können Sie diese Daten dann visualisieren, da Sie keine 50-dimensionalen Daten auf einem zweidimensionalen Computermonitor darstellen können ?



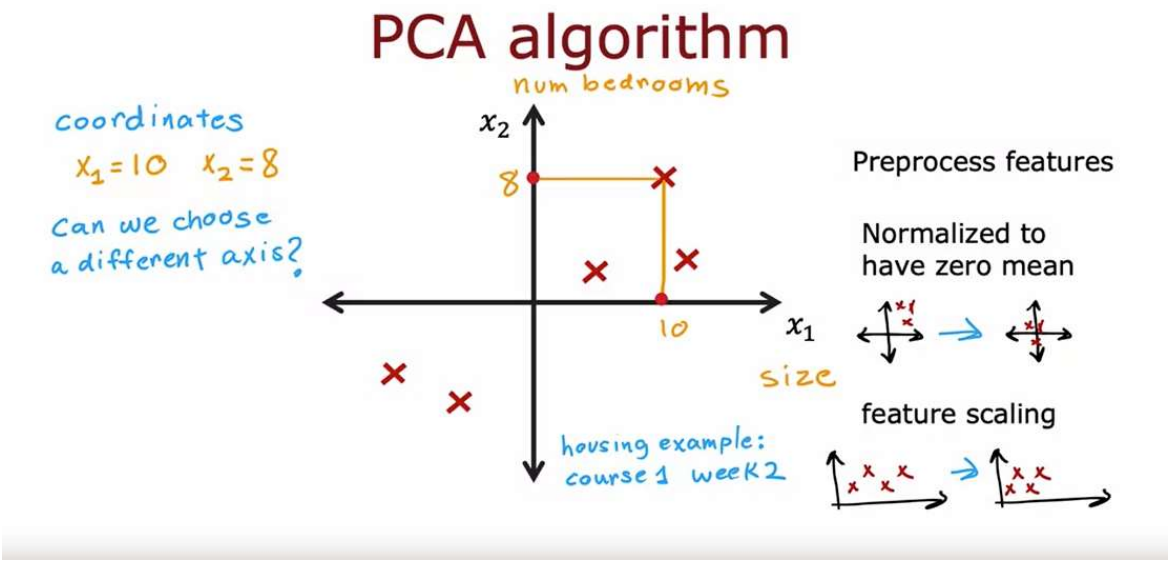
Was PCLS tun wird, ist, diese 50 Features, X_1, und z_2. Sie könnten beispielsweise feststellen, dass Z_1 grob der Größe des Landes und dem Gesamt-BIP entspricht. Weil größere Länder tendenziell ein höheres BIP haben, weil große Länder viele Menschen haben und tendenziell eine größere Wirtschaft haben, und vielleicht stellen Sie fest, dass Z_2 ungefähr dem BIP pro Person oder dem Umfang der Wirtschaftstätigkeit pro Person entspricht. Zum Beispiel könnten die Vereinigten Zustanden, die ein relativ großes Land sind und eine relativ hohe Wirtschaftsaktivität pro Person aufweisen, irgendwo oben und rechts in diesem Diagramm liegen, und ein Land wie Singapur, in dem ich ebenfalls viele Jahre lebe, ist eins Es handelt sich um ein kleineres Land mit einer relativ hohen Wirtschaftsaktivität pro Kopf. Nehmen Sie auf der z_1-Achse einen niedrigeren Wert an, auf der z_2-Achse jedoch immer noch einen relativ hohen Wert.



Während ein Land wie dieses vielleicht ein kleineres Land mit einer geringeren Wirtschaftsaktivität pro Person wäre. Während es sich bei einem solchen Land möglicherweise um ein großes Land mit einer geringeren Wirtschaftsaktivität pro Person handelt, können Sie bei einer solchen Zahl eine große Anzahl von Merkmalen berücksichtigen, nämlich 50 Merkmale. Manchmal sagen wir auch, dass es sich um 50-dimensionale Daten handelt. Es bedeutet einfach, dass wir 50 Features haben und diese auf zwei Features reduzieren, oder manchmal sagen wir, dass es sich um zweidimensionale Daten handelt, weil Sie sie dann in diesem zweidimensionalen Diagramm darstellen können, wie Sie es hier sehen. Immer wenn ich einen neuen Datensatz erhalte, möchte ich die Daten oft visualisieren, denn das hilft mir zu verstehen, wie die Daten aussehen, wie die Länder aussehen oder wie die Autos aussehen Dieser Datensatz oder welche Daten auch immer Sie untersuchen. Sie werden auch feststellen, dass die Visualisierung des Datensatzes Ihnen manchmal dabei hilft, herauszufinden, was in diesen Daten vor sich geht. Es passiert etwas Unerwartetes. PCA ist ein leistungsstarker Algorithmus, um Daten mit vielen Merkmalen, mit vielen Dimensionen oder hochdimensionalen Daten zu erfassen und sie auf zwei oder drei Merkmale zu zwei- oder dreidimensionalen Daten zu reduzieren, damit Sie sie grafisch darstellen, visualisieren und besser verstehen können Was ist in Ihren Daten? Das kann der PCA-Algorithmus für Sie tun. Im nächsten Video beginnen wir damit, einen Blick darauf zu werfen, wie genau der PCA-Algorithmus funktioniert.

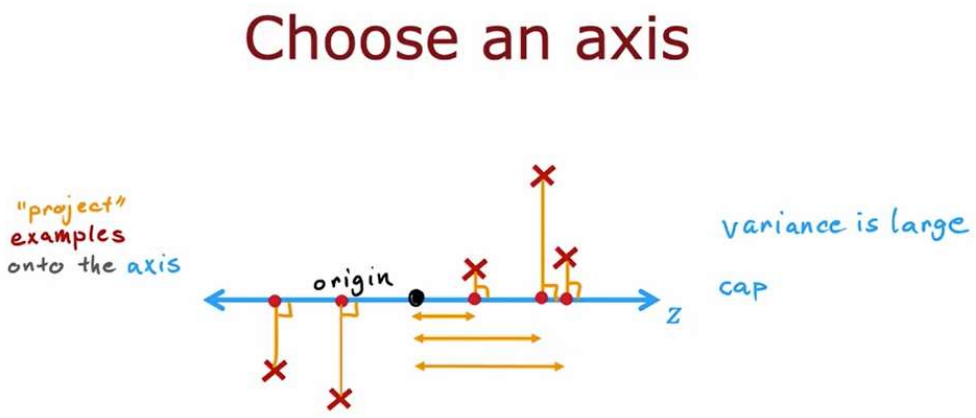
PCA-Algorithmus (optional)

Wie funktioniert PCA? Wenn Sie einen Datensatz mit zwei Features haben, x_1 und x_2. Zunächst werden Ihre Daten mithilfe der Achsen x_1 und x_2 aufgezeichnet oder dargestellt. Sie möchten diese beiden Funktionen jedoch durch nur eine Funktion ersetzen. Wie können Sie eine neue Achse auswählen, nennen wir sie die Z-Achse, die irgendwie eine gute Funktion zum Erfassen und Darstellen der Daten ist? Schauen wir uns an, wie PCA dies tut. Hier sind die Datensätze mit fünf Trainingsbeispielen. Denken Sie daran, dass dies ein unbeaufsichtigter Lernalgorithmus ist, wir haben also nur x_1 und x_2, es gibt keine Bezeichnung y. Ein Beispiel wie dieses könnte die Koordinaten x_1 gleich 10 und x_2 gleich 8 haben.

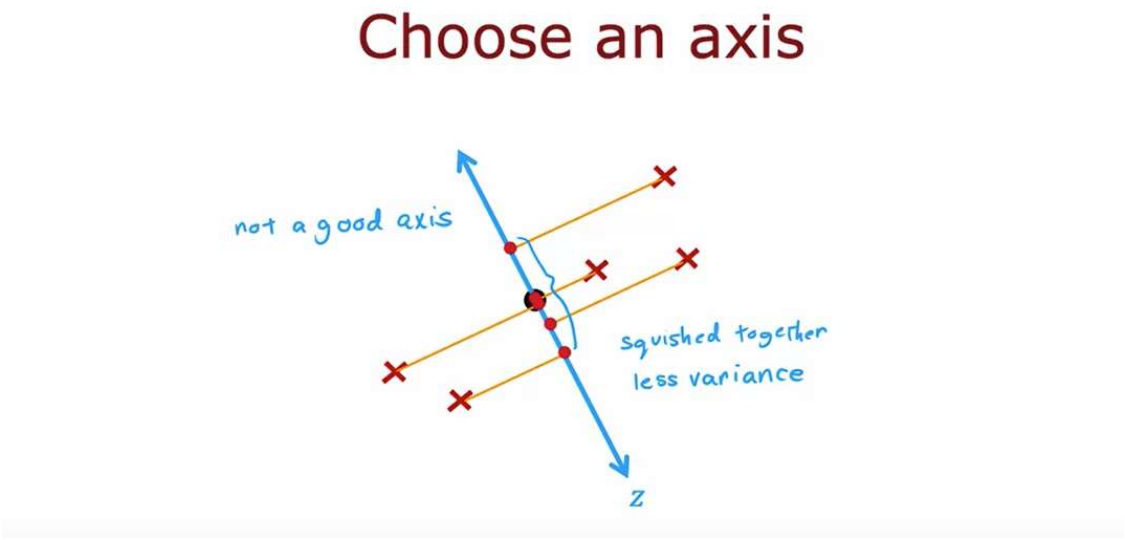


Wenn wir die Achsen x_1 und x_2 nicht verwenden möchten, wie können wir dann eine andere Achse auswählen, mit der wir erfassen, was in den Daten enthalten ist, oder mit der wir darstellen möchten? die Daten? Ein Hinweis zur Vorverarbeitung: Bevor die nächsten PCA-Schritte angewendet werden, sollten die Funktionen zunächst so normalisiert werden, dass sie einen Mittelwert von Null haben, und das habe ich hier bereits getan. Wenn die Merkmale x_1 und x_2 sehr unterschiedliche

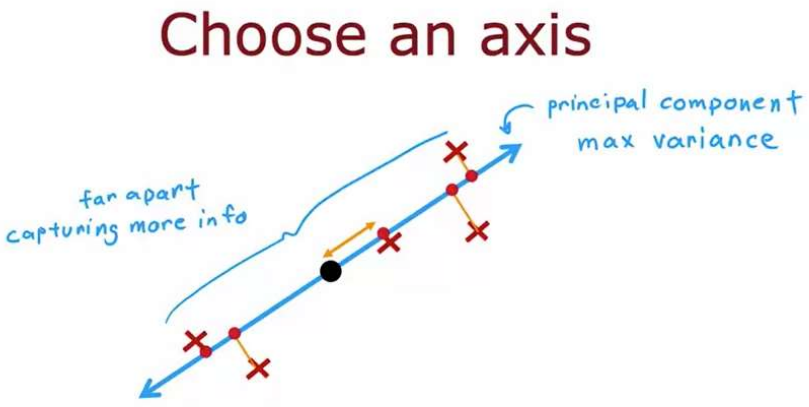
Maßstäbe annehmen, wenn Sie sich beispielsweise an unser Wohnbeispiel erinnern: Wenn x_1 die Größe eines Hauses in Quadratfuß und x_2 die Anzahl der Schlafzimmer wäre, dann könnte x_1 1.000 oder ein paar sein Tausend, während x_2 eine kleine Zahl ist. Wenn die Features sehr unterschiedliche Maßstäbe annehmen, führen Sie zunächst eine Feature-Skalierung durch, bevor Sie die nächsten PCA-Schritte anwenden. Angenommen, die Features wurden so normalisiert, dass sie einen Mittelwert von Null haben. Subtrahieren Sie also den Mittelwert von jedem Feature und wenden Sie dann möglicherweise auch die Feature-Skalierung an, damit die Bereiche nicht zu weit voneinander entfernt sind. Was macht PCA als nächstes? Um zu untersuchen, was PCA macht, möchte ich die Achsen x_1 und x_2 entfernen, sodass nur noch die fünf Trainingsbeispiele übrig bleiben. Dieser Punkt stellt hier den Ursprung dar. Die Position Null in diesem Diagramm ist immer noch. Was wir jetzt mit PCA tun müssen, ist, eine Achse anstelle der zwei Achsen auszuwählen, die wir zuvor hatten, um zu erfassen, was an diesen fünf Beispielen wichtig ist. Wenn wir diese Achse als unsere neue Z-Achse wählen würden, wäre sie tatsächlich dieselbe wie die x_1-Achse, nur für dieses Beispiel. Was wir dann sagen, ist, dass wir für dieses Beispiel einfach diesen Wert, diese Koordinate auf der Z-Achse, erfassen werden. Im zweiten Beispiel werden wir diesen Wert erfassen, und dann wird dieser Wert erfasst, und so weiter für alle fünf Beispiele. Anders ausgedrückt:

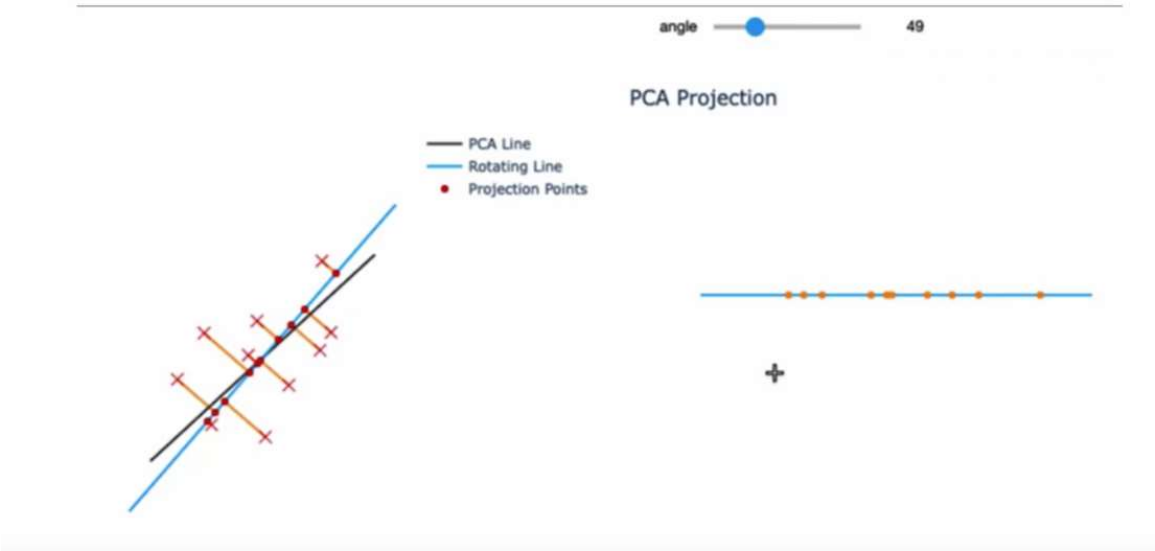


Wir nehmen jedes dieser Beispiele und projizieren es auf einen Punkt auf der Z-Achse. Das Wort Projekt bezieht sich darauf, dass Sie dieses Beispiel nehmen und es mithilfe dieses Liniensegments, das in einem 90-Grad-Winkel zur Z-Achse steht, auf die Z-Achse übertragen. Dieses kleine Kästchen hier wird verwendet, um anzuzeigen, dass dieses Liniensegment im 90-Grad-Winkel zur Z-Achse verläuft. Der Begriff „Projekt“ bedeutet lediglich, dass Sie einen Punkt nehmen und diesen entsprechenden Punkt auf der Z-Achse mithilfe dieses Liniensegments im 90-Grad-Winkel finden. Diese Richtung als Z-Achse auszuwählen ist keine schlechte Wahl, aber es gibt noch bessere Möglichkeiten. Diese Wahl ist nicht schlecht, denn wenn Sie Ihre Beispiele auf die Z-Achse projizieren, erfassen Sie immer noch einen Großteil der Datenverteilung. Diese fünf Punkte hier sind ziemlich weit auseinander, sodass Sie immer noch einen Großteil der Variation bzw. Varianz im Originaldatensatz erfassen. Damit meine ich, dass diese fünf Punkte ziemlich weit voneinander entfernt sind und daher die Varianz oder Variation zwischen diesen fünf Punkten, den Projektionen der Daten auf die Z-Achse, recht groß ist. Das bedeutet, dass wir immer noch einen Großteil der Informationen der ursprünglichen fünf Beispiele erfassen. Schauen wir uns einige andere mögliche Optionen für die z-Achse an. Hier ist eine andere Wahl.

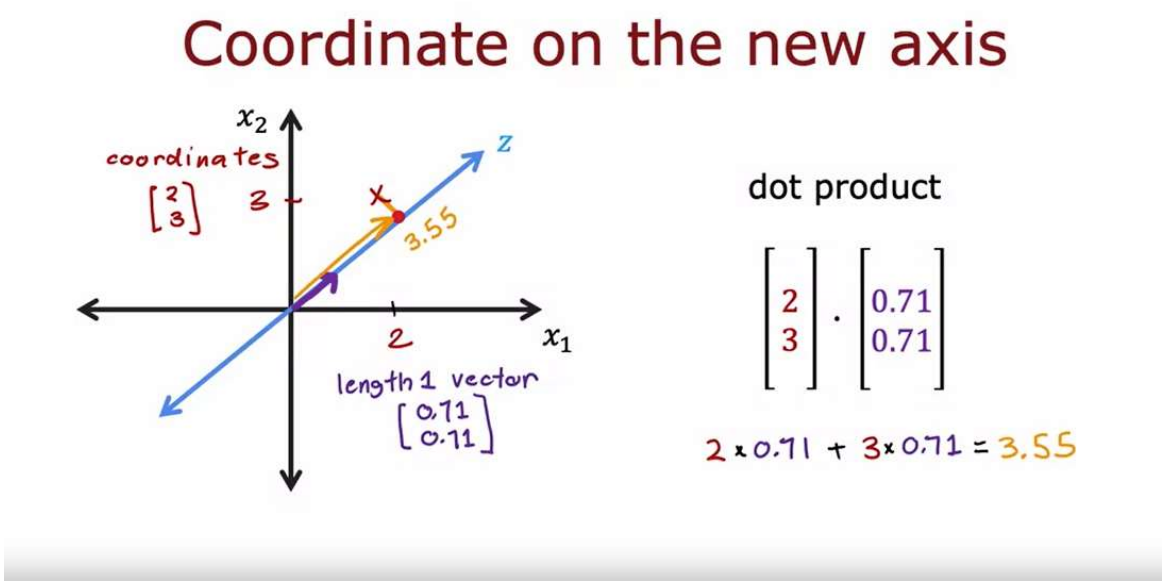


Das ist eigentlich keine gute Wahl. Aber wenn ich dies als meine Z-Achse wählen würde, wenn ich dieselben fünf Beispiele nehme und sie auf die Z-Achse projiziere, erhalte ich am Ende diese fünf Punkte. Sie bemerken, dass diese fünf Punkte im Vergleich zur vorherigen Auswahl ziemlich zusammengequetscht sind. Der Betrag, um den sie sich voneinander unterscheiden, oder ihre Varianz oder Variation ist viel geringer. Das bedeutet, dass Sie mit dieser Wahl von z viel weniger Informationen im Originaldatensatz erfassen, da Sie alle fünf Beispiele teilweise zusammengequetscht haben. Sehen wir uns noch eine letzte Möglichkeit an: Wenn ich diese Achse als Z-Achse wähle. Dies ist tatsächlich eine bessere Wahl als die beiden vorherigen, die wir gesehen haben, denn wenn wir die Projektionen der Daten auf die Z-Achse nehmen, stellen wir fest, dass diese Punkte hier tatsächlich ziemlich weit voneinander entfernt sind. Wir erfassen viele Variationen und viele Informationen im ursprünglichen Datensatz, auch wenn wir jetzt nur noch eine Koordinate oder eine Zahl verwenden, um jedes der Trainingsbeispiele darzustellen oder zu erfassen, anstatt zwei Zahlen oder zwei Koordinaten, X_1 und X_2. Im PCA-Algorithmus wird diese Achse als Hauptkomponente bezeichnet. Wenn Sie die Daten auf die Z-Achse projizieren, erhalten Sie am Ende die größtmögliche Varianz. Wenn Sie die Daten auf eine Achse oder ein Feature reduzieren würden, ist diese Hauptkomponente tatsächlich eine gute Wahl, und genau das wird PCA tun. Wenn Sie die Daten auf eindimensionale Merkmale reduzieren möchten, wird diese Hauptkomponente ausgewählt. Lassen Sie mich Ihnen eine Visualisierung zeigen, wie sich unterschiedliche Achsenwahlen auf die Projektion auswirken.

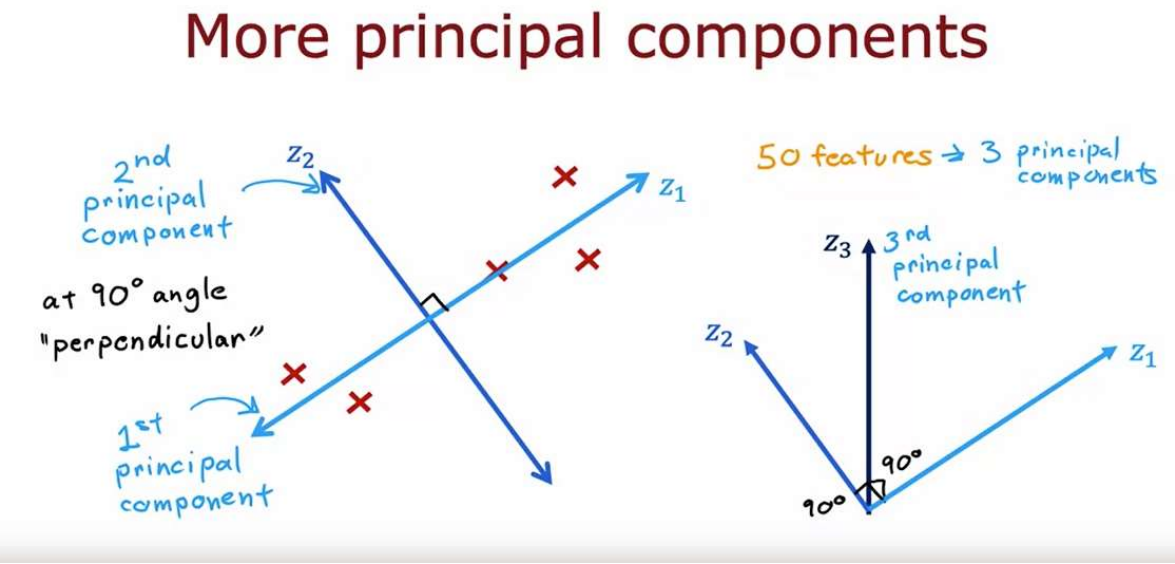




Hier haben wir 10 Trainingsbeispiele, und wenn wir diesen Schieberegler hierher schieben, können Sie in einem der optionalen Labore selbst damit experimentieren. Wenn Sie den Schieberegler hierher verschieben, ändert sich der Winkel der Z-Achse. Was Sie auf der linken Seite sehen, ist jedes der Beispiele, projiziert über dieses kurze Liniensegment im 90-Grad-Winkel zur Z-Achse. Hier rechts ist die Projektion der Daten, also der Wert dieser 10 Beispiele, Z-Koordinate. Sie bemerken, dass, wenn ich die Achse ungefähr hier einstelle, die Punkte ziemlich zusammengedrückt sind. Dies weist also einen geringeren Automatisierungsgrad der Originaldaten auf. Wenn ich dagegen beispielsweise die Z-Achse so einstelle, variieren diese Punkte viel stärker. Dadurch werden wesentlich mehr Informationen im Originaldatensatz erfasst. Aus diesem Grund entspricht die Hauptkomponente der Einstellung der Z-Achse etwa hier. Dies ist die Wahl, die PCA treffen würde, wenn Sie es bitten würden, die Daten auf eine Dimension zu reduzieren. Eine Bibliothek für maschinelles Lernen wie scikit -learn, über die Sie im nächsten Video mehr erfahren, kann Ihnen dabei helfen, die Hauptkomponente automatisch zu finden. Aber schauen wir uns etwas genauer an, wie das funktioniert. Hier sind meine x_1- und x_2-Achsen. Hier ist ein Trainingsbeispiel mit den Koordinaten 2 auf der x_1-Achse und drei auf der x_2-Achse. Nehmen wir an, PCA hat diese Richtung für die Z-Achse gefunden. Was ich hier zeichne, dieser kleine Pfeil ist ein Vektor der Länge 1, der in die Richtung dieser Z-Achse zeigt, die PCA wählen wird oder die wir gewählt haben. Es stellt sich heraus, dass dieser Vektor der Länge 1 der Vektor 0,710 ist, 0,71 etwas gerundet. Es ist tatsächlich 0,707 und dann noch ein paar andere Ziffern. Wie projizieren wir dieses Beispiel anhand dieses Beispiels mit den Koordinaten 2,3 auf der x_1-, x_2-Achse auf die z-Achse? Es stellt sich heraus, dass die Formel hierfür darin besteht, ein Skalarprodukt zwischen dem Vektor 2,3 und diesem Vektor 0,71, 0,71 zu bilden. Wenn Sie das tun, ergibt sich aus dem 2,3 - Skalarprodukt mit 0,71, 0,71 2 mal 0,71 plus 3 mal 0,71, was 3,55 entspricht. Das bedeutet, dass der Abstand vom Ursprung dieses Punktes hier 3,55 beträgt.

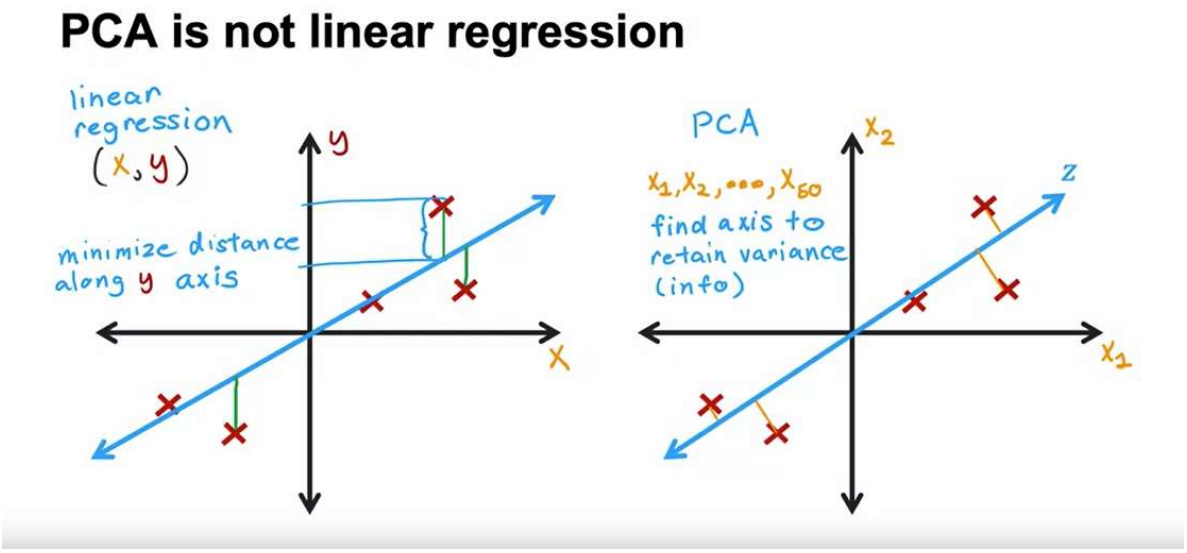


Wenn wir also eine Zahl darstellen oder verwenden würden, um zu versuchen, dieses Beispiel zu erfassen, wäre diese eine Zahl 3,55. Bisher haben wir darüber gesprochen, wie man PCA verwenden kann, um Daten auf eine Dimension oder auf eine Zahl zu reduzieren. Dazu haben wir die Hauptkomponente gefunden, die manchmal auch als erste Hauptkomponente bezeichnet wird. In diesem Beispiel hatten wir dies als erste Achse gefunden. Es stellt sich heraus, dass bei Auswahl einer zweiten Achse die zweite Achse immer im 90-Grad-Winkel zur ersten Achse steht. Wenn Sie sogar eine dritte Achse wählen würden, dann wäre die dritte Achse im 90-Grad-Winkel zur ersten und zweiten Achse. In der Mathematik werden 90 Grad übrigens manchmal als Senkrechte bezeichnet. Der Begriff senkrecht bedeutet einfach im 90-Grad-Winkel. Mathematiker sagen manchmal, dass die zweite Achse, z_2, im 90-Grad-Winkel oder senkrecht zur ersten Achse, z_1, steht. Wenn Sie zusätzliche Achsen auswählen, stehen diese ebenfalls im 90-Grad-Winkel oder senkrecht zu z_1 und z_2 und zu allen anderen Achsen, die PCA auswählt. Wenn Sie 50 Features haben und drei Hauptkomponenten finden möchten, dann steht die zweite Achse, wenn dies die erste Achse ist, im 90-Grad-Winkel dazu.



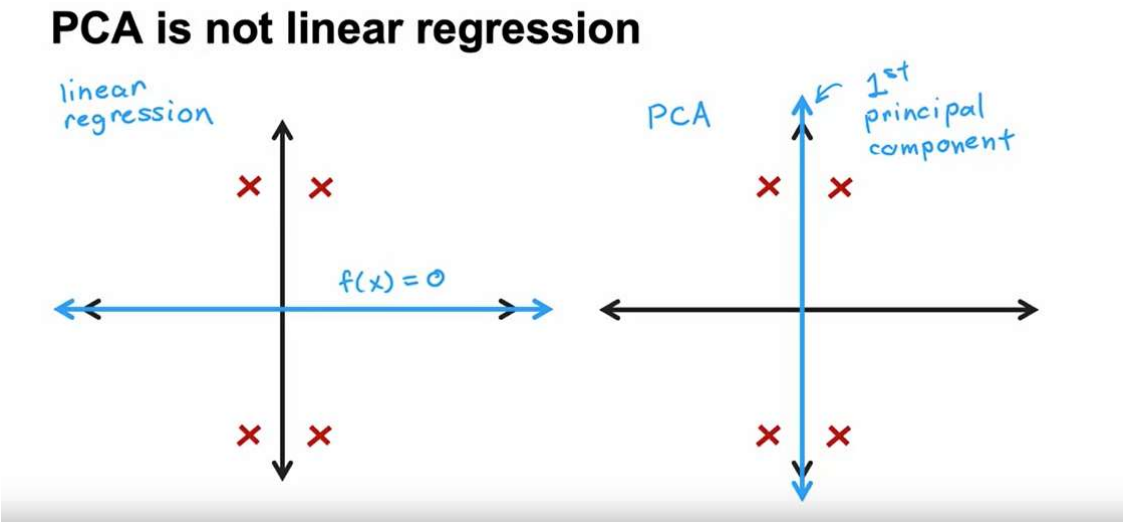
Dann steht die dritte Achse ebenfalls im 90-Grad-Winkel zur ersten und zweiten Achse. Nun wird mir oft die Frage gestellt: Wie unterscheidet sich PCA von der linearen Regression? Es stellt sich heraus, dass PCA keine lineare Regression ist, sondern ein völlig anderer Algorithmus. Lassen Sie mich

erklären, warum. Bei der linearen Regression, einem überwachten Lernalgorithmus, liegen die Daten x und y vor. Hier ist ein Datensatz, bei dem die horizontale Achse das Merkmal x und die vertikale Achse hier die Beschriftung y ist. Bei der linearen Regression versuchen Sie, eine gerade Linie so anzupassen, dass der vorhergesagte Wert so nah wie möglich an der Grundwahrheitsbezeichnung y liegt. Mit anderen Worten: Sie versuchen, die Länge dieser kleinen Liniensegmente in vertikaler Richtung zu minimieren. Sie sind gerade an der Y-Achse ausgerichtet. Im Gegensatz dazu gibt es in PCA kein Ground-Truth-Label y. Sie haben lediglich unbeschriftete Daten, X1 und X2, und außerdem versuchen Sie nicht, eine Linie anzupassen, um X1 zur Vorhersage von X2 zu verwenden. Stattdessen behandelt der Durchschnitt X1 und X2 gleich. Wir versuchen, diese Achse Z zu finden. Es stellt sich heraus, dass wir diese kleinen Liniensegmente am Ende klein machen, wenn Sie die Daten auf Z projizieren. In der linearen Regression gibt es eine Zahl Y, die eine ganz besondere Behandlung erhält. Wir versuchen immer, den Abstand zwischen der angepassten Linie und Y zu messen, weshalb diese Abstände nur in Richtung der y-Achse gemessen werden. Bei PCA hingegen können Sie viele Funktionen haben, X1, X2, vielleicht sogar bis zu X50, wenn Sie 50 Funktionen haben. Alle 50 Features werden gleich behandelt.

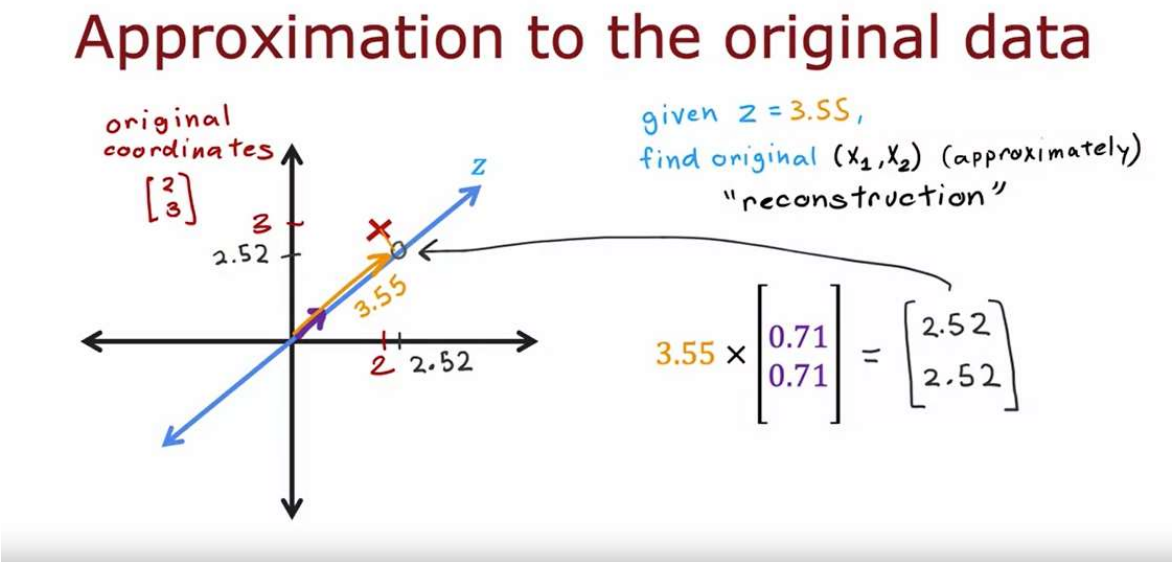


Wir versuchen lediglich, eine Z-Achse zu finden, damit bei der Projektion der Daten auf die Z-Achse mithilfe dieser Liniensegmente immer noch so viel Varianz der Originaldaten wie möglich erhalten bleibt. Ich weiß, wenn ich diese Dinge zweidimensional zeichne, haben wir nur zwei Funktionen, nämlich, dass ich auf einem flachen Computermonitor zeichnen kann. Diese Pfeile sehen aus, als wären sie vielleicht ein bisschen ähnlich. Wenn Sie jedoch mehr als zwei Merkmale haben, was in den meisten Fällen der Fall ist, ist der Unterschied zwischen linearer Regression und PCA und der Funktionsweise der Algorithmen sehr groß. Diese Algorithmen werden für völlig unterschiedliche Zwecke verwendet und liefern sehr unterschiedliche Antworten. Wenn lineare Regression verwendet wird, um eine Zielausgabe Y vorherzusagen, versucht PCA, viele Merkmale zu übernehmen und sie alle gleich zu behandeln und die Anzahl der Achsen zu reduzieren, die für eine gute Darstellung der Daten erforderlich sind. Es stellt sich heraus, dass die Maximierung der Streuung dieser Projektionen einer Minimierung der Abstände dieser Liniensegmente entspricht. Die Abstände zu den Punkten müssen sich verschieben, um auf Z projiziert zu werden. Um den Unterschied zwischen linearer Regression und PCA auf andere Weise zu veranschaulichen, wenn Sie haben einen Datensatz, der so aussieht, lineare Regression, alles, was er tun kann, ist, eine Linie anzupassen, die so aussieht. Wenn Ihr Datensatz hingegen so aussieht, wählt PCA dies als Hauptkomponente aus. Sie sollten also die lineare Regression verwenden, wenn Sie versuchen, den Wert von y vorherzusagen, und Sie sollten

PCA verwenden, wenn Sie versuchen, die Anzahl der Features in Ihrem Datensatz zu reduzieren, beispielsweise um ihn zu visualisieren.



Bevor wir dieses Video abschließen, gibt es noch etwas, was Sie mit PCA tun können: Erinnern Sie sich an dieses Beispiel, das sich bei den Koordinaten 2,3 befand. Wir haben festgestellt, dass Sie bei einer Projektion auf die Z-Achse am Ende 3,55 erhalten. Eine Sache, die Sie tun könnten: Wenn Sie ein Beispiel haben, bei dem Z gleich 3,55 ist, und nur diese eine Zahl Z, 3,55, haben, können wir dann versuchen herauszufinden, was das ursprüngliche Beispiel war? Es stellt sich heraus, dass es in der PCA einen Schritt namens Rekonstruktion gibt, der darin besteht, von dieser einen Zahl Z gleich 3,55 zurück zu den ursprünglichen beiden Zahlen X1 und X2 zu gelangen. Es stellt sich heraus, dass Sie nicht über genügend Informationen verfügen, um X1 und X2 genau zu ermitteln, aber Sie können versuchen, sie anzunähern. Die Formel lautet insbesondere: Nehmen Sie die Zahl 3,55, die Z ist, und multiplizieren Sie sie mit der Länge eines Vektors, die wir gerade hatten, also 0,71, 0,71. Das ergibt am Ende 2,52, 2,52, was diesem Punkt hier entspricht. Wir können das ursprüngliche Trainingsbeispiel, bei dem es sich um die Koordinaten 2, 3 handelte, mit diesem neuen Punkt hier annähern, der bei 2,52, 2,52 liegt. Der Unterschied zwischen dem ursprünglichen Punkt und dem projizierten Punkt ist hier dieses kleine Liniensegment. In diesem Fall ist 2,52 keine schlechte Näherung, 2,52 ist nicht so weit von 2, 3 entfernt.



Mit nur einer Zahl könnten wir eine vernünftige Annäherung an die Koordinaten des ursprünglichen Trainingsbeispiels erhalten. Dies wird als Rekonstruktionsschritt der PCA bezeichnet. Zusammenfassend betrachtet betrachtet der PCA-Algorithmus Ihre Originaldaten und wählt eine oder mehrere neue Achsen, Z oder vielleicht Z1 und Z2, zur Darstellung Ihrer Daten aus, indem er Ihren Originaldatensatz nimmt und ihn auf Ihre neue Achse oder Achse projiziert. Dadurch erhalten Sie einen kleineren Zahlensatz, sodass Sie Ihre Daten bei Bedarf grafisch darstellen können. Du siehst die Mathematik. Schauen wir uns nun an, wie Sie dies im Code implementieren können. Im nächsten Video schauen wir uns an, wie Sie PCA mithilfe der Bibliothek scikit -learn selbst nutzen können. Kommen wir zum nächsten Video.

PCA im Code (optional)

In diesem Video werfen wir einen Blick darauf, wie Sie die Bibliothek scikit -learn zur Implementierung von PCA verwenden können. Dies sind die Hauptschritte. Wenn Ihre Features sehr unterschiedliche Wertebereiche annehmen, können Sie zunächst eine Vorverarbeitung durchführen, um die Features so zu skalieren, dass sie vergleichbare Wertebereiche annehmen. Wenn Sie sich die Merkmale verschiedener Länder ansehen, nehmen diese Merkmale sehr unterschiedliche Wertebereiche an. Das BIP könnte Billionen Dollar betragen, während andere Features weniger als 100 betragen. Die Feature-Skalierung in solchen Anwendungen wäre wichtig, um PCA dabei zu helfen, eine gute Auswahl an Achsen für Sie zu finden. Der nächste Schritt besteht dann darin, den PCA-Algorithmus auszuführen, um die Daten „anzupassen“, um zwei oder drei neue Achsen zu erhalten, Z_1, Z_2 und vielleicht Z_3. Hier gehe ich davon aus, dass Sie zwei oder drei Achsen benötigen, wenn Sie die Daten in 2D oder 3D visualisieren möchten. Wenn Sie eine Anwendung haben, bei der Sie mehr als zwei oder drei Achsen benötigen, können Sie mit der PCA-Implementierung auch mehr als zwei oder drei Achsen erhalten, allerdings wäre die Visualisierung dann schwieriger. In scikit -learn verwenden Sie dazu die Fit-Funktion bzw. die Fit-Methode. Die Anpassungsfunktion in PCA führt automatisch eine Mittelwertnormalisierung durch und subtrahiert den Mittelwert jedes Merkmals. Sie müssen also keine separate Mittelwertnormalisierung durchführen. Nach dem Ausführen der Anpassungsfunktion erhalten Sie die neuen Achsen Z_1, Z_2, vielleicht Z_3, und in PCA nennen wir diese auch die Hauptkomponenten, wobei Z_1 die erste Hauptkomponente, Z_2 die zweite Hauptkomponente und Z_3 die dritte ist Hauptbestandteil. Danach würde ich empfehlen, einen Blick darauf zu werfen, inwieweit jede dieser neuen Achsen oder jede dieser neuen Hauptkomponenten die Varianz in Ihren Daten erklärt. Auf der nächsten Folie zeige ich anhand eines konkreten Beispiels, was dies bedeutet. Dadurch erhalten Sie jedoch einen Eindruck davon, ob die Projektion der Daten auf diese Achsen dabei hilft, den größten Teil der Variabilität bzw. die meisten Informationen im Original beizubehalten Datensatz. Dies geschieht mithilfe der erläuterten Varianzverhältnisfunktion. Schließlich können Sie die Daten transformieren, also einfach auf die neuen Achsen und die neuen Hauptkomponenten projizieren, was Sie mit der Transformationsmethode tun. Dann hätten Sie für jedes Trainingsbeispiel nur zwei oder drei Zahlen und können diese zwei oder drei Zahlen dann grafisch darstellen, um Ihre Daten zu visualisieren. Im Detail sieht PCA im Code so aus. Hier ist der Datensatz X mit sechs Beispielen. X entspricht dem NumPy- Array, die sechs Beispiele hier. Um PCA auszuführen, um diese Daten von zwei Zahlen, X_1, X_2, auf nur eine Zahl Z zu reduzieren, würden Sie PCA ausführen und bitten, eine Hauptkomponente anzupassen. N Komponenten sind hier gleich eins und passen PCA an X an. Pca_1 hier ist meine Notation für PCA mit einer einzelnen Hauptkomponente und einer einzelnen Achse. Es stellt sich heraus, dass dies 0,992 ist, wenn Sie pca_1.explained_variance_ratio ausdrucken würden. Dies zeigt Ihnen, dass in diesem Beispiel bei Auswahl einer Achse 99,2 Prozent der Variabilität oder der Informationen im Originaldatensatz erfasst werden. Wenn Sie schließlich jedes dieser Trainingsbeispiele nehmen und es auf eine einzelne Zahl projizieren möchten, rufen Sie dies auf und es wird dieses Array mit sechs Zahlen ausgegeben, die Ihren sechs Trainingsbeispielen entsprechen.

Das erste Trainingsbeispiel 1,1 ergibt beispielsweise, projiziert auf die Z-Achse, diese Zahl, 1,383, und so weiter. Wenn Sie diesen Datensatz also mit nur einer Dimension visualisieren würden, wäre dies die Zahl, die ich zur Darstellung des ersten Beispiels verwende. Das zweite Beispiel soll diese Zahl sein und so weiter. Ich hoffe, Sie werfen einen Blick auf das optionale Labor, in dem Sie sehen, dass diese sechs Beispiele auf diese Achse, auf diese Linie, die jetzt Y ist, projiziert wurden. Alle sechs Beispiele liegen jetzt auf dieser Linie, die so aussieht. Das erste Trainingsbeispiel, das 1,1 war, wurde auf dieses Beispiel abgebildet, das einen Abstand von 1,38 vom Ursprung hat, daher ist dieser 1,38. Nur noch ein kurzes Beispiel. Bei diesen Daten handelt es sich um zweidimensionale Daten, die wir auf eine Dimension reduziert haben. Was wäre, wenn Sie zwei Hauptkomponenten berechnen würden? Beginnt mit zwei Dimensionen und endet dann auch mit zwei Dimensionen. Dies ist für die Visualisierung nicht besonders nützlich, aber es könnte uns helfen, besser zu verstehen, wie PCA und wie der PCA-Code funktioniert. Hier ist derselbe Code, außer dass ich n Komponenten in zwei geändert habe. Ich werde den Algorithmus bitten, zwei Hauptkomponenten zu finden. Wenn Sie dies tun, beträgt das Erklärungsverhältnis pca_2 0,992, 0,008. Das bedeutet, dass z_1, die ersten Hauptkomponenten, immer noch stetig, 99,2 Prozent der Varianz erklären, Z_2, die zweiten Hauptkomponenten oder die zweite Achse, 0,8 Prozent der Varianz erklären. Diese beiden Zahlen zusammen ergeben eins. Denn während diese Daten zweidimensional sind, erklären die beiden Achsen Z_1 und Z_2 zusammen 100 Prozent der Varianz in den Daten. Wenn Sie die Daten unseres Projekts auf die Z_1- und Z_2-Achsen transformieren würden, erhalten Sie Folgendes: Das erste Trainingsbeispiel wird nun auch mit diesen beiden Zahlen versehen, was seiner Projektion auf z_1 und z_2 entspricht, und das zweite Beispiel: Das heißt, projiziert auf z_1 und z_2, werden diese beiden Zahlen. Wenn Sie die Originaldaten ungefähr rekonstruieren würden, ist dies z_1 und dieses z_2, dann hat das erste Trainingsbeispiel, das ein [1, 1] war, einen Abstand von 1,38 auf der z_1-Achse, diese Zahl und den Abstand hier von 0,29 dieser Abstand auf der z_2-Achse, und die Rekonstruktion sieht tatsächlich genauso aus wie die Originaldaten. Denn wenn Sie zweidimensionale Daten auf zweidimensionale Daten reduzieren oder nicht wirklich reduzieren, gibt es keine Annäherung und Sie können zu Ihrem ursprünglichen Datensatz mit den Projektionen auf z_1 und z_2 zurückkehren. So sieht der Code zum Ausführen von PCA aus. Ich hoffe, Sie werfen einen Blick auf das optionale Labor, in dem Sie selbst mehr damit spielen können. Versuchen Sie auch, die Parameter zu variieren. Sehen Sie sich ein konkretes Beispiel an, um Ihr Verständnis für die Funktionsweise von PCA zu vertiefen. Bevor ich zum Abschluss komme, möchte ich noch ein paar Ratschläge für die Anwendung von PCA geben. PCA wird häufig zur Visualisierung verwendet, bei der Sie Daten auf zwei oder drei Zahlen reduzieren, damit Sie sie grafisch darstellen können. Wie Sie in einem früheren Video mit den Daten zu verschiedenen Ländern gesehen haben, können Sie sich verschiedene Länder vorstellen. Es gibt einige andere Anwendungen von PCA, von denen Sie möglicherweise gelegentlich hören. Das war früher beliebter, vielleicht vor 10, 15, 20 Jahren, aber heute ist es viel weniger beliebt. Eine weitere mögliche Anwendung von PCA ist die Datenkomprimierung. Wenn Sie beispielsweise über eine Datenbank mit vielen verschiedenen Autos verfügen und 50 Funktionen pro Auto haben, diese aber einfach zu viel Platz in Ihrer Datenbank einnimmt oder vielleicht die Übertragung von 50 Nummern über das Internet einfach zu lange dauert. Dann könnten Sie diese 50 Funktionen auf eine kleinere Anzahl von Funktionen reduzieren. Es können 10 Features mit 10 Achsen oder 10 Hauptkomponenten sein. So einfach können Sie 10-dimensionale Daten visualisieren, aber das kostet 1/5 des Speicherplatzes oder vielleicht 1/5 der erforderlichen Netzwerkübertragungskosten. Vor vielen Jahren habe ich gesehen, dass PCA für diese Anwendung verwendet wird, und zwar noch häufiger, aber heute, da moderne Speicher in der Lage sind, ziemlich große Datensätze zu speichern, und moderne Netzwerke sind sie in der Lage, schneller und mehr Daten als je zuvor zu übertragen. Ich sehe diesen Einsatz viel seltener als Anwendung von PCA. Eine der Anwendungen von PCA, die vor vielleicht 10 oder 20 Jahren noch häufiger vorkamen, heute jedoch weitaus seltener, ist die Verwendung zur Beschleunigung des Trainings eines

überwachten Lernmodells. Die Idee dahinter ist: Wenn Sie 1.000 Features hätten und 1.000 Features hätten, könnte der überwachte Lernalgorithmus Sie langsam laufen lassen. Vielleicht können Sie es mit PCA auf 100 Funktionen reduzieren, dann ist Ihr Datensatz grundsätzlich kleiner und Ihr überwachter Lernalgorithmus läuft möglicherweise schneller. Dies führte früher zu einem Unterschied in der Laufzeit einiger älterer Generationen von Lernalgorithmen, beispielsweise wenn Sie über Support-Vektor-Maschinen verfügten. Dadurch wird eine Support-Vektor-Maschine beschleunigt. Aber es stellt sich heraus, dass dies bei modernen Algorithmen für maschinelles Lernen, Algorithmen wie Deep Learning, nicht wirklich gilt und es viel üblicher ist, einfach den hochdimensionalen Datensatz zu nehmen und ihn beispielsweise in Ihr neuronales Netzwerk einzuspeisen. Anstatt PCA auszuführen, da PCA auch einen gewissen Rechenaufwand verursacht. Möglicherweise hören Sie in einigen anderen Forschungsarbeiten davon, aber ich sehe nicht mehr, dass dies wirklich viel getan wird. Aber die häufigste Sache, für die ich PCA heute verwende, ist die Visualisierung, und dann finde ich es sehr nützlich, die Dimensionsdaten zu reduzieren, um sie zu visualisieren. Vielen Dank, dass Sie bis zum Ende der optionalen Videos für diese Woche bei mir geblieben sind. Ich wünsche Ihnen viel Spaß beim Erlernen von PCA und wünsche Ihnen, dass Sie bei der Erstellung eines neuen Datensatzes eine nützliche Anleitung finden, mit der Sie die Dimension des Datensatzes auf zwei oder drei Dimensionen reduzieren können Visualisieren Sie es und gewinnen Sie hoffentlich neue Erkenntnisse über Ihre Datensätze. Es hat mir oft geholfen, meine eigenen Datensätze zu verstehen, und ich hoffe, dass Sie es ebenso nützlich finden. Vielen Dank, dass Sie sich diese Videos angesehen haben, und ich freue mich darauf, Sie nächste Woche zu sehen.

Lernziele

- Verstehen Sie Schlüsselbegriffe wie Rückkehr, Zustand, Aktion und Richtlinie, die für das verstärkende Lernen gelten
- Verstehen Sie die Bellman-Gleichungen
- Verstehen Sie die Zustandsaktionswertfunktion
- Kontinuierliche Zustandsräume verstehen
- Bauen Sie ein tiefes Q-Learning-Netzwerk auf

Was ist Reinforcement Learning?

Willkommen zu dieser letzten Woche der Spezialisierung auf maschinelles Lernen. Es ist ein bisschen bittersüß für mich, dass wir uns dem Ende dieser Spezialisierung nähern, aber ich freue mich auf diese Woche und werde einige spannende Ideen zum Reinforcement Learning mit Ihnen teilen. Beim maschinellen Lernen ist Reinforcement Learning eine dieser Ideen, die zwar heute noch nicht sehr weit verbreitet in kommerziellen Anwendungen angewendet wird, aber eine der Säulen des maschinellen Lernens darstellt. Und es gibt jede Menge spannende Forschungsergebnisse, die es jeden Tag untermauern und verbessern. Werfen wir also zunächst einen Blick darauf, was Reinforcement Learning ist. Beginnen wir mit einem Beispiel. Hier ist ein Bild eines autonomen Hubschraubers. Das ist eigentlich der autonome Stanford-Hubschrauber, wiegt 32 Pfund und steht gerade in meinem Büro. Wie viele andere autonome Hubschrauber ist er mit einem Bordcomputer, GPS, Beschleunigungsmessern, Gyroskopen und einem Magnetkompass ausgestattet, sodass er jederzeit recht genau weiß, wo er sich befindet. Und wenn ich Ihnen die Schlüssel zu diesem Hubschrauber geben würde und Sie bitten würde, ein Programm zu schreiben, um ihn zu fliegen, wie würden Sie das tun? Funkgesteuerte Hubschrauber werden mit solchen Joysticks gesteuert. Die Aufgabe besteht also darin, zehnmal pro Sekunde die Position, Ausrichtung, Geschwindigkeit usw. des Hubschraubers zu bestimmen. Und Sie müssen entscheiden, wie Sie diese beiden Steuerknüppel bewegen, um den Hubschrauber in der Luft im Gleichgewicht zu halten. Ich bin übrigens selbst sowohl ferngesteuerte Helikopter als auch Quad-Rotor-Drohnen geflogen. Und funkgesteuerte

Hubschrauber sind tatsächlich um einiges schwieriger zu fliegen und um einiges schwieriger in der Luft im Gleichgewicht zu bleiben. Wie schreibt man also ein Programm, das dies automatisch erledigt?

Lassen Sie mich Ihnen ein lustiges Video von etwas zeigen, für das wir einen autonomen Stanford-Hubschrauber eingesetzt haben. Hier ist ein Video davon, wie es unter der Kontrolle eines Reinforcement-Learning-Algorithmus fliegt. Und lass mich das Video abspielen. Ich war an diesem Tag tatsächlich der Kameramann und das ist der Hubschrauber, der computergesteuert fliegt, und wenn ich das Video verkleinere, sieht man die am Himmel gepflanzten Bäume. Durch Reinforcement Learning haben wir diesen Hubschrauber tatsächlich dazu gebracht, das Fliegen auf dem Kopf zu lernen. Wir sagten ihm, er solle kopfüber fliegen. Und so wurde verstärktes Lernen genutzt, um Hubschrauber dazu zu bringen, eine Vielzahl von Stunts zu fliegen, oder wir nennen sie Kunstflugmanöver. Übrigens, wenn Sie daran interessiert sind, weitere Videos zu sehen, können Sie diese auch unter dieser URL ansehen. Wie bringt man einen Hubschrauber mithilfe von Reinforcement Learning dazu, selbst zu fliegen? Die Aufgabe besteht darin, die Position des Hubschraubers festzulegen, um zu entscheiden, wie die Steuerknüppel bewegt werden sollen. Beim verstärkenden Lernen nennen wir die Position, Ausrichtung, Geschwindigkeit usw. des Hubschraubers den Zustand s. Die Aufgabe besteht also darin, eine Funktion zu finden, die den Zustand des Hubschraubers auf eine Aktion a abbildet, also darauf, wie weit die beiden Steuerknüppel gedrückt werden müssen, um den Hubschrauber im Gleichgewicht in der Luft und im Flug zu halten, ohne abzustürzen. Eine Möglichkeit, dieses Problem zu lösen, besteht darin, überwachtes Lernen zu verwenden. Es stellt sich heraus, dass dies kein großartiger Ansatz für den autonomen Hubschrauberflug ist. Aber man könnte sagen, wenn wir eine Reihe von Zustandsbeobachtungen machen könnten und uns vielleicht einen erfahrenen menschlichen Piloten sagen könnten, was die besten Maßnahmen sind. Sie könnten dann ein neuronales Netzwerk mithilfe von überwachtem Lernen trainieren, um direkt die Zuordnung von den Zuständen s, die ich hier x nenne, zu einer Aktion a zu lernen, die ich hier die Bezeichnung y nenne. Aber es stellt sich heraus, dass, wenn sich der Hubschrauber durch die Luft bewegt, eigentlich sehr unklar ist, was genau die richtige Maßnahme ist. Neigen Sie etwas nach links oder viel mehr nach links oder erhöhen Sie die Helikopterbelastung ein wenig oder stark? Es ist tatsächlich sehr schwierig, einen Datensatz von x und der idealen Aktion y zu erhalten. Aus diesem Grund funktioniert der Ansatz des überwachten Lernens für viele Aufgaben der Steuerung eines Roboters wie eines Hubschraubers und anderer Roboter nicht gut und wir verwenden stattdessen verstärkendes Lernen. Ein wichtiger Input für ein verstärkendes Lernen ist nun die sogenannte Belohnung oder Belohnungsfunktion, die dem

Hubschrauber mitteilt, wann es ihm gut und wann schlecht geht. Die Art und Weise, wie ich mir die Belohnungsfunktion vorstelle, ist ein bisschen so, als würde man einen Hund trainieren. Als ich aufwuchs, hatte meine Familie einen Hund und es war meine Aufgabe, dem Hund oder Welpen das Verhalten beizubringen. Wie bringt man einen Welpen dazu, sich gut zu benehmen? Na ja, so viel kann man dem Welpen nicht zeigen. Stattdessen lässt du es seine Sache machen und wann immer es etwas Gutes tut, gehst du, guter Hund. Und wann immer sie etwas Schlimmes getan haben, gehst du, böser Hund. Und dann lernt es hoffentlich von selbst, mehr von den guten und weniger von den bösen Dingen zu tun. Das Training mit dem Reinforcement-Learning-Algorithmus ist also so. Wenn der Helikopter gut fliegt, sagen Sie, guter Helikopter, und wenn er etwas Schlechtes wie einen Absturz macht, sagen Sie, schlechter Helikopter. Und dann ist es die Aufgabe des Reinforcement-Learning-Algorithmus herauszufinden, wie man mehr gute Helikopter- und weniger schlechte Helikopter- Ergebnisse erzielt. Eine Möglichkeit, darüber nachzudenken, warum Reinforcement Learning so wirkungsvoll ist, besteht darin, ihm zu sagen, was er tun soll, und nicht, wie er es tun soll. Und die Angabe der Belohnungsfunktion anstelle der optimalen Aktion gibt Ihnen viel mehr Flexibilität bei der Gestaltung des Systems. Konkret können Sie für das Fliegen des Hubschraubers, wann immer er gut fliegt, eine Belohnung von plus eins für jede Sekunde geben, in der er gut fliegt. Und wenn es schlecht fliegt, können Sie ihm vielleicht eine negative Belohnung geben, oder wenn es einmal abstürzt, können Sie ihm eine sehr hohe negative Belohnung geben, beispielsweise minus 1.000. Und das wäre ein Anreiz für den Hubschrauber, viel mehr Zeit gut zu fliegen und hoffentlich nie abzustürzen. Aber hier ist noch ein lustiges Video. Ich habe viele Jahre lang die Analogie „Guter Hund, böser Hund“ für das Verstärkungslernen verwendet. Und dann gelang es mir eines Tages tatsächlich, einen Roboterhund in die Hände zu bekommen, und ich konnte diese Verstärkungsmethode „Guter Hund, böser Hund“ tatsächlich nutzen, um einem Roboterhund beizubringen, Hindernisse zu überwinden. Dies ist also ein Video eines Roboterhundes, der durch Verstärkungslernen, das ihn belohnt, indem er sich zum linken Bildschirmrand bewegt, gelernt hat, seine Füße vorsichtig zu platzieren oder über verschiedene Hindernisse zu klettern. Und wenn Sie darüber nachdenken, was nötig ist, um einen Hund auf diese Weise zu programmieren, habe ich keine Ahnung, ich weiß wirklich nicht, wie ich ihm sagen soll, wie er seine Beine am besten platzieren soll, um ein bestimmtes Hindernis zu überwinden. All diese Dinge wurden vom Roboter automatisch herausgefunden, indem er ihm einfach Belohnungen gab, die ihn dazu anspornten, Fortschritte in Richtung des Ziels auf der linken Seite des Bildschirms zu machen. Heutzutage wird Reinforcement Learning erfolgreich auf eine Vielzahl von Anwendungen angewendet, die von der Steuerung von Robotern reichen.

Robotic Dog Example

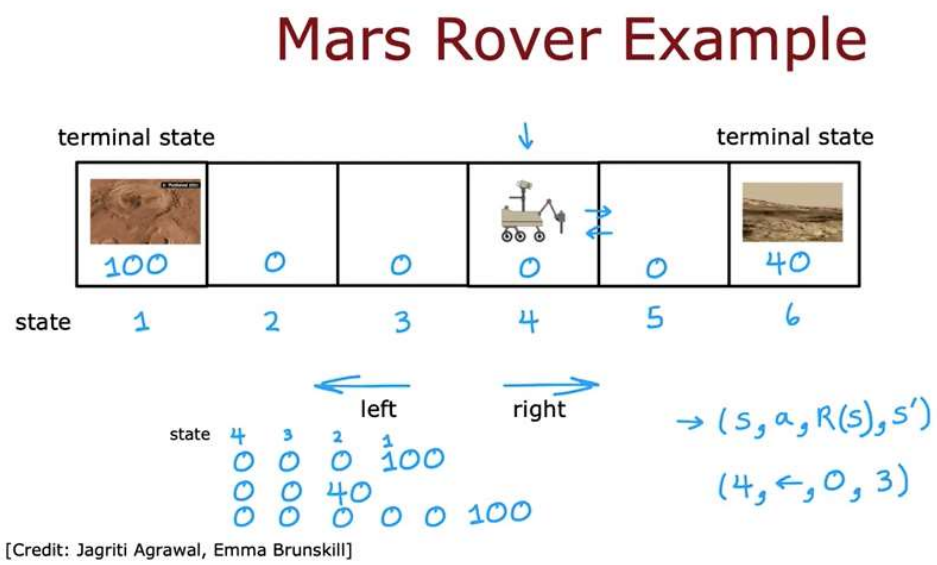
[Thanks to Zico Kolter]

Und tatsächlich implementieren Sie später in dieser Woche im Übungslabor selbst einen Reinforcement-Learning-Algorithmus, um eine Mondlandefähre in einer Simulation zu landen. Es wurde auch zur Fabrikoptimierung verwendet. Wie ordnen Sie die Dinge in der Fabrik neu, um den Durchsatz und die Effizienz sowie den Finanzaktienhandel zu maximieren? Einer meiner Freunde arbeitete beispielsweise an einer effizienten Aktienausführung. Wenn Sie sich also dazu entschließen, in den nächsten Tagen eine Million Aktien zu verkaufen, möchten Sie vielleicht nicht plötzlich eine Million Aktien an die Börse werfen, weil sich dadurch die Kurse zu Ihren Ungunsten auswirken würden. Was ist also der beste Weg, Ihre Geschäfte im Laufe der Zeit zu ordnen, damit Sie die Aktien, die Sie verkaufen möchten, verkaufen und hoffentlich den bestmöglichen Preis dafür erzielen können?

Schließlich gab es auch viele Anwendungen des Verstärkungslernens beim Spielen, von Dame über Schach bis zum Kartenspiel Bridge to go sowie beim Spielen vieler Videospiele. Das ist also verstärkendes Lernen. Auch wenn Reinforcement Learning nicht annähernd so häufig eingesetzt wird wie überwachtes Lernen, wird es auch heute noch in einigen wenigen Anwendungen eingesetzt. Und die Kernidee besteht darin, dass Sie dem Algorithmus nicht sagen müssen, welches die richtige Ausgabe y für jede einzelne Eingabe ist, sondern dass Sie stattdessen lediglich eine Belohnungsfunktion angeben müssen, die ihm sagt, wann er gut und wann schlecht abschneidet. Und es ist die Aufgabe des Algorithmus, automatisch herauszufinden, wie man gute Aktionen auswählt. Kommen wir nun zum nächsten Video, in dem wir das Problem des Verstärkungslernens formalisieren und auch mit der Entwicklung von Algorithmen zur automatischen Auswahl guter Aktionen beginnen

Beispiel für einen Marsrover

Um den Reinforcement-Learning-Formalismus zu vervollständigen, können wir, anstatt etwas so Kompliziertes wie einen Hubschrauber oder einen Roboterhund zu betrachten, ein vereinfachtes Beispiel verwenden, das lose vom Marsrover inspiriert ist. Dies ist eine Übernahme des Beispiels von Stanford-Professorin Emma Branskill und einer meiner Mitarbeiterinnen, Jagriti Agrawal, die tatsächlich Code geschrieben hatten, der derzeit tatsächlich den Marsrover steuert, der mir auch beim Durchsprechen und bei der Entwicklung dieses Beispiels geholfen hat. Lass uns einen Blick darauf werfen. Wir entwickeln Reinforcement Learning anhand eines vereinfachten Beispiels, das vom Marsrover inspiriert ist.



[Credit: Jagriti Agrawal, Emma Brunskill]

In dieser Anwendung kann sich der Rover in einer von sechs Positionen befinden, wie durch die sechs Kästchen hier gezeigt. Der Rover könnte beispielsweise in der hier gezeigten vierten Box starten. Die Position des Mars-Rover wird beim Reinforcement Learning als Zustand bezeichnet, und ich werde diese sechs Zustände Zustand 1, Zustand 2, Zustand 3, Zustand 4, Zustand 5 und Zustand 6 nennen, und so startet der Rover ausgeschaltet im Zustand 4. Nun wurde der Rover zum Mars geschickt, um zu versuchen, verschiedene wissenschaftliche Missionen durchzuführen. Es kann an verschiedene Orte gehen, um seine Sensoren wie einen Bohrer, ein Radar oder ein Spektrometer zu verwenden, um das Gestein an verschiedenen Orten auf dem Planeten zu analysieren, oder es kann an verschiedene Orte gehen, um interessante Bilder zu machen, die Wissenschaftler auf der Erde betrachten können. In diesem Beispiel weist Zustand 1 hier links eine sehr interessante Oberfläche auf, die Wissenschaftler gerne mit dem Rover beproben würden. Zustand 6 hat auch eine ziemlich interessante Oberfläche, die Wissenschaftler gerne vom Rover beproben würden, aber nicht so interessant wie Zustand 1. Wir würden die wissenschaftliche Mission eher in Zustand 1 durchführen als in Zustand 6, aber Zustand 1 ist weiter entfernt. Die Art und Weise, wie wir Zustand 1 als potenziell wertvoller darstellen, erfolgt über die Belohnungsfunktion. Die Belohnung in Stufe 1 ist 100, und die Belohnung in Stufe 6 ist 40, und die Belohnungen in allen anderen Stufen dazwischen werde ich als Belohnung von Null schreiben, weil es nicht so viel interessante Wissenschaft gibt Dies kann in den Zuständen 2, 3, 4 und 5 erfolgen. Bei jedem Schritt kann der Rover eine von zwei Aktionen auswählen. Es kann entweder nach links oder nach rechts gehen. Die Frage ist: Was soll der Rover tun? Beim verstärkenden Lernen legen wir großen Wert auf die Belohnungen, denn so wissen wir, ob es dem Roboter gut oder schlecht geht. Schauen wir uns einige Beispiele dafür an, was passieren könnte, wenn der Roboter ausgehend von Zustand 4 nach links gehen würde. Dann erhält er zunächst, ausgehend von Zustand 4, eine Belohnung von Null, und nachdem er nach links gegangen ist, gelangt er zu Zustand 3, wo er erhält erneut eine Belohnung von Null. Dann gelangt es zu Zustand 2, erhält die Belohnung 0 und schließlich nur noch Zustand 1, wo es eine Belohnung von 100 erhält. Für diese Anwendung gehe ich davon aus, dass es entweder Zustand 1 oder Zustand 6 erhält der Tag geht zu Ende. Beim verstärkenden Lernen nennen wir dies manchmal einen Endzustand. Das bedeutet, dass man, nachdem man einen dieser Endzustände erreicht hat, in diesem Zustand eine Belohnung erhält, danach aber nichts mehr passiert. Vielleicht geht den Robotern der Treibstoff aus oder sie haben keine Zeit mehr für den Tag, weshalb sie nur entweder die 100er- oder die 40er-Belohnung genießen können, aber das war's dann auch schon für heute. Danach können keine weiteren Belohnungen mehr verdient werden. Anstatt nach links zu gehen, könnte sich der Roboter nun auch dafür entscheiden, nach rechts zu gehen. In diesem Fall hätte er ab

Zustand 4 zunächst eine Belohnung von Null, bewegt sich dann nach rechts und gelangt zu Zustand 5, wo er eine weitere Belohnung von erhält Null, und dann gelangt es zu diesem anderen Endzustand auf der rechten Seite, Zustand 6 und erhält eine Belohnung von 40. Aber nach links und nach rechts zu gehen sind die einzigen Optionen. Eine Sache, die der Roboter tun könnte, ist, dass er von Zustand 4 aus starten und sich entscheiden kann, sich nach rechts zu bewegen. Es geht von Zustand 4 nach 5, erhält in Zustand 4 eine Belohnung von Null und in Zustand 5 eine Belohnung von Null, und dann ändert es vielleicht seine Meinung und beschließt, wie folgt nach links zu gehen. In diesem Fall wird es erhalten eine Belohnung von Null bei Zustand 4, bei Zustand 3, bei Zustand 2 und dann die Belohnung von 100, wenn er Zustand 1 erreicht. In dieser Abfolge von Aktionen und Zuständen verschwendet der Roboter mehr Zeit. Das ist vielleicht keine besonders gute Möglichkeit, Maßnahmen zu ergreifen, aber es ist eine Möglichkeit, die der Algorithmus auswählen könnte, aber hoffentlich werden Sie diese nicht auswählen. Zusammenfassend lässt sich sagen, dass sich der Roboter bei jedem Zeitschritt in einem Zustand befindet, den ich S nenne, und er kann eine Aktion auswählen, und er genießt auch einige Belohnungen, R von S, die er aus diesem Zustand erhält. Als Ergebnis dieser Aktion gelangt es in einen neuen Zustand S-Primzahl. Als konkretes Beispiel: Als sich der Roboter in Zustand 4 befand und die Aktion ausführte, ging er nach links und genoss möglicherweise nicht die Belohnung von Null, die mit diesem Zustand 4 verbunden ist, und er wird keinen neuen Zustand 3 haben. Wenn Sie mehr darüber erfahren Bei bestimmten Algorithmen für das verstärkende Lernen sehen Sie, dass diese vier Dinge, der Zustand, die Aktion, die Belohnung und der nächste Zustand, der im Grunde jedes Mal passiert, wenn Sie eine Aktion ausführen, nur Kernelemente dessen sind, was Algorithmen für das verstärkende Lernen bei der Entscheidung berücksichtigen wie man Maßnahmen ergreift. Nur der Klarheit halber: Die Belohnung hier, R von S, ist die Belohnung, die mit diesem Zustand verbunden ist. Diese Belohnung von Null ist eher mit Zustand 4 als mit Zustand 3 verbunden. Das ist der Formalismus, wie eine Reinforcement-Learning-Anwendung funktioniert. Schauen wir uns im nächsten Video an, wie wir genau angeben, was der Reinforcement-Learning-Algorithmus tun soll. Insbesondere werden wir über eine wichtige Idee beim Reinforcement Learning sprechen, die Rückkehr genannt wird. Fahren wir mit dem nächsten Video fort, um zu sehen, was das bedeutet.

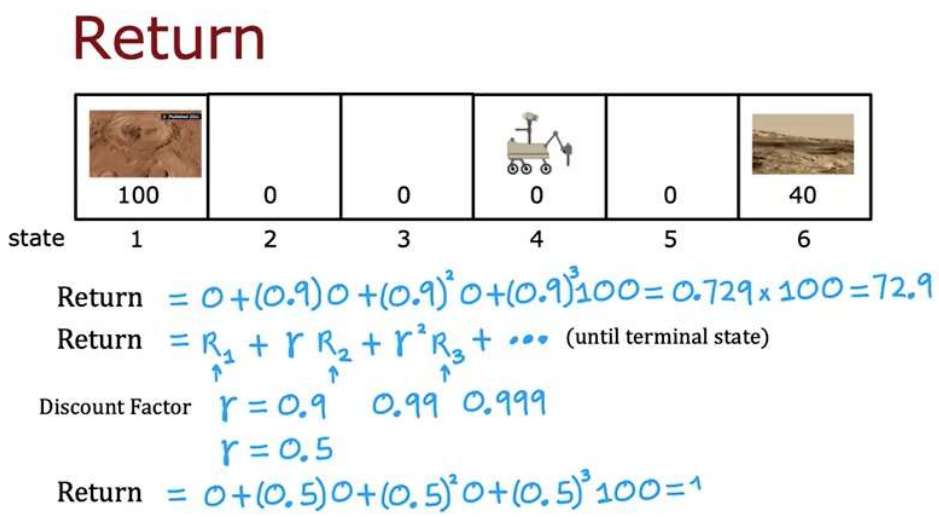
Die Rückkehr beim verstärkten Lernen

Im letzten Video haben Sie gesehen, welche Stadien es bei der Anwendung von Reinforcement Learning gibt und wie Sie je nach den von Ihnen ergriffenen Maßnahmen unterschiedliche Stadien durchlaufen und auch in den Genuss unterschiedlicher Belohnungen kommen. Aber woher wissen Sie, ob ein bestimmter Belohnungssatz besser oder schlechter ist als ein anderer Belohnungssatz?

Die Rendite des verstärkenden Lernens, die wir in diesem Video definieren, ermöglicht es uns, dies zu erfassen. Während wir dies durchgehen, könnte Ihnen ein Vergleich hilfreich sein: Wenn Sie sich vorstellen, dass Sie einen Fünf-Dollar-Schein zu Füßen haben, können Sie nach unten greifen und ihn aufheben, oder Sie können eine halbe Stunde quer durch die Stadt laufen und nach einer halbe Stunde laufen nehmen Sie einen 10-Dollar-Schein. Welches würdest du lieber wählen? Zehn Dollar sind viel besser als fünf Dollar, aber wenn Sie eine halbe Stunde laufen müssen, um den 10-Dollar-Schein zu holen, dann ist es vielleicht bequemer, stattdessen einfach den Fünf-Dollar-Schein mitzunehmen.

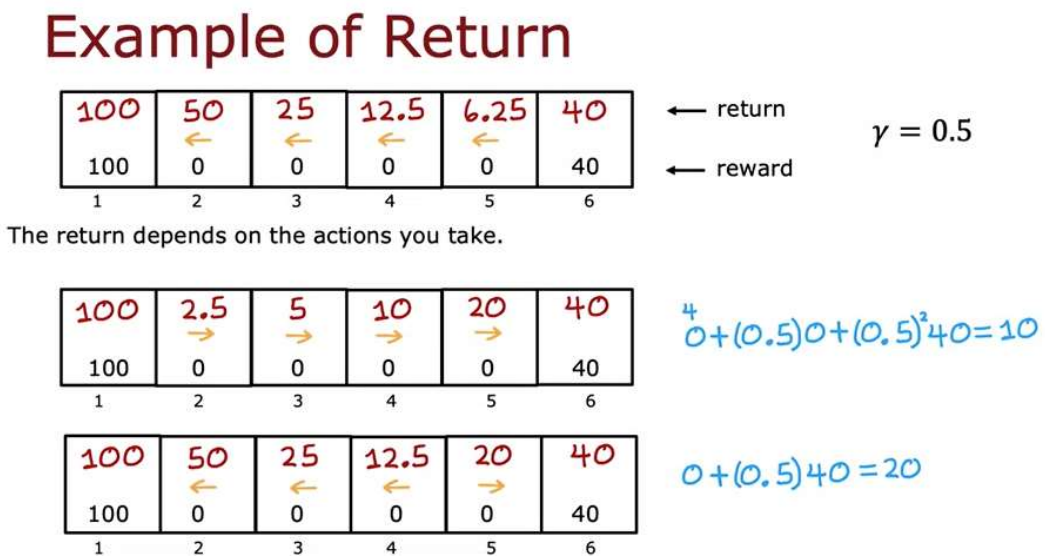
Das Konzept eines Return Captures, bei dem Belohnungen, die Sie schneller erhalten können, möglicherweise attraktiver sind als Belohnungen, bei denen Sie lange brauchen, um sie zu erhalten. Schauen wir uns einmal genau an, wie das funktioniert. Hier ist ein Beispiel für einen Mars-Rover. Wenn Sie ausgehend von Zustand 4 nach links gehen, sehen wir, dass die Belohnungen, die Sie

erhalten, beim ersten Schritt von Zustand 4 Null, von Zustand 3 Null, von Zustand 2 Null und dann 100 bei Zustand 1, dem Endzustand, sind.



Die Rendite ist definiert als die Summe dieser Belohnungen, jedoch gewichtet mit einem zusätzlichen Faktor, dem sogenannten Diskontfaktor. Der Abzinsungsfaktor ist eine Zahl etwas kleiner als 1. Lassen Sie mich 0,9 als Abzinsungsfaktor wählen. Ich werde die Belohnung im ersten Schritt einfach mit Null gewichten, die Belohnung im zweiten Schritt ist ein Rabattfaktor, das 0,9-fache dieser Belohnung, und dann plus den Rabattfaktor^2-fache dieser Belohnung und dann plus den Rabattfaktor ^3-fache Belohnung. Wenn man das ausrechnet, ergibt sich ein Wert von 0,729 mal 100, also 72,9. Die allgemeinere Formel für die Rückkehr lautet: Wenn Ihr Roboter eine Folge von Zuständen durchläuft und beim ersten Schritt die Belohnung R_1, beim zweiten Schritt R_2 und beim dritten Schritt R_3 usw. erhält, beträgt die Rückkehr R_1 plus den Abzinsungsfaktor Gamma, dieses griechische Alphabet Gamma, das ich in diesem Beispiel auf 0,9 gesetzt habe, das Gamma mal R_2 plus Gamma^2 mal R_3 plus Gamma^3 mal R_4 und so weiter, bis Sie zum Endzustand gelangen. Der Abzinsungsfaktor Gamma bewirkt, dass der Reinforcement-Learning-Algorithmus etwas ungeduldig wird. Da die Rendite die volle Gutschrift für die erste Belohnung beträgt, beträgt 100 Prozent 1 mal R_1, aber dann gibt sie der Belohnung etwas weniger Gutschrift, die Sie im zweiten Schritt mit 0,9 multipliziert erhalten, und dann noch weniger Gutschrift für die Belohnung, die Sie erhalten Holen Sie sich beim nächsten Mal Schritt R_3 und so weiter. Wenn Sie also früher Belohnungen erhalten, ergibt sich ein höherer Wert für die Gesamtrendite. In vielen Reinforcement-Learning-Algorithmen ist die übliche Wahl für den Abzinsungsfaktor eine Zahl, die ziemlich nahe bei 1 liegt, etwa 0,9 oder 0,99 oder sogar 0,999. Aber zur Veranschaulichung werde ich im laufenden Beispiel, das ich verwenden werde, tatsächlich einen Abzinsungsfaktor von 0,5 verwenden. Dies führt zu einer sehr starken Reduzierung der Gewichte oder, sagen wir, zu sehr starken Rabatten auf Belohnungen in der Zukunft, da Sie mit jedem zusätzlichen Parsing-Zeitstempel nur halb so viel Gutschrift erhalten wie Belohnungen, die Sie einen Schritt früher erhalten hätten. Wenn Gamma gleich 0,5 wäre, wäre die Rendite im obigen Beispiel 0 plus 0,5 mal 0 gewesen, wobei diese Gleichung oben ersetzt wurde, plus 0,5^2 0 plus 0,5^3 mal 100. Das ist verlorene Belohnung, weil Zustand 1 zum Endzustand wird, und dies ergibt eine Rendite von 12,5. In Finanzanwendungen wird der Abzinsungsfaktor auch ganz natürlich als Zinssatz oder Zeitwert des Geldes interpretiert. Wenn Sie heute einen Dollar haben können, ist dieser vielleicht etwas mehr wert, als wenn Sie in Zukunft nur noch einen Dollar bekommen könnten. Denn selbst einen Dollar können Sie heute auf die Bank legen, etwas Zinsen verdienen und in einem Jahr am Ende etwas mehr Geld haben. Bei Finanzanwendungen gibt dieser

Abzinsungsfaktor oft an, wie viel weniger ein Dollar in der Zukunft sein wird, verglichen mit einem Dollar heute. Schauen wir uns einige konkrete Beispiele für Renditen an.

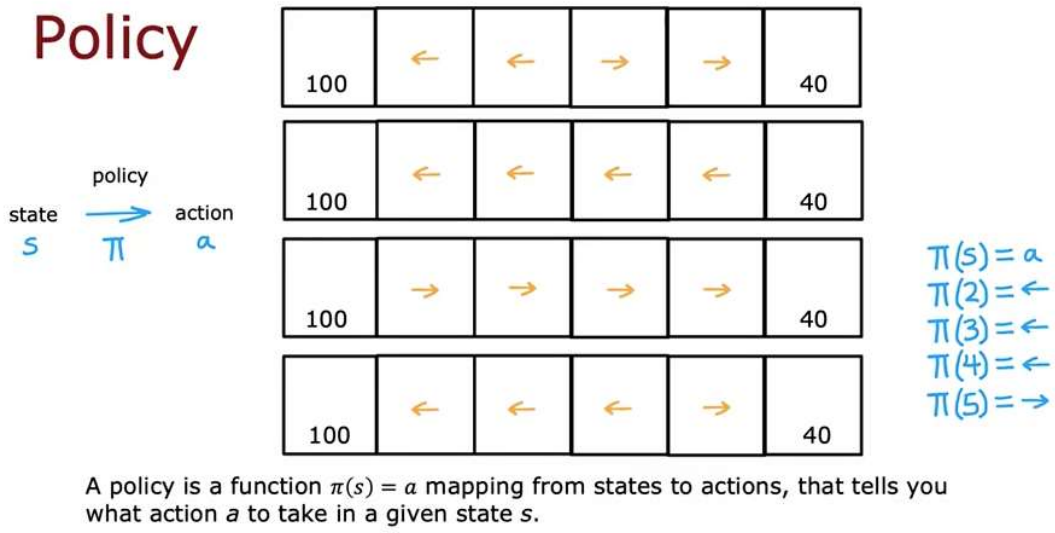


Die Rendite, die Sie erhalten, hängt von den Belohnungen ab, und die Belohnungen hängen von den Maßnahmen ab, die Sie ergreifen, und somit hängt die Rendite von den Maßnahmen ab, die Sie ergreifen. Nehmen wir unser übliches Beispiel und sagen wir für dieses Beispiel: Ich gehe immer nach links. Wir haben bereits zuvor gesehen, dass die Rendite 12,5 beträgt, wenn der Roboter im Zustand 4 startet, wie wir auf der vorherigen Folie berechnet haben. Es stellt sich heraus, dass die Rendite bei einem Start von sagen wir drei 25 betragen würde, weil die 100-Belohnung einen Schritt früher erreicht wird und daher weniger abgezinst wird. Wenn es in Zustand 2 beginnen würde, wäre die Rendite 50. Wenn es einfach beginnen würde und Zustand 1 wäre, erhält es sofort die Belohnung von 100, es wird also nicht niedrig abgezinst. Wenn wir in Zustand 1 beginnen würden, beträgt die Rendite 100, und in diesen beiden Zuständen beträgt die Rendite 6,25. Es stellt sich heraus, dass Sie, wenn Sie in Zustand 6 beginnen, dem Endzustand, nur die Belohnung und damit die Rendite von 40 erhalten. Wenn Sie nun eine andere Reihe von Aktionen ausführen würden, wären die Renditen tatsächlich anders. Wenn wir zum Beispiel immer nach rechts gehen würden, wenn das unsere Aktionen wären, dann würden Sie, wenn Sie in Zustand 4 beginnen würden, eine Belohnung von 0 erhalten. Dann gelangen Sie zu Zustand 5, erhalten eine Belohnung von 0 und das war's gelangt zu Zustand 6 und erhält eine Belohnung von 40. In diesem Fall wäre die Rendite 0 plus 0,5, der Abzinsungsfaktor mal 0 plus 0,5 zum Quadrat mal 40, und das ergibt 0,5 zum Quadrat, also 1/4. also ist 1/4 von 40 10. Die Rendite aus diesem Zustand, aus Zustand 4, ist 10. Wenn Sie Maßnahmen ergreifen, gehen Sie immer nach rechts. Aus ähnlichen Gründen beträgt die Rendite aus diesem Zustand 20, die Rendite aus diesem Zustand beträgt fünf, die Rendite aus diesem Zustand beträgt 2,5 und dann beträgt die Rendite, der bestimmende Zustand, 140. Übrigens, wenn diese Zahlen nicht zutreffen Wenn die Werte völlig sinnvoll sind, können Sie das Video jederzeit anhalten und die Berechnung noch einmal überprüfen, um zu sehen, ob Sie sich selbst davon überzeugen können, dass dies die angemessenen Werte für die Rendite sind. Denn wenn Sie von verschiedenen Zustanden ausgehen und immer nach rechts gehen würden. Wir sehen, dass es immer nach rechts gehen würde. Die erwartete Rendite ist in den meisten Zustanden niedriger. Vielleicht ist es keine so gute Idee, immer nach rechts zu gehen wie immer nach links. Aber es stellt sich heraus, dass wir nicht immer nach links, sondern immer nach rechts gehen müssen. Wir könnten auch entscheiden, wenn Sie sich in Zustand 2 befinden, gehen Sie nach links. Wenn Sie sich in Zustand 3 befinden, gehen Sie nach links. Wenn Sie sich in Zustand 4 befinden, gehen Sie nach links. Aber wenn Sie sich in Zustand 5 befinden, dann sind Sie dieser Belohnung so nah. Lass uns richtig gehen. Dies ist eine andere Art der

Auswahl der zu ergreifenden Maßnahmen, je nachdem, in welchem Zustand Sie sich befinden. Es stellt sich heraus, dass die Rendite, die Sie aus den verschiedenen Zustanden erhalten, 100, 50, 25, 12,5, 20 und 40 beträgt. Nur zur Veranschaulichung ein Fall. Wenn Sie in Zustand 5 beginnen würden, würden Sie hier nach rechts gehen, und so wären die Belohnungen, die Sie erhalten, zuerst Null in Zustand 5 und dann 4. Die Rendite ist Null, die erste Belohnung plus der Rabattfaktor 0,5 mal 40, also 20, weshalb die Rückkehr von diesem Status 20 ist, wenn Sie die hier gezeigten Maßnahmen ergreifen. Zusammenfassend lässt sich sagen, dass die Rendite beim verstärkenden Lernen die Summe der Belohnungen ist, die das System erhält, gewichtet mit dem Abzinsungsfaktor, wobei Belohnungen in ferner Zukunft mit dem Abzinsungsfaktor gewichtet werden, der auf eine höhere Potenz erhöht wird. Dies hat tatsächlich einen interessanten Effekt, wenn es Systeme mit negativen Belohnungen gibt. In dem Beispiel, das wir durchgegangen sind, waren alle Belohnungen null oder positiv. Wenn es jedoch negative Belohnungen gibt, dann gibt der Abzinsungsfaktor tatsächlich einen Anreiz für das System, die negativen Belohnungen so weit wie möglich in die Zukunft zu verschieben. Nehmen wir ein finanzielles Beispiel: Wenn Sie jemandem 10 US-Dollar zahlen müssten, wäre das vielleicht eine negative Belohnung von minus 10. Aber wenn Sie die Zahlung um ein paar Jahre verschieben könnten, wären Sie tatsächlich besser dran, weil Sie in ein paar Jahren 10 US-Dollar zahlen müssten Der Zinssatz ist tatsächlich weniger als 10 $ wert, die Sie heute zahlen mussten. Bei Systemen mit negativen Belohnungen versucht der Algorithmus, die Belohnungen so weit wie möglich in die Zukunft zu verschieben. Für Finanzanwendungen und andere Anwendungen erweist sich das tatsächlich als die richtige Vorgehensweise des Systems. Sie wissen jetzt, was die Rendite beim Reinforcement-Learning ist. Fahren wir mit dem nächsten Video fort, um das Ziel des Reinforcement-Learning-Algorithmus zu formalisieren.

Entscheidungen treffen: Richtlinien für verstärktes Lernen

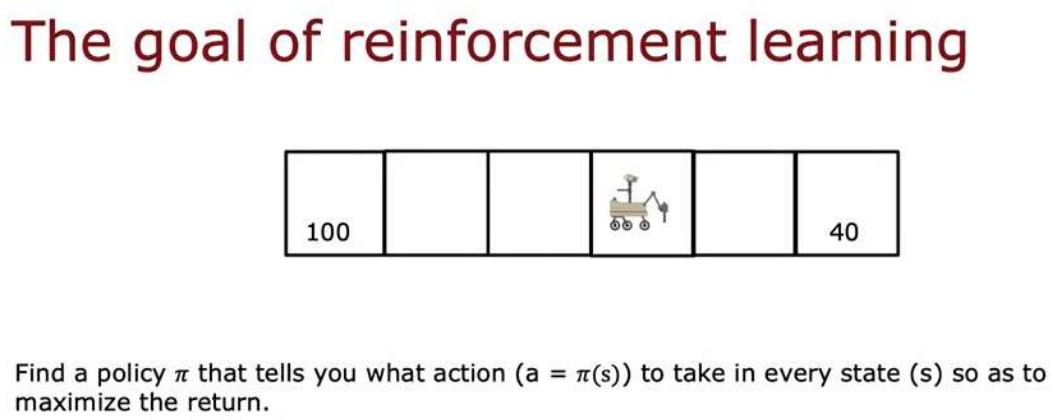
Lassen Sie uns formalisieren, wie ein Reinforcement-Learning-Algorithmus Aktionen auswählt. In diesem Video erfahren Sie, was eine Richtlinie im Reinforcement-Learning-Algorithmus ist. Lass uns einen Blick darauf werfen. Wie wir gesehen haben, gibt es viele verschiedene Möglichkeiten, wie Sie beim Problem des verstärkenden Lernens Maßnahmen ergreifen können. Beispielsweise könnten wir entscheiden, dass wir uns immer für die nähere Belohnung entscheiden.



A policy is a function $\pi(s) = a$ mapping from states to actions, that tells you what action *a* to take in a given state *s*.

Sie gehen also nach links, wenn diese ganz linke Belohnung näher ist, oder nach rechts, wenn diese ganz rechte Belohnung näher ist. Eine andere Möglichkeit, Aktionen auszuwählen, besteht darin, sich immer für die größere Belohnung zu entscheiden, oder wir könnten uns immer für eine kleinere Belohnung entscheiden. Das scheint keine gute Idee zu sein, ist aber eine andere Option, oder Sie

könnten sich dafür entscheiden, nach links zu gehen, es sei denn, Sie sind gerecht einen Schritt von der geringeren Belohnung entfernt. In diesem Fall entscheiden Sie sich für diese. Beim Reinforcement Learning besteht unser Ziel darin, eine Funktion namens Policy Pi zu entwickeln, deren Aufgabe es ist, beliebige Zustände als Eingabe zu nehmen und sie einer Aktion a zuzuordnen, die wir ausführen sollen. Für diese Richtlinie hier unten würde diese Richtlinie beispielsweise sagen, dass, wenn Sie sich in Status 2 befinden, sie uns der linken Aktion zuordnet.

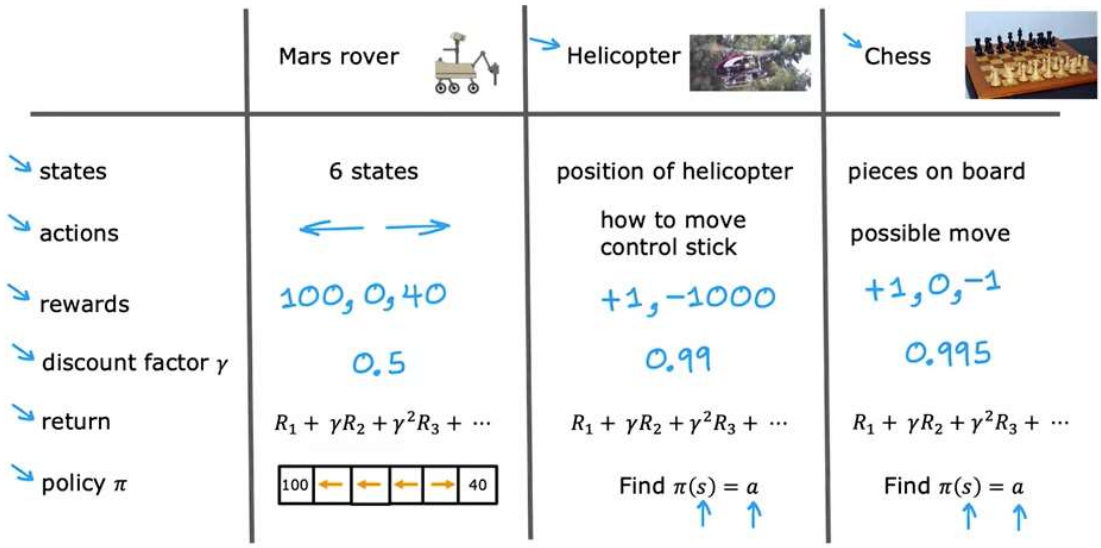


Wenn Sie sich in Status 3 befinden, lautet die Richtlinie: Gehen Sie nach links. Wenn Sie sich in Zustand 4 befinden, gehen Sie ebenfalls nach links und wenn Sie sich in Zustand 5 befinden, gehen Sie nach rechts. Pi, angewendet auf den Zustand S, sagt uns, welche Maßnahmen wir in diesem Zustand ergreifen sollen. Das Ziel des verstärkenden Lernens besteht darin, einen Richtlinien-Pi oder Pi von S zu finden, der Ihnen sagt, welche Maßnahmen Sie in jedem Zustand ergreifen müssen, um die Rendite zu maximieren. Ich weiß übrigens nicht, ob „Policy“ der beschreibendste Begriff für Pi ist, aber es ist einer dieser Begriffe, die beim Reinforcement Learning zum Standard geworden sind. Vielleicht wäre es eine natürlichere Terminologie, Pi als Controller und nicht als Richtlinie zu bezeichnen, aber jeder, der sich mit Reinforcement Learning beschäftigt, nennt dies jetzt Politik. Im letzten Video haben wir einige Konzepte des verstärkenden Lernens von Zustanden über Aktionen zur Belohnung bis hin zu Renditen und Richtlinien durchgegangen. Lassen Sie uns im nächsten Video einen kurzen Überblick darüber geben und dann mit der Entwicklung von Algorithmen zum Auffinden dieser Richtlinien beginnen. Kommen wir zum nächsten Video.

Überprüfung der Schlüsselkonzepte

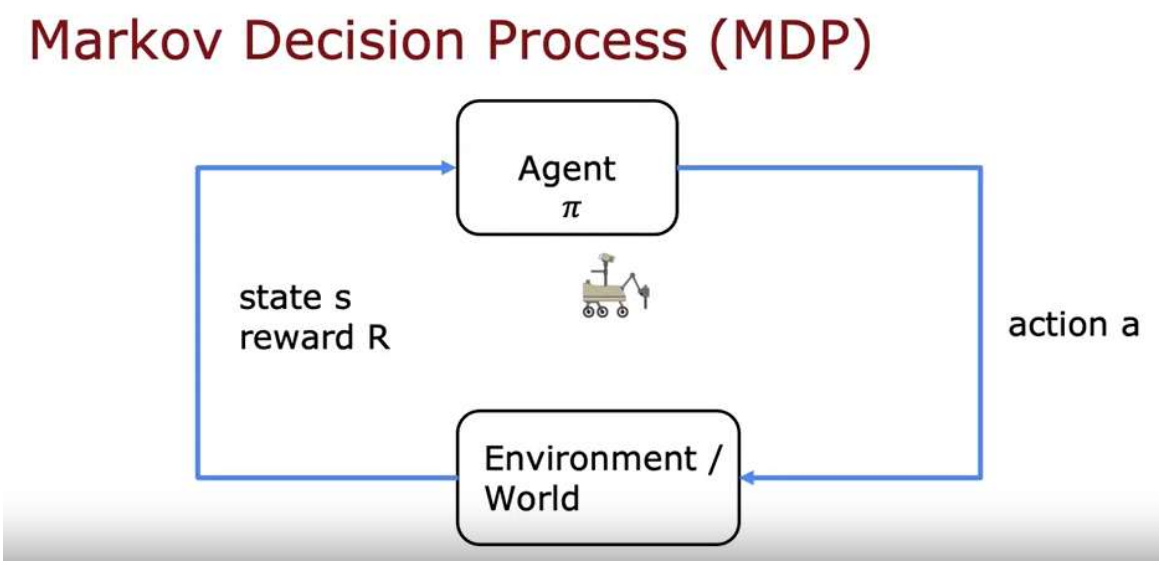
Wir haben einen Reinforcement-Learning-Formalismus am Beispiel des Mars-Rovers mit sechs Zuständen entwickelt. Lassen Sie uns einen kurzen Überblick über die Schlüsselkonzepte geben und sehen, wie diese Konzepte auch für andere Anwendungen verwendet werden können. Einige der Konzepte, die wir besprochen haben, sind Zustände eines Reinforcement-Learning-Problems, die Reihe von Aktionen, die Belohnungen, ein Abzinsungsfaktor, dann die Art und Weise, wie Belohnungen und der Abzinsungsfaktor insgesamt zur Berechnung der Rendite verwendet werden, und schließlich eine Richtlinie, deren Aufgabe Es soll Ihnen dabei helfen, Maßnahmen auszuwählen, um die Rendite zu maximieren. Für das Beispiel des Mars-Rovers hatten wir sechs Zustände, die wir mit 1–6 nummerierten, und die Aktionen bestanden darin, nach links oder nach rechts zu gehen. Die Belohnungen betrugen 100 für den Zustand ganz links, 40 für den Zustand ganz rechts und null dazwischen, und ich habe einen Abzinsungsfaktor von 0,5 verwendet. Die Rendite wurde durch diese Formel angegeben und wir könnten unterschiedliche Richtlinien haben, um Aktionen abhängig davon

darzustellen, in welchem Zustand Sie sich befinden. Derselbe Formalismus oder Zustände, Aktionen, Belohnungen usw. können auch für viele andere Anwendungen verwendet werden. Nehmen Sie das Problem an oder finden Sie einen autonomen Hubschrauber. Ein Zustand festzulegen wäre die Menge möglicher Positionen, Ausrichtungen, Geschwindigkeiten usw. des Hubschraubers. Die möglichen Aktionen wären die Reihe möglicher Möglichkeiten, den Steuerknüppel eines Hubschraubers zu bewegen, und die Belohnung könnte ein Plus von eins sein, wenn er gut fliegt, und ein Minus von 1.000, wenn er nicht wirklich schwer fällt oder abstürzt. Belohnungsfunktion, die Ihnen sagt, wie gut der Helikopter fliegt. Der Abzinsungsfaktor, eine Zahl etwas kleiner als eins, sagen wir mal 0,99, und dann berechnen Sie basierend auf den Prämien und dem Abzinsungsfaktor die Rendite nach derselben Formel. Die Aufgabe eines Reinforcement-Learning-Algorithmus besteht darin, einige Richtlinien-Pi von s zu finden, sodass Sie anhand der als Eingabe eingegebenen Position der Hubschrauber-s erkennen, welche Maßnahmen Sie ergreifen müssen. Das heißt, es erklärt Ihnen, wie Sie die Steuerknüppel bewegen. Hier ist noch ein Beispiel. Hier ist ein Spiel. Angenommen, Sie möchten Reinforcement Learning nutzen, um das Schachspielen zu erlernen. Der Zustand dieses Problems wäre die Position aller Figuren auf dem Brett. Übrigens, wenn Sie Schach spielen und die Regeln gut kennen, weiß ich, dass das etwas mehr Informationen sind als nur die Position der Figuren, die für Schach wichtig sind, aber ich werde es für dieses Video ein wenig vereinfachen. Die Aktionen sind die möglichen legalen Züge im Spiel, und dann wäre eine übliche Wahl der Belohnung, wenn Sie Ihrem System eine Belohnung von plus eins geben, wenn es ein Spiel gewinnt, minus eins, wenn es das Spiel verliert, und eine Belohnung von null, wenn es bindet ein Spiel. Für Schach wird normalerweise ein Abzinsungsfaktor verwendet, der sehr nahe bei eins liegt, also vielleicht 0,99 oder sogar 0,995 oder 0,999, und die Rendite verwendet dieselbe Formel wie die anderen Anwendungen. Auch hier wird dem Ziel eine Vorstandsposition gegeben, um mithilfe eines Richtlinien-Pi eine gute Aktion auszuwählen. Dieser Formalismus einer Reinforcement-Learning-Anwendung hat tatsächlich einen Namen.



Man nennt es einen Markov-Entscheidungsprozess, und ich weiß, dass das nach einem großen, technisch komplizierten Begriff klingt. Aber wenn Sie jemals den Begriff Markov-Entscheidungsprozess oder kurz MDP hören, dann ist das nur der Formalismus, über den wir in den letzten Videos gesprochen haben. Der Begriff Markov im MDP- oder Markov-Entscheidungsprozess bezieht sich darauf, dass die Zukunft nur vom aktuellen Zustand abhängt und nicht von irgendetwas, das vor dem Erreichen des aktuellen Zustands geschehen sein könnte. Mit anderen Worten: In einem Markov-Entscheidungsprozess hängt die Zukunft nur davon ab, wo Sie sich jetzt befinden, und nicht davon, wie Sie hierher gekommen sind. Eine andere Möglichkeit, sich den Markov-

Entscheidungsprozessformalismus vorzustellen, besteht darin, dass wir einen Roboter oder einen anderen Agenten haben, den wir kontrollieren möchten, und dass wir Aktionen auswählen können und auf der Grundlage dieser Aktionen etwas in der Welt oder in der Welt passieren wird die Umwelt, wie zum Beispiel unsere Position in der Welt, ändert sich oder wir können ein Stück Gestein entnehmen und die wissenschaftliche Mission ausführen. Die Art und Weise, wie wir die Aktion a auswählen, erfolgt mit einem Richtlinien-Pi.

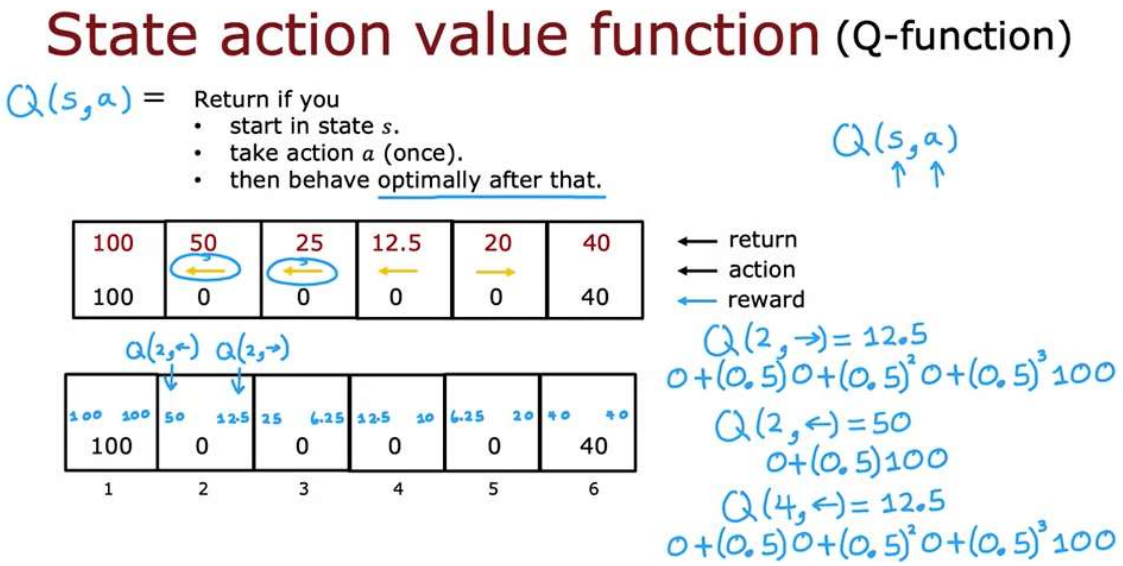


Basierend auf dem, was in der Welt passiert, können wir dann sehen oder beobachten, in welchem Zustand wir uns befinden und welche Belohnungen wir erhalten. Manchmal sehen Sie, dass verschiedene Autoren ein Diagramm wie dieses verwenden, um den Markov-Entscheidungsprozess oder den MDP-Formalismus darzustellen. Dies ist jedoch nur eine weitere Möglichkeit, die Konzepte zu veranschaulichen, die Sie in den letzten Videos kennengelernt haben. Sie wissen jetzt, wie ein Reinforcement-Learning-Problem funktioniert. Im nächsten Video beginnen wir mit der Entwicklung eines Algorithmus zur Auswahl guter Aktionen. Der erste Schritt dorthin besteht darin, die Zustandsaktionswertfunktion zu definieren und schließlich zu lernen, sie zu berechnen. Dies erweist sich als eine der Schlüsselgrößen für die Entwicklung eines Lernalgorithmus. Gehen wir zum nächsten Video, um zu sehen, was das ist, die Funktion „State Action Value“.

Definition der State action value Funktion

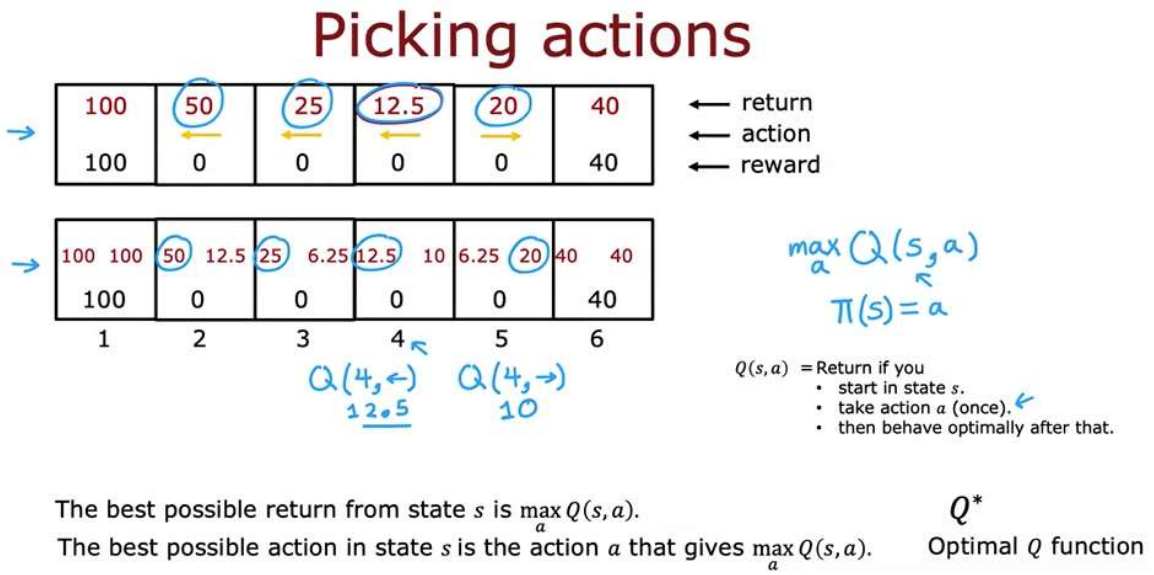
Wenn wir später in dieser Woche mit der Entwicklung von Reinforcement-Learning-Stunden beginnen, sehen Sie, dass es eine Schlüsselgröße gibt, die Reinforcement-Learning-Pfeile zu berechnen versuchen, und zwar die Zustandsaktionswertfunktion. Schauen wir uns einmal an, was diese Funktion ist. Die Zustandsaktionswertfunktion ist eine Funktion, die normalerweise mit dem Großbuchstaben Q bezeichnet wird. Sie ist eine Funktion eines Zustands, in dem Sie sich möglicherweise befinden, sowie der Aktion, die Sie in diesem Zustand und in QFSA ausführen möchten. Gibt eine Zahl an, die der Rendite entspricht. Wenn Sie in diesem Zustand beginnen. S und Aktion A nur einmal ausführen und nach einmaligem Ausführen von Aktion A verhält man sich danach optimal. Anschließend ergreifen Sie alle Maßnahmen, die zu einer möglichst hohen Rendite führen. Jetzt denken Sie vielleicht, dass diese Definition etwas seltsam ist, denn woher wissen wir, was das optimale Verhalten ist? Und wenn wir wüssten, wie sich das Auto verhält, wenn wir bereits wüssten, welche Maßnahmen in jedem Zustand am besten zu ergreifen sind, warum müssen wir dann immer noch Q von SA berechnen? Weil wir bereits die Auto-Richtlinie haben. Daher möchte ich zugeben, dass diese Definition etwas seltsam ist. Diese Definition hat fast etwas Zirkuläres, aber

seien Sie versichert, wenn wir uns später bestimmte Ergebnisse des verstärkenden Lernens ansehen, werden wir diese leicht zirkuläre Definition auflösen und eine Möglichkeit finden, die Q-Funktion zu berechnen, noch bevor wir sie gefunden haben optimale Politik.



Aber das seht ihr in einem späteren Video. Machen Sie sich deshalb vorerst keine Sorgen. Schauen wir uns ein Beispiel an, das wir zuvor gesehen haben, dass dies eine ziemlich gute Richtlinie ist. Gehen Sie von Stufe 2, 3 und vier nach links und von Stufe fünf nach rechts. Es stellt sich heraus, dass dies tatsächlich die optimale Richtlinie für die Mars-Rover-Anwendung ist. Wenn der Abzinsungsfaktor Gamma 0,5 beträgt, ist Q von S. A gleich der Gesamtrendite. Wenn Sie beispielsweise mit Aktion A beginnen und sich dann optimal verhalten danach. Das heißt, Maßnahmen gemäß dieser Richtlinie ergreifen. Wie hier gezeigt, wollen wir herausfinden, welches Q von s,a ist. Ist für ein paar verschiedene Zustanden. Schauen wir uns auch Q des Zustandes an. Und was wäre, wenn wir die Maßnahme ergreifen, um nach rechts zu gehen? Wenn Sie sich in Zustand zwei befinden und nach rechts gehen, landen Sie bei Zustand drei. Und nachdem Sie sich optimal verhalten, gehen Sie von ST drei nach links und dann nach links Von Zustand zu Zustand und dann erhalten Sie schließlich die Belohnung von 100. In diesem Fall wären die Belohnungen, die Sie erhalten, Null von Zustand zu Null, wenn Sie bei drei bleiben, Null, wenn Sie zu Zustand zwei zurückkehren, und dann 100, wenn Sie schließlich bei Zustand zwei ankommen Der Endzustand ist eins, und daher ist die Rendite Null plus 0,5 mal das plus 0,5 zum Quadrat mal ac plus 0,5 kubisch mal 100. Und das ergibt 12,5. Und so ist Q von ST zwei nach rechts gleich 12,5. Beachten Sie, dass dies kein Urteil darüber ist, ob es eine gute Idee ist, richtig vorzugehen oder nicht. Es ist eigentlich keine so gute Idee von Zustand, richtig zu gehen, aber es meldet die Rendite einfach getreu, wenn man Aktion A durchführt und sich danach optimal verhält. Hier ist ein weiteres Beispiel. Wenn Sie sich im Status 2 befinden und nach links gehen, ist die Reihenfolge der Belohnungen, die Sie erhalten, Null, wenn Sie sich im Status 2 befinden, gefolgt von 100. Die Rendite ist also Null plus 0,5 mal 100, was 50 Zoll entspricht um die Werte von QSA aufzuschreiben. In diesem Diagramm schreibe ich hier rechts 12,5, um anzuzeigen, dass dies das Q des Zustands ist, der nach rechts geht. Und wenn ich dann hier links eine kleine 50 schreibe, um anzuzeigen, dass dies das Q von ST zwei ist, und nach links gehe, nur um ein weiteres Beispiel zu nennen. Was wäre, wenn wir in ST vier wären und uns dazu entschließen würden, nach links zu gehen? Nun, wenn Sie sich in Stufe vier befinden, gehen Sie nach links, Sie erhalten eine Belohnung von Null und dann werden Sie hier links aktiv. Also null Gewinn, nimm die Aktion links hier, null und dann 100. Q von vier links führt also zu einer Belohnung von null, weil die erste Aktion übrig bleibt und dann, weil wir danach die optimale Richtlinie befolgt haben. Du kannst 00 100 belohnen. Und so ist die Rendite null plus das 00,5-fache davon. Plus 4,5 zum Quadrat mal

das plus 0,5 Q mal das. Das entspricht also 12,5. Das verbleibende Q4 beträgt also 12,5. Ich werde dieses Jahr als 12.5 schreiben. Und es stellt sich heraus, dass Sie, wenn Sie diese Übung für alle anderen Zustände und alle anderen Aktionen durchführen würden, am Ende das Q von s,a erhalten. Für verschiedene Zustände und unterschiedliche Aktionen. Und schließlich beim Terminal State. Nun, es spielt keine Rolle, was Sie tun, Sie erhalten einfach die Endprämie von 100 oder 40. Schreiben Sie also einfach diese Endprämien hier auf. Das ist also Q von s,a. Für jeden Zustand Zustand eins bis sechs und für die beiden Aktionen Aktion links und Aktion rechts. Denn die Zustandsaktionswertfunktion wird fast immer mit dem Buchstaben Q bezeichnet. Dieser wird oft auch als Q-Funktion bezeichnet. Die Begriffe Q. Funktion und Zustandsaktionswertfunktion werden also synonym verwendet und sagen Ihnen, wie hoch Ihre Rendite ist oder welchen Wert sie tatsächlich hat? Wie gut ist es? Führen Sie einfach Aktion A und ST S aus und verhalten Sie sich danach optimal. Nun stellt sich heraus, dass Sie, sobald Sie die Q-Funktion berechnen können, auch die Möglichkeit haben, Aktionen auszuwählen. Hier finden Sie die Richtlinien und die Rückgabe. Und hier sind die Werte zwei von s,a. Von der vorherigen Folie.

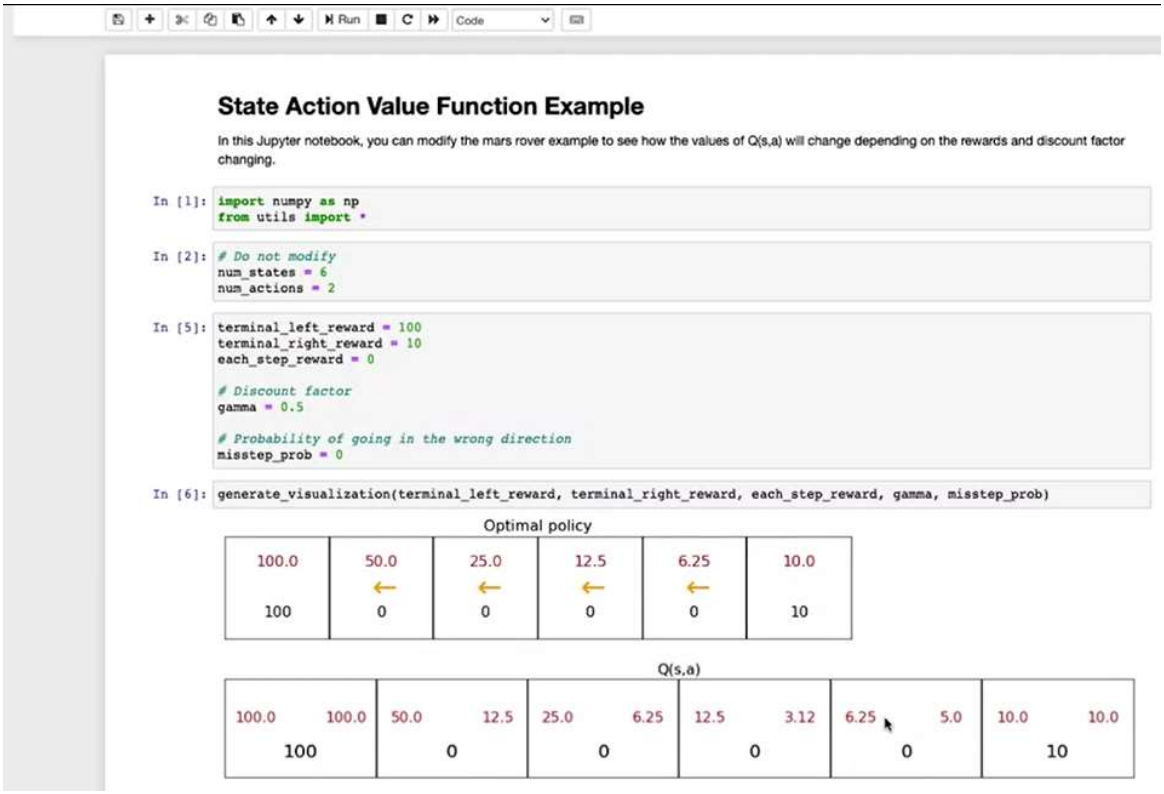


Wenn Sie sich die verschiedenen Zustände ansehen, fällt Ihnen eine interessante Sache auf: Wenn Sie den Zustand auf die Ausführung der verbleibenden Aktion umstellen, ergibt sich ein, q. Wert oder Zustandsaktionswert von 50, was tatsächlich die bestmögliche Rendite ist, die Sie aus diesem Zustand erzielen können. Im Zustand drei zwei von s,a. Denn die verbleibende Aktion bringt Ihnen auch die höhere Rendite, daher bringt Ihnen die verbleibende Aktion die gewünschte Rendite. Und in Zustand fünf gibt es tatsächlich die Aktion nach rechts, die Ihnen die höhere Rendite von 20 gibt. Es stellt sich also heraus, dass die bestmögliche Rendite aus jedem Zustand S. der größte Wert von Q, F, S ist. A maximiert über A. Um sicherzugehen, dass das klar ist, sage ich, dass in sagen wir mal „Stay for“ noch zwei von Zustand vier übrig sind, also 12,5 und q. von Zustand vier richtig, was sich als 10 herausstellt. Und der größere dieser beiden Werte, nämlich 12,5, ist die bestmögliche Rendite aus diesem Zustand vier. Mit anderen Worten: Die höchste Rendite, die Sie von Zustand vier erwarten können, beträgt 12,5. Und es ist tatsächlich die größere dieser beiden Zahlen 12,5 und 10. Und wenn Sie außerdem möchten, dass Ihr Mars Rover eine Rendite von 12,5 statt beispielsweise 10 erzielt, dann sollten Sie die Maßnahme A ergreifen. Dadurch erhalten Sie den größeren Wert von Q von s,a. Der bestmögliche Aktionsstatus ist also die Aktion A. Das maximiert tatsächlich Q von s,a. Dies könnte Ihnen also einen Hinweis darauf geben, warum Q, of s,a berechnet wird. Ist ein wichtiger Teil des Reinforcement-Learning-Algorithmus, der später erstellt wird. Nämlich, wenn Sie eine Möglichkeit haben, Q von s,a zu berechnen. Für jeden Zustand und für jede Aktion müssen Sie sich

dann, wenn Sie sich in einem bestimmten Zustand befinden, nur die verschiedenen Aktionen A ansehen. Und die Aktion A auswählen. Das maximiert Q von s,a. Und so kann pi F. S einfach die Aktion A auswählen. Das ergibt den größten Wert von Q von s,a. Und das wird eine gute Aktion sein. Tatsächlich erwies es sich als die optimale Aktion. Eine weitere Intuition, warum dies sinnvoll ist, ist Qof s,a. Wird zurückgegeben, wenn Sie plötzlich Status haben und die Aktion A ausführen. Und sich danach optimal verhalten. Um also die größtmögliche Rendite zu erzielen, sollten Sie unbedingt die Maßnahme A ergreifen. Das führt zur größtmöglichen Gesamtrendite. Das ist der Grund, warum wir nur eine Möglichkeit hätten, Q f s,a zu berechnen. Für jeden Zustand scheint es die beste Maßnahme zu sein, unter diesen Umständen Hilfe zu ergreifen, die den Ertrag maximiert. Obwohl dies nichts ist, was Sie dafür wissen müssen. Denn ich möchte auch erwähnen, dass man, wenn man online nachschaut oder sich die Literatur zum Reinforcement Learning anschaut, diese Q-Funktion manchmal auch als Q. Star statt als Q geschrieben sieht. Und diese Q-Funktion wird manchmal auch die optimale Q-Funktion genannt. Diese Begriffe beziehen sich lediglich auf die Q-Funktion genauso, wie wir sie definiert haben. Wenn Sie sich also die Literatur zum Reinforcement Learning ansehen und etwas über Q. Star oder die Q- Funktion lesen, ist damit lediglich die Zustandsaktionswertfunktion gemeint, über die wir gesprochen haben. Aber für die Zwecke dieses Kurses brauchen Sie sich darüber keine Sorgen zu machen. Zusammenfassend lässt sich also sagen, ob Sie Q von s,a berechnen können. Das gibt uns für jeden Zustand und jede Aktion eine gute Möglichkeit, den Auto-Policy-Pi von S zu berechnen. Das ist also die Zustandsaktionswertfunktion oder die Q-Funktion. Wir werden später darüber sprechen, wie man trotz des leicht kreisförmigen Aspekts der Definition der Q-Funktion einen Algorithmus zur Berechnung dieser Werte entwickelt. Aber werfen wir zunächst einen Blick auf das nächste Video mit einigen konkreten Beispielen für die Werte Q von s,a. Eigentlich so aussehen

Beispiel für eine Zustandsaktionswertfunktion

Verwenden des Beispiels für die Funktion „State-Action-Value“. Sie sehen, wie die Werte von QSA aussehen. Um unsere Intuition über Reinforcement-Learning-Probleme und die Art und Weise, wie sich die Werte von QSA je nach Problem ändern, aufrechtzuerhalten, wird ein optionales Labor bereitgestellt. Auf diese Weise können Sie herumspielen, das Beispiel [UNVERSTÄNDLICH] modifizieren und selbst sehen, wie sich QSA ändern wird. Lass uns einen Blick darauf werfen. Hier ist ein Jupyter- Notizbuch, mit dem Sie hoffentlich spielen, nachdem Sie sich dieses Video angesehen haben. Ich werde diese Hilfsfunktionen ausführen.



State Action Value Function Example

In this Jupyter notebook, you can modify the mars rover example to see how the values of Q(s,a) will change depending on the rewards and discount factor changing.

```
In [1]: import numpy as np
        from utils import *

In [2]: # Do not modify
        num_states = 6
        num_actions = 2

In [5]: terminal_left_reward = 100
        terminal_right_reward = 10
        each_step_reward = 0

        # Discount factor
        gamma = 0.5

        # Probability of going in the wrong direction
        misstep_prob = 0

In [6]: generate_visualization(terminal_left_reward, terminal_right_reward, each_step_reward, gamma, misstep_prob)
```

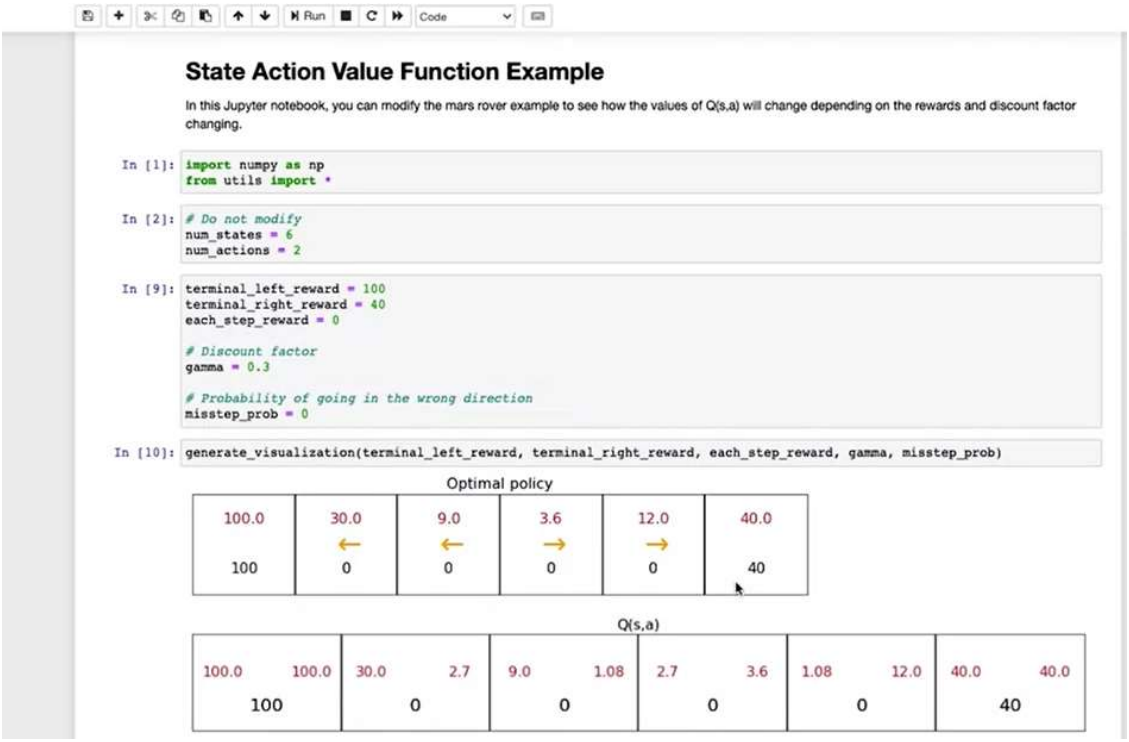
Optimal policy

100.0	50.0	25.0	12.5	6.25	10.0
100	← 0	← 0	← 0	← 0	10

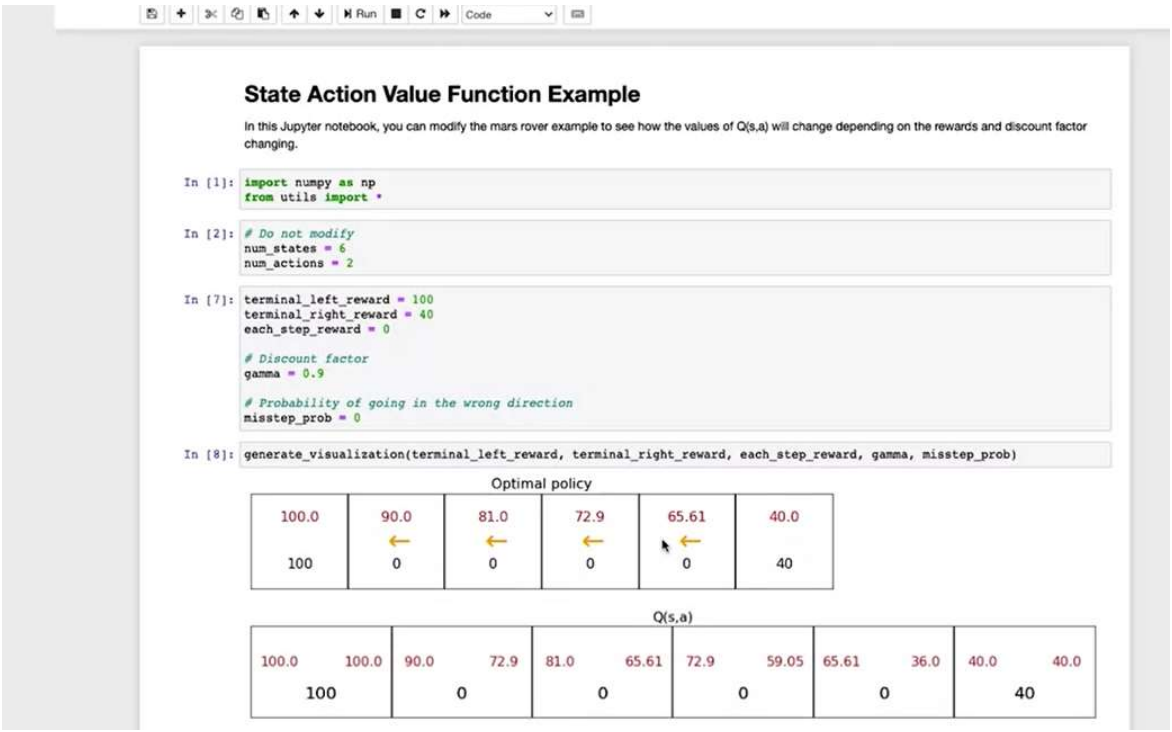
Q(s,a)

100.0 100.0	50.0 12.5	25.0 6.25	12.5 3.12	6.25 5.0	10.0 10.0
100	0	0	0	0	10

Beachten Sie hier, dass dies die Nummer sechs der beiden Aktionen angibt, sodass ich diese nicht ändern würde. Und dies gibt die Belohnungen des Terminals links im Terminal rechts an, die 140 waren, und dann waren die Belohnungen der Zwischenzustände Null. Der Abzinsungsfaktor beträgt 0,5. Und lassen Sie uns die Wahrscheinlichkeit eines Fehltritts vorerst außer Acht lassen. Darüber sprechen wir in einem späteren Video. Wenn Sie diesen Code mit diesen Werten ausführen, werden die optimale Richtlinie sowie die Q-Funktion Q von SA berechnet und visualisiert. Später erfahren Sie, wie Sie einen Lernalgorithmus entwickeln, um Q of SA selbst zu schätzen oder zu berechnen. Machen Sie sich also vorerst keine Gedanken darüber, welchen Code wir zur Berechnung von Q of SA geschrieben haben. Aber Sie sehen, dass die Werte Q von SA hier die Werte sind, die wir in der Vorlesung gesehen haben. Jetzt beginnt der Spaß. Lassen Sie uns einige Werte ändern und sehen, wie sich diese Dinge ändern. Ich werde die Terminal-Rechtsbelohnung auf einen viel kleineren Wert aktualisieren, also nur 10. Wenn ich jetzt den Code erneut ausführe, schaue mir an, wie sich Q von SA ändert und denke jetzt, dass du dich in Zustand 5 befindest. Wenn du dann nach links gehst und sich optimal verhalten bekommst du 6,25.



Wenn Sie dagegen richtig vorgehen und sich auch danach verhalten, erhalten Sie eine Rendite von nur fünf. Wenn also die Belohnung auf der rechten Seite so gering ist, beträgt sie nur 10. Selbst wenn Sie so nah bei Ihnen sind, gehen Sie lieber ganz nach links. Und tatsächlich sieht die Autopolitik nun vor, in jedem einzelnen Bundesstaat nach links zu fahren. Nehmen wir noch einige weitere Änderungen vor. Ich werde die endgültige rechte Belohnung wieder auf 40 ändern. Aber lassen Sie mich den Rabattfaktor auf 0,9 ändern, mit einem Rabattfaktor, der näher bei eins liegt. Dies macht den Marsrover weniger ungeduldig und ist bereit, länger zu warten, um auf eine höhere Belohnung zu warten, da Belohnungen in der Zukunft nicht mit 0,5 für eine hohe Leistung multipliziert werden, sondern mit 0,9 für eine hohe Leistung. Und das gilt auch für die Bereitschaft, mehr Patienten zu haben, da die Belohnungen in Zukunft nicht mit einer so kleinen Zahl abgezinst oder multipliziert werden wie damals, als der Rabatt 0,5 betrug. Lassen Sie uns den Code also erneut ausführen. Und jetzt sehen Sie, dass dies das Q von SA für die verschiedenen Zustanden ist, und wenn Sie nun für Zustand 5 nach links gehen, erhalten Sie tatsächlich eine höhere Belohnung von 65,61 im Vergleich zu 36. Beachten Sie übrigens, dass 36 das 0,9-fache dieser Endbelohnung von 40 ist. Also diese Zahlen Sinn ergeben.

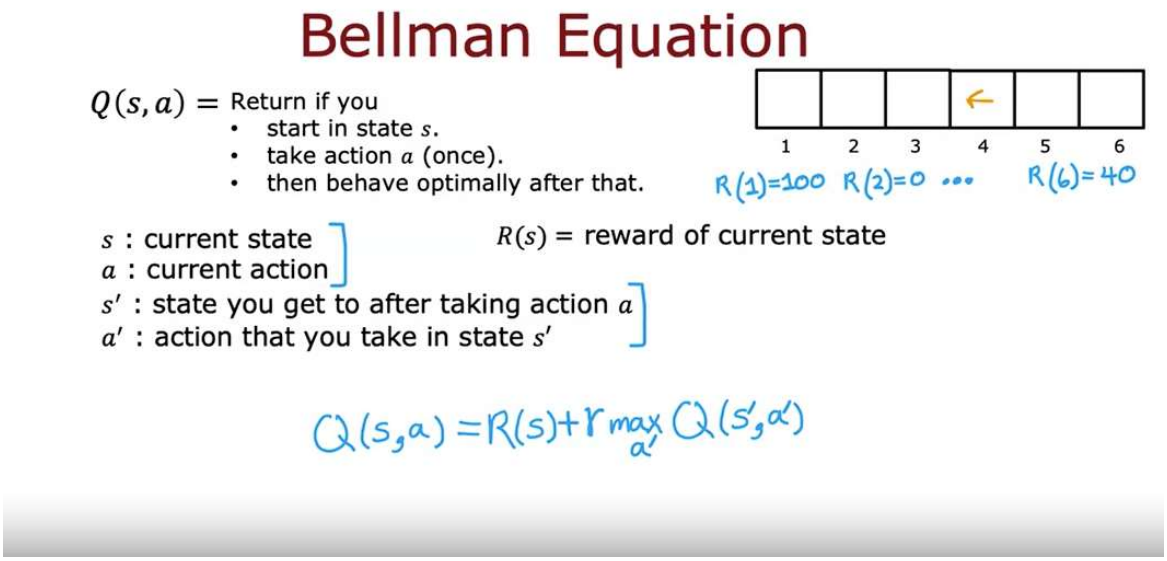


Aber wenn ein kleiner Patient bereit ist, nach links zu gehen, selbst wenn Sie sich in Zustand 5 befinden, ändern wir nun Gamma auf eine viel kleinere Zahl wie 0,3. Dadurch werden Prämien in der Zukunft sehr stark reduziert. Das macht ihn unglaublich ungeduldig. Lassen Sie mich diesen Code noch einmal ausführen, und jetzt hat sich das Verhalten geändert. Ich habe festgestellt, dass ich jetzt in Zustand 4 nicht die Geduld haben werde, mich für die größere Belohnung von 100 zu entscheiden, da der Abzinsungsfaktor Gamma jetzt so klein ist, dass er 0,3 beträgt. Wir würden uns lieber für die Belohnung von 40 entscheiden, auch wenn diese viel kleiner ist. Die Belohnung liegt näher und ist die Frage, ob wir uns dafür entscheiden. Ich hoffe also, dass Sie ein Gefühl dafür bekommen, indem Sie selbst mit diesen Zahlen herumspielen und diesen Code ausführen. Wie sich die Werte von Q von SA ändern und wie die optimale Rendite, die Sie bemerken, die größere dieser beiden Zahlen QSA ist. Wie sich das ändert und wie sich auch die optimale Richtlinie ändert. Ich hoffe also, dass Sie mit dem optionalen Labor herumexperimentieren, die Belohnungsfunktion und den Gamma-Rabattfaktor ändern und andere Werte ausprobieren. Und sehen Sie selbst, wie sich die Werte von Q of SA ändern, wie sich die optimale Rendite aus verschiedenen Zustanden ändert und wie sich die Auto-Richtlinie abhängig von diesen unterschiedlichen Werten ändert. Ich hoffe, dass Sie dadurch Ihr Gespür dafür schärfen, wie sich diese unterschiedlichen Größen abhängig von den Belohnungen usw. bei der Anwendung des verstärkenden Lernens auswirken. Nachdem Sie im Labor gespielt haben, sind wir bereit, zurückzukommen und über die wahrscheinlich wichtigste Gleichung beim Reinforcement Learning zu sprechen, die sogenannte Bellman-Gleichung. Ich wünsche Ihnen also viel Spaß beim Spielen mit dem optionalen Labor und lasst uns danach wieder über Bellman-Gleichungen sprechen.

Bellman-Gleichung

Lassen Sie mich zusammenfassen, wo wir stehen. Wenn Sie die Zustandsaktionswertfunktion Q von S, A berechnen können, haben Sie die Möglichkeit, aus jeder Szene eine gute Aktion auszuwählen. Wählen Sie einfach die Aktion A aus, die Ihnen den größten Wert von Q von S,A ergibt. Die Frage ist, wie berechnet man diese Werte Q von S,A ? Beim Reinforcement Learning gibt es eine Schlüsselgleichung namens Bellman-Gleichung, die uns bei der Berechnung der Zustandsaktionswertfunktion hilft. Werfen wir einen Blick auf die Gleichung. Zur Erinnerung: Dies ist

die Definition von Q von S, A. Wenn wir im Zustand S beginnen, führen wir die Aktion einmal durch und sie verhalten sich danach optimal. Um die Bellman-Gleichung zu beschreiben, werde ich die folgende Notation verwenden. Ich werde S verwenden, um den aktuellen Zustand zu kennzeichnen. Als nächstes werde ich R von S verwenden, um die Belohnungen des aktuellen Zustands zu bezeichnen.

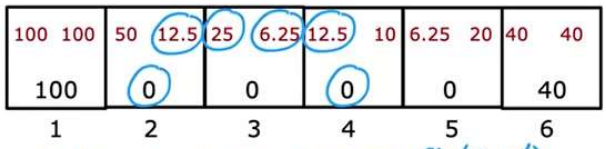


Für unser kleines MDP-Beispiel haben wir, dass r eines Zustands 1 100 ist. Die Belohnung von Zustand 2 ist 0 und so weiter. Die Belohnung für Zustand 6 beträgt 40. Ich werde das Alphabet A verwenden, um die aktuelle Aktion zu bezeichnen, die Aktion, die Sie im Zustand S ausführen. Nachdem Sie die Aktion a ausgeführt haben, gelangen Sie in einen neuen Zustand. Wenn Sie sich beispielsweise in Zustand 4 befinden und die verbleibende Aktion ausführen, gelangen Sie zu Zustand 3. Ich verwende S prim, um den Zustand zu bezeichnen, in den Sie gelangen, nachdem Sie diese Aktion a vom aktuellen Zustand S ausgeführt haben. Ich werde auch A prim verwenden, um die Aktion zu bezeichnen, die Sie im Zustand S prim ausführen könnten, der neuen Steady, die Sie erreichen müssen. Die Notationskonvention besagt übrigens, dass S,A dem aktuellen Zustand und der aktuellen Aktion entsprechen. Wenn wir die Primzahl hinzufügen, ist das der nächste Zustand und dann die nächste Aktion. Die Bellman-Gleichung lautet wie folgt. Darin heißt es, dass Q von S,A, also die Rendite unter diesen Annahmen, die gleich r von S ist, die Belohnung angibt, die man dafür erhält, dass man sich in diesem Zustand befindet, zuzüglich des Abzinsungsfaktors Gamma mal max. über alle möglichen Aktionen, eine Primzahl von q von S prim, dem neuen Zustand, den Sie gerade erreicht haben, und dann einer Primzahl. In dieser Gleichung ist eine Menge los. Schauen wir uns zunächst einige Beispiele an. Wir werden noch einmal darauf zurückkommen, um zu sehen, warum diese Gleichung sinnvoll sein könnte. Schauen wir uns ein Beispiel an. Schauen wir uns Q of State 2 und die Aktion an. Wenden Sie die Bellman-Gleichung darauf an, um zu sehen, welchen Wert sie uns gibt.

Bellman Equation

$$Q(s,a) = R(s) + \gamma \max_{a'} Q(s',a')$$

$Q(s,a) = R(s)$



$s = 2$
$a = \rightarrow$
$s' = 3$

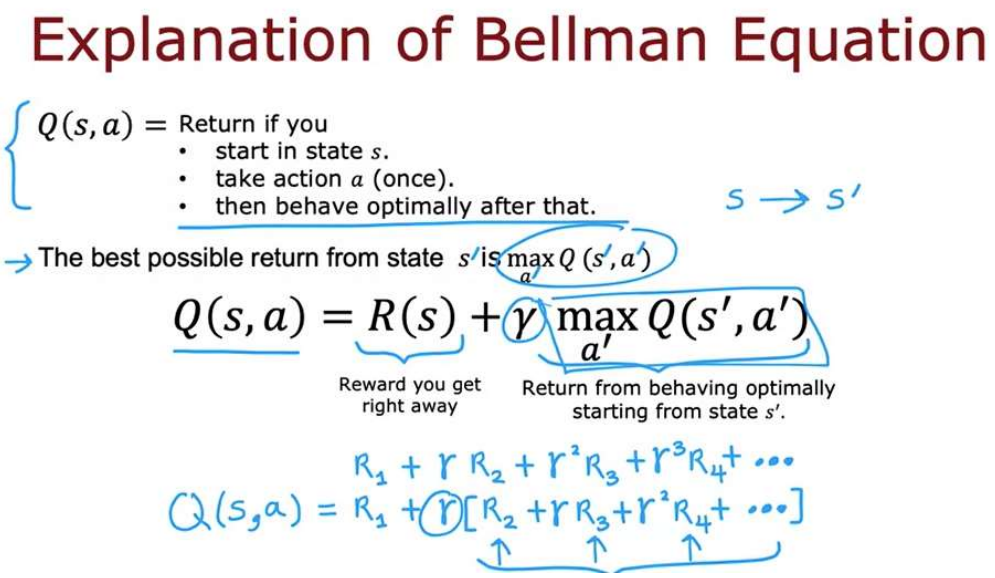
$Q(2,\rightarrow) = R(2) + 0.5 \max_{a'} Q(3,a')$
$= 0 + (0.5)25 = 12.5$

$s = 4$
$a = \leftarrow$
$s' = 3$

$Q(4,\leftarrow) = R(4) + 0.5 \max_{a'} Q(3,a')$
$= 0 + (0.5)25 = 12.5$

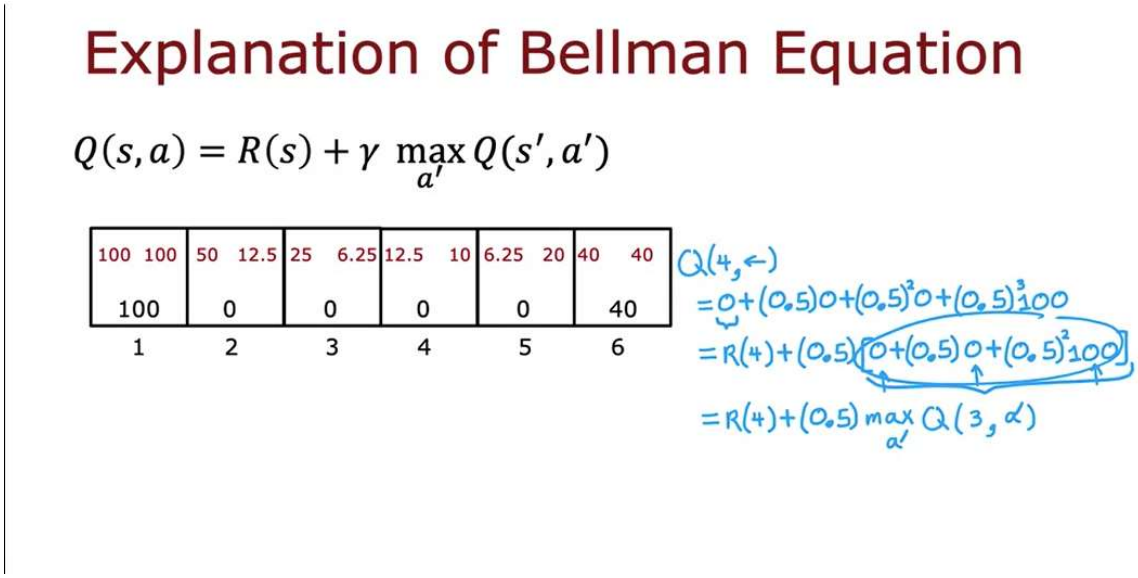
Wenn der aktuelle Zustand Zustand zwei ist und die Aktion darin besteht, nach rechts zu gehen, dann kommt man am nächsten Tag dazu, nachdem man geschrieben hat, dass S prim der Zustand 3 wäre. Die Bellman-Gleichung besagt, dass Q von 2 ist, rechts ist R von S. Dies R Zustand 2, der lediglich die Belohnung Null plus den Abzinsungsfaktor Gamma ist, den wir in diesem Beispiel auf 0,5 gesetzt haben, multipliziert mit dem Maximum der Q-Werte in Zustand S Primzahl in Zustand 3. Dies wird das Maximum von 25 und 6,25 sein, da dies max über einer Primzahl von q von S Primzahl Komma eine Primzahl ist. Dabei wird der größere Wert von 25 oder 6,25 angenommen, da dies die beiden Optionen für Zustand 3 sind. Dies entspricht Null plus 0,5 mal 25, was 12,5 entspricht, was glücklicherweise Q von zwei und dann die richtige Aktion ist. Schauen wir uns noch ein weiteres Beispiel an. Lassen Sie mich den Zustand 4 nehmen und sehen, was Q von Zustand 4 ist, wenn Sie sich entscheiden, nach links zu gehen. In diesem Fall ist der aktuelle Status vier. Die aktuelle Aktion besteht darin, nach links zu gehen. Der nächste Zustand, wenn Sie bei vier nach links beginnen können. Sie landen auch bei Zustand 3. Lassen Sie uns diese drei noch einmal primieren, die Bellman-Gleichung, wir sagen, sie ist gleich R von S. Unser Zustand vier, der Null plus 0,5 ist, der Abzinsungsfaktor Gamma von max über einer Primzahl von q von S prim. Das ist wieder der Zustand 3, Komma eine Primzahl. Auch hier betragen die Q-Werte für Zustand 3 25 und 6,25 und der größere davon ist 25. Das ergibt unsere 40 plus 0,5 mal 25, was wiederum 12,5 entspricht. Deshalb ist q von vier mit der verbleibenden Aktion auch gleich 12,5, nur eine Anmerkung: Wenn Sie sich in einem Endzustand befinden, vereinfacht sich die Bellman-Gleichung zu q von SA gleich r von S, weil es keinen Zustand S-Primzahl gibt und so weiter Die zweite Amtszeit würde wegfallen. Aus diesem Grund beträgt Q von S,A in den Endzuständen nur 100, 100 oder 40 40. Wenn Sie möchten, können Sie das Video gerne anhalten und die Bellman-Gleichung auf jede andere Zustandsaktion in diesem MDP anwenden und selbst überprüfen, ob dies der Fall ist klappt. Um es noch einmal zusammenzufassen: So hatten wir Q von S,A definiert. Wir haben zuvor gesehen, dass die bestmögliche Rendite aus jedem Zustand S max über einem Q von S,A ist. Um SNA umzubenennen, stellt sich tatsächlich heraus, dass die bestmögliche Rendite aus einem Zustand S-Primzahl maximal über S-Primzahl einer Primzahl liegt. Ich habe eigentlich nichts anderes gemacht, als S, S prim und a in eine Primzahl umzubenennen. Aber das wird einige der Intuitionen später etwas einfacher machen. Aber für jeden Zustand S prim, wie z. B. Zustand 3, ist die bestmögliche Rendite von beispielsweise Zustand 3 das Maximum über alle möglichen Aktionen von Q von S prim E prim. Auch hier gilt die Bellman-Gleichung. Die Intuition, die dies widerspiegelt, ist, dass man im Laufe der Zeit eine Reihe von Belohnungen sehen wird, wenn man von Zustand s ausgeht und Maßnahmen ergreift und danach optimal handelt. Insbesondere wird die Rendite aus der Belohnung im ersten Schritt plus

Gamma mal Belohnung im zweiten Schritt plus Gamma im Quadrat mal Belohnung im dritten Schritt usw. berechnet. Plus Punkt, Punkt, Punkt, bis Sie zum Endzustand gelangen. Was die Bellman-Gleichung besagt, ist, dass diese Folge von Belohnungen, also der Abzinsungsfaktor, in zwei Komponenten zerlegt werden kann. Erstens, dieses R von s, das ist die Belohnung, die Sie sofort erhalten. In der Literatur zum Reinforcement Learning wird dies manchmal auch als unmittelbare Belohnung bezeichnet, aber genau das ist R_1. Es ist die Belohnung, die Sie erhalten, wenn Sie in einigen Zuständen anfangen. Der zweite Term ist dann der folgende; Nachdem Sie in Zustand s begonnen und die Aktion a ausgeführt haben, gelangen Sie zu einem neuen Zustand s Prime. Die Definition von Q von s, a geht davon aus, dass wir uns danach optimal verhalten. Sobald wir die Primzahl erreicht haben, werden wir uns optimal verhalten und die bestmögliche Rendite aus der Primzahl des Zustandes erzielen. Was das ist, max einer Primzahl von Q von s Primzahl eine Primzahl, das ist die Rendite aus optimalem Verhalten, ausgehend von der Primzahl des Zustandes. Das ist genau das, was wir hier geschrieben haben: die bestmögliche Rendite, wenn man von der Spitze des Zustandes aus startet. Anders ausgedrückt ist die Gesamtrendite hier unten ebenfalls gleich R_1 plus, und dann werden wir Gamma in der Karte herausrechnen, also Gamma mal R_2 plus, und statt Gamma im Quadrat ist es einfach Gamma mal R_3 plus Gamma Quadrat mal R_4 plus Punkt Punkt Punkt. Beachten Sie, dass die Reihenfolge der Belohnungen, die Sie erhalten, wenn Sie mit der Primzahl des Zustands beginnen, R_2, R_3, dann R_4 usw. ist. Aus diesem Grund ist dieser Ausdruck hier die Gesamtrendite, wenn man von der Primzahl des Zustandes ausgeht. Wenn Sie sich optimal verhalten würden, dann sollte dieser Ausdruck die bestmögliche Rendite für den Start von der Primzahl des Zustandes sein, weshalb diese Folge von Rabattbelohnungen dem Maximum einer Primzahl von Q von s Primzahl einer Primzahl entspricht und es auch noch übrig blieb Dieser zusätzliche Abzinsungsfaktor Gamma ist vorhanden, weshalb Q von s, a auch hier gleich diesem Ausdruck ist.



Wenn Sie der Meinung sind, dass dies ziemlich kompliziert ist und Sie nicht alle Details befolgen, machen Sie sich darüber keine Sorgen. Solange Sie diese Gleichung anwenden, werden Sie die richtigen Ergebnisse erzielen. Ich hoffe jedoch, dass Sie die allgemeine Intuition mitnehmen, dass der Gesamtertrag, den Sie beim Problem des Reinforcement Learning erhalten, aus zwei Teilen besteht. Der erste Teil ist diese Belohnung, die Sie sofort erhalten, und der zweite Teil ist das Gamma-fache der Rendite, die Sie ab dem Prime des nächsten Zustandes erhalten. Wenn man diese beiden Komponenten zusammennimmt, ist R von s plus Gamma multipliziert mit der Rendite aus dem nächsten Zustand, also gleich der Gesamtrendite aus dem aktuellen Zustand s. Das ist der Kern der Bellman-Gleichung. Um dies nur auf unser früheres Beispiel Q von 4 links zurückzuführen. Das ist die

Gesamtrendite für den Start von Zustand 4 und den Übergang nach links. Wenn Sie in Zustand 4 nach links gehen würden, wären die Belohnungen, die Sie erhalten, 0 in Zustand 4, 0 in Zustand 3, 0 in Zustand 2 und dann 100, weshalb die Gesamtrendite so ist; 0,5 zum Quadrat plus 0,5 zum Quadrat, also 12,5. Die Bellman-Gleichung besagt, dass wir sie in zwei Teile aufteilen können. Es gibt diese Null, die R des Zustands vier ist, und dann plus 0,5 mal diese andere Folge, 0 plus 0,50 plus 0,5 zum Quadrat mal 100. Aber wenn man sich diese Folge anschaut, ist das wirklich die optimale Rendite aus der nächsten Zustand s Primzahl, den Sie erreicht haben, nachdem Sie die Aktion links von Zustand vier ausgeführt haben. Aus diesem Grund entspricht dies der Belohnung 4 plus dem 0,5-fachen der optimalen Rendite aus Zustand 3. Denn wenn Sie bei Zustand 3 beginnen würden, wären die Belohnungen, die Sie erhalten, Null, gefolgt von Null, gefolgt von 100, also ist dies die optimale Rendite aus Zustand 3 und deshalb ist dies nur R von 4 plus maximal 0,5 über einem Primzahl-Q von Zustand 3, einer Primzahl.



Ich kenne die Bellman-Gleichung, das ist eine etwas komplizierte Gleichung, die Ihre Gesamtrendite in die Belohnung zerlegt, die Sie sofort erhalten. Die unmittelbare Belohnung plus Gamma multipliziert mit den Erträgen aus der Primzahl des nächsten Zustands. Wenn es für Sie Sinn ergibt, aber nicht vollständig, ist es in Ordnung. Machen Sie sich darüber keine Sorgen. Sie können immer noch Bellmans Gleichungen anwenden, um einen Reinforcement-Learning-Algorithmus zum korrekten Funktionieren zu bringen, aber ich hoffe, dass Sie zumindest ein umfassendes Verständnis dafür haben, warum Sie die Belohnungen in das aufteilen, was Sie sofort erhalten, und in das, was Sie in der Zukunft erhalten. Ich hoffe das ergibt Sinn. Bevor wir mit der Entwicklung eines Reinforcement-Learning-Algorithmus fortfahren, präsentieren wir als nächstes ein optionales Video zu stochastischen Markov-Entscheidungsprozessen oder zu Reinforcement-Learning-Anwendungen, bei denen die Aktionen, wenn Sie sie ergreifen, einen leicht zufälligen Effekt haben können. Schauen Sie sich bei Bedarf das optionale Video an. Danach beginnen wir mit der Entwicklung eines Reinforcement-Learning-Algorithmus.

Zufällige (stochastische) Umgebung (optional)

Bei manchen Anwendungen ist das Ergebnis einer Aktion nicht immer völlig zuverlässig. Wenn Sie beispielsweise Ihrem Mars-Rover befehlen, nach links zu fahren, gibt es möglicherweise einen leichten Steinschlag, oder der Boden ist wirklich rutschig, sodass er ausrutscht und in die falsche Richtung fährt. In der Praxis gelingt es vielen Robotern nicht immer, genau das zu tun, was Sie ihnen sagen, weil der Wind weht, der Kurs abweicht, das Rad durchrutscht oder etwas anderes passiert. Es gibt eine Verallgemeinerung des Reinforcement-Learning-Frameworks, über das wir bisher

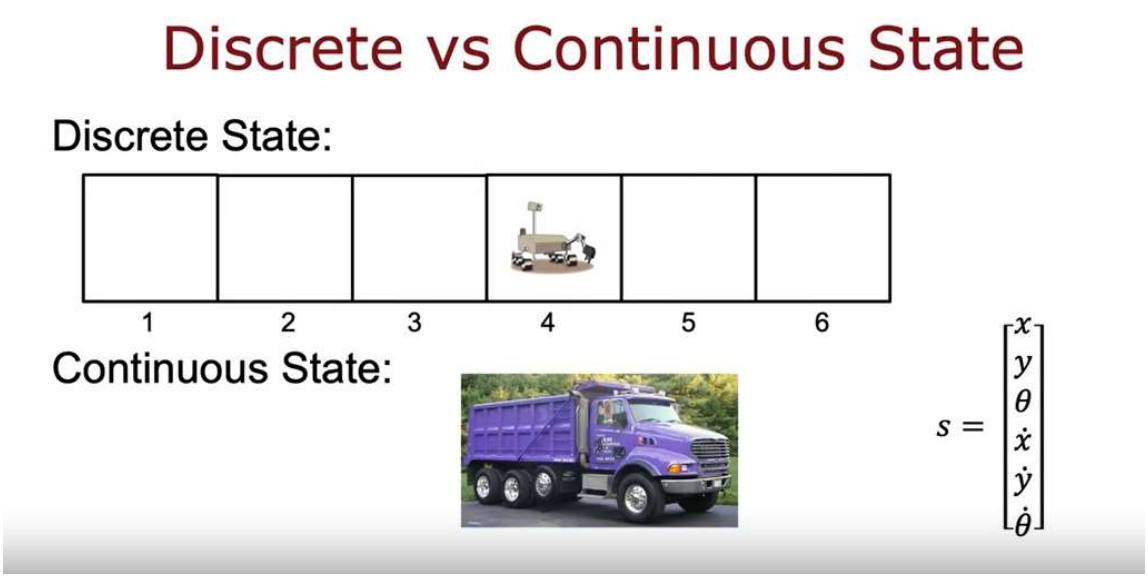
gesprochen haben und das zufällige oder stochastische Umgebungen modelliert. In diesem optionalen Video sprechen wir über die Funktionsweise dieser Reinforcement-Learning-Probleme und fahren mit unserem vereinfachenden Mars-Rover-Beispiel fort. Nehmen wir an, Sie führen die Aktion aus und befehlen ihm, nach links zu fahren. In den meisten Fällen wird Ihnen das gelingen, aber was ist, wenn es in 10 Prozent der Fälle oder in 0,1 Prozent der Fälle tatsächlich versehentlich ausrutscht und in die entgegengesetzte Richtung geht? Wenn Sie ihm befehlen, nach links zu gehen, besteht eine Wahrscheinlichkeit von 90 Prozent oder eine Chance von 0,9, dass er korrekt in die linke Richtung geht. Aber die Wahrscheinlichkeit von 0,1, tatsächlich nach rechts zu gehen, beträgt in diesem Beispiel eine Wahrscheinlichkeit von 9 Prozent, in beispielsweise drei zu landen, und eine Chance von 10 Prozent, in Zustand fünf zu landen. Wenn Sie ihm umgekehrt befehlen würden, nach rechts zu gehen und die richtige Aktion auszuführen, besteht eine Chance von 0,9, dass er in Zustand fünf landet, und eine Chance von 0,1, dass er in Zustand drei landet. Dies wäre ein Beispiel für eine stochastische Umgebung. Mal sehen, was bei diesem Problem des verstärkenden Lernens passiert. Nehmen wir an, Sie verwenden die hier gezeigte Richtlinie, bei der Sie in den Stufen 2, 3 und 4 nach links gehen und in Stufe fünf nach rechts gehen oder versuchen, in Stufe fünf nach rechts zu gehen. Wenn Sie in BundesZustand vier beginnen und diese Richtlinie befolgen würden, könnte die tatsächliche Reihenfolge der besuchten BundesZustanden zufällig sein. Zum Beispiel gehst du in Zustand vier nach links, und vielleicht hat deine Schleife Glück, und sie erreicht tatsächlich den Zustand drei, und dann versuchst du noch einmal, nach links zu gehen, und vielleicht kommt sie tatsächlich dort an. Sie sagen ihm, er soll wieder nach links gehen, und er gelangt in diesen Zustand. Wenn dies der Fall ist, erhalten Sie am Ende die Belohnungsfolge 000100. Wenn Sie jedoch genau dieselbe Richtlinie ein zweites Mal ausprobieren würden, haben Sie möglicherweise etwas weniger Glück, wenn Sie hier zum zweiten Mal beginnen. Versuchen Sie, nach links zu gehen und sehen Sie, wie es gelingt, also eine Null aus Zustand vier und eine Null aus Zustand drei. Hören Sie, wie Sie ihm sagen, er solle nach links gehen, aber dieses Mal haben Sie Pech, und der Roboter rutscht aus und kehrt stattdessen zu Zustand vier zurück. Dann wird Ihnen beigebracht, links und links und links zu rufen und schließlich die Belohnung von 100 zu erreichen. In diesem Fall ist dies die Reihenfolge der Belohnungen, die Sie beobachten. Dieses von vier auf drei auf vier drei zwei dann eins, oder es ist sogar möglich, wenn Sie von Stufe vier sagen, dass sie gemäß der Richtlinie nach links gehen soll, könnten Sie schon beim ersten Schritt Pech haben und am Ende zu Stufe fünf gehen, weil sie verrutscht ist. Geben Sie dann fünf ein, Sie befehlen ihm, nach rechts zu fahren, und es gelingt Ihnen, wenn Sie hier landen. In diesem Fall ist die Reihenfolge der Prämien, die Sie sehen, 0040, da sie von vier auf fünf ging und dann sechs lautet. Wir hatten die Rendite zuvor als diese Summe der ermäßigten Prämien ausgeschrieben. Wenn das Problem des verstärkenden Lernens jedoch stochastisch ist, sehen Sie nicht eine Folge von Belohnungen, die Sie mit Sicherheit sehen, sondern diese Folge verschiedener Belohnungen. Bei einem stochastischen Verstärkungslernproblem geht es uns nicht darum, die Rendite zu maximieren, da es sich dabei um eine Zufallszahl handelt. Unser Interesse besteht darin, den Durchschnittswert der Summe der ermäßigten Prämien zu maximieren. Mit Durchschnittswert meine ich, wenn Sie Ihre Police tausendmal oder 100.000 Mal oder eine Million Mal ausprobieren würden, erhalten Sie viele verschiedene Belohnungssequenzen wie diese, und wenn Sie den Durchschnitt über all diese nehmen würden Verschiedene Sequenzen der Summe der ermäßigten Prämien, dann nennen wir das die erwartete Rendite. In der Statistik ist der Begriff „erwartet“ nur eine andere Art, „Durchschnitt“ zu sagen. Dies bedeutet jedoch, dass wir das maximieren möchten, was wir im Durchschnitt in Bezug auf die Summe der ermäßigten Prämien erwarten. Die mathematische Notation hierfür ist, dies als E zu schreiben. E steht für den erwarteten Wert von R1 plus Gamma R2 plus und so weiter. Die Aufgabe des Reinforcement-Learning-Algorithmus besteht darin, einen Richtlinien-Pi auszuwählen, um den Durchschnitt oder die erwartete Summe der ermäßigten Belohnungen zu maximieren. Zusammenfassend lässt sich sagen, dass das Ziel bei einem stochastischen Verstärkungslernproblem oder einem stochastischen Markov-

Entscheidungsprozess darin besteht, eine Richtlinie auszuwählen, die uns sagt, welche Maßnahmen wir im Zustand S ergreifen sollen, um die erwartete Rendite zu maximieren. Die letzte Art und Weise, wie sich dies ändert, worüber wir gesprochen haben, besteht darin, dass es die Bellman-Gleichung ein wenig modifiziert. Hier ist die Bellman-Gleichung genau so, wie wir sie aufgeschrieben haben. Der Unterschied besteht nun darin, dass, wenn Sie die Aktion a im Zustand s ausführen, die nächste Primzahl des Zustands, die Sie erreichen, zufällig ist. Wenn Sie sich in Zustand 3 befinden und ihm sagen, dass er nach links zum nächsten Zustand s Primzahl gehen soll, könnte es Zustand 2 oder Zustand 4 sein. S Primzahl ist jetzt zufällig, weshalb wir auch einen Durchschnittsoperator oder einsetzen unerwarteter Operator hier. Wir sagen, dass die Gesamtrendite von Zustand s, wenn er Maßnahmen ergreift, gleich der Belohnung ist, die Sie sofort erhalten, auch Sofortbelohnung genannt, zuzüglich des Abzinsungsfaktors Gamma plus dem, was Sie im Durchschnitt erwarten die Zukunft kehrt zurück. Wenn Sie Ihre Intuition darüber schärfen möchten, was mit diesen stochastischen Verstärkungslernproblemen passiert. Sie würden zu dem optionalen Labor zurückkehren, das ich Ihnen gerade gezeigt habe, wo dieses Parameter-Fehltrittsproblem die Wahrscheinlichkeit ist, dass Ihr Marsrover in die entgegengesetzte Richtung fährt, als Sie es befohlen hatten. Wenn wir sagen, dass die zweite Fehltritt-Stütze 0,1 beträgt und das Notebook erneut ausgeführt wird, und diese Zahlen hier oben die optimale Rendite darstellen, wenn Sie die bestmöglichen Maßnahmen ergreifen würden, würden Sie diese optimale Richtlinie anwenden, aber der Roboter würde um 10 Prozent in die falsche Richtung gehen der Zeit und dies sind die q-Werte für dieses stochastische NTP. Beachten Sie, dass diese Werte jetzt etwas niedriger sind, da Sie den Roboter nicht mehr so gut steuern können wie zuvor. Die q-Werte sowie die optimalen Renditen sind etwas gesunken. Wenn man die Wahrscheinlichkeit eines Fehltritts erhöhen würde, würde der Roboter in sagen wir 40 Prozent der Fälle nicht einmal in die Richtung gehen. Sie beherrschen es nur in 60 Prozent der Fälle. Wenn er dorthin geht, wo Sie ihn angewiesen haben, fallen diese Werte am Ende sogar noch niedriger aus, weil Ihr Grad an Kontrolle über den Roboter abgenommen hat. Ich ermutige Sie, mit dem optionalen Labor herumzuspielen und den Wert der Fehltrittwahrscheinlichkeit zu ändern und zu sehen, wie sich das auf die for- Return oder die automatisch erwartete Rendite sowie auf die Q-Werte q von s a auswirkt. Bei allem, was wir bisher getan haben, haben wir diesen Markov-Entscheidungsprozess verwendet, diesen Marsrover mit nur sechs Zuständen. Für viele praktische Anwendungen wird die Anzahl der Zustände viel größer sein. Im nächsten Video nehmen wir das bisher besprochene Framework für Verstärkungslernen oder Markov-Entscheidungsprozess und verallgemeinern es auf diese viel umfangreichere und möglicherweise sogar interessantere Reihe von Problemen mit viel größeren und insbesondere kontinuierlichen Zustandsräumen. Schauen wir uns das im nächsten Video an.

Beispiel für continuous state space Anwendungen

Viele Robotersteuerungsanwendungen, einschließlich der Mondlanderanwendung, an der Sie im Übungslabor arbeiten, verfügen über kontinuierliche Zustandsräume. Werfen wir einen Blick darauf, was das bedeutet und wie wir das Konzept, über das wir gesprochen haben, auf diese kontinuierlichen Zustandsräume verallgemeinern können. Als Beispiel für einen vereinfachten Marsrover verwende ich einen diskreten Satz von Zuständen. Das bedeutet, dass sich der vereinfachte Marsrover nur an einer von sechs möglichen Positionen befinden kann. Aber die meisten Roboter können sich an mehr als einer von sechs oder einer beliebigen diskreten Anzahl von Positionen befinden, stattdessen können sie sich an einer sehr großen Anzahl kontinuierlicher Wertpositionen befinden. Wenn sich der Marsrover beispielsweise irgendwo auf einer Linie befinden könnte, wurde seine Position durch eine Zahl im Bereich von 0 bis 6 Kilometern angegeben, wobei jede Zahl dazwischen gültig ist. Das wäre ein Beispiel für einen kontinuierlichen Zustandsraum, da die Position durch eine Zahl wie 2,7 Kilometer oder 4,8 Kilometer oder eine andere Zahl zwischen null und sechs dargestellt würde. Schauen wir uns ein anderes Beispiel an. Ich werde für dieses Beispiel

die Anwendung der Steuerung eines Autos oder eines Lastwagens verwenden. Hier ist ein Spielzeugauto, ein russischer Spielzeuglastwagen. Dieser gehört meiner Tochter.



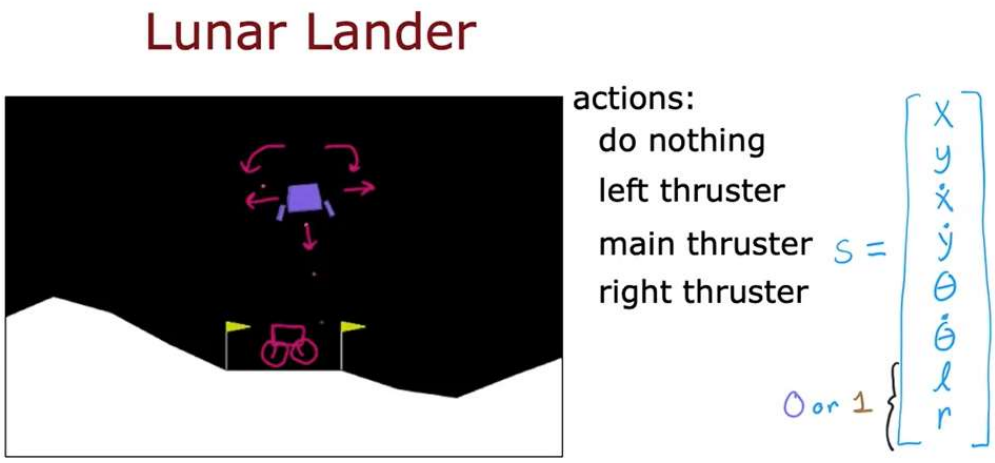
Wenn Sie ein selbstfahrendes Auto oder einen selbstfahrenden LKW bauen und dies so steuern möchten, dass es reibungslos fährt, kann der Zustand dieses LKWs einige Zahlen wie diese, die X-Position, die Y-Position und möglicherweise die Ausrichtung enthalten. In welche Richtung ist es ausgerichtet? Vorausgesetzt, der LKW bleibt auf dem Boden, müssen Sie sich wahrscheinlich keine Gedanken darüber machen, wie groß und wie hoch er ist. Dieser Zustand umfasst x, y und den Winkel Theta sowie möglicherweise seine Geschwindigkeiten in x-Richtung, die Geschwindigkeit in y-Richtung und wie schnell er sich dreht. Dreht es sich mit einem Grad pro Sekunde oder mit 30 Grad pro Sekunde oder dreht es sich sehr schnell mit 90 Grad pro Sekunde? Bei einem Lastkraftwagen oder einem Auto enthält der Zustand möglicherweise nicht nur eine Zahl, etwa die Anzahl der Kilometer auf dieser Linie, sondern möglicherweise auch sechs Zahlen: die X-Position, die Y-Position und die Ausrichtung, worauf ich hinauswill um mit dem griechischen Alphabet Theta zu bezeichnen, sowie seine Geschwindigkeit in der x-Richtung, die ich mit x-Punkt bezeichnen werde, das heißt also, wie schnell sich diese, Theta-Punkt, der angibt, wie schnell sich der Winkel des Autos ändert. Beim Beispiel des Mars-Rovers 60 hingegen war der Zustand nur eine von sechs möglichen Zahlen. Es könnte eins, zwei, drei, vier, fünf oder sechs sein. Für das Auto würde der Zustand diesen Vektor aus sechs Zahlen umfassen, und jede dieser Zahlen kann jeden Wert innerhalb ihres gültigen Bereichs annehmen. Theta sollte beispielsweise zwischen null und 360 Grad liegen. Schauen wir uns ein anderes Beispiel an. Wie würden Sie die Position eines Hubschraubers charakterisieren, wenn Sie einen Reinforcement-Learning-Algorithmus zur Steuerung eines autonomen Hubschraubers entwickeln würden? Zur Veranschaulichung habe ich hier einen kleinen Spielzeughubschrauber dabei. Die Positionierung des Hubschraubers würde die X-Position umfassen, z. B. wie weit nördlich oder südlich ein Hubschrauber entfernt ist, und die Y-Position. Vielleicht, wie weit der Hubschrauber auf der Ost-West-Achse entfernt ist, und dann auch z, die Höhe des Hubschraubers über dem Boden. Aber außer der Position hat der Helikopter auch eine Orientierung, und herkömmlicherweise besteht eine Möglichkeit, seine Orientierung zu erfassen, darin, drei zusätzliche Zahlen zu verwenden, von denen eine die Reihe des Helikopters erfasst. Rollt es nach links oder nach rechts? Die Neigung, ist es Vorwärtsneigung oder Neigung nach oben, Neigung nach hinten, und schließlich die Gierrichtung, die westlich der Kompassausrichtung ist. Wenn nach Norden oder Osten oder Süden oder Westen ausgerichtet? Zusammenfassend umfasst der Zustand des Hubschraubers seine Position beispielsweise in Nord-Süd-Richtung, seine Position in Ost-West-Richtung, y ist die Höhe über dem Boden sowie die Ausrichtung, die Neigung und die Gierung des Hubschraubers. Um dies

aufzuschreiben, umfasst der Zustand daher die Position x, y, z und dann den Zeilenabstand und die Gierung, die mit den griechischen Alphabeten Phi, Theta und Omega bezeichnet werden. Um den Hubschrauber steuern zu können, müssen wir aber auch seine Geschwindigkeit in x-Richtung, in y-Richtung und in z-Richtung sowie seine Drehrate, auch Winkelgeschwindigkeit genannt, kennen. Wie schnell ändert sich diese Reihe und wie schnell ändert sich diese Neigung und wie schnell ändert sich ihre Gierbewegung? Dies ist eigentlich der Zustand, der zur Steuerung autonomer Hubschrauber verwendet wird. Ist diese Liste von 12 Zahlen die Eingabe in eine Richtlinie, und die Aufgabe einer Richtlinie besteht darin, diese 12 Zahlen zu betrachten und zu entscheiden, welche Maßnahmen im Helikopter angemessen sind.

Daher ist jedes kontinuierliche Zustandliche Verstärkungslernproblem oder ein kontinuierlicher Zustandlicher Markov-Entscheidungsprozess ein kontinuierliches MTP. Der Zustand des Problems ist nicht nur einer von wenigen möglichen diskreten Werten, wie etwa eine Zahl von 1 bis 6. Stattdessen handelt es sich um einen Zahlenvektor, von dem jeder einen beliebigen Wert aus einer großen Anzahl annehmen kann. Im Übungslabor dieser Woche können Sie selbst einen Reinforcement-Learning-Algorithmus implementieren, der auf eine simulierte Mondlanderanwendung angewendet wird. Etwas auf dem Mond zu landen ist eine Simulation. Werfen wir im nächsten Video einen Blick darauf, was diese Anwendung beinhaltet, da es eine weitere kontinuierliche Statusanwendung geben wird.

Mondlander

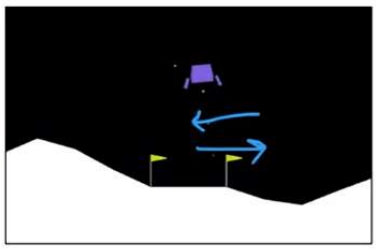
Mit dem Mondlander können Sie ein simuliertes Fahrzeug auf dem Mond landen. Es ist wie ein lustiges kleines Videospiel, das von vielen Forschern des Reinforcement Learning verwendet wird. Werfen wir einen Blick darauf, was es ist. In dieser Anwendung steuern Sie eine Mondlandefähre, die sich schnell der Mondoberfläche nähert. Und Ihre Aufgabe besteht darin, die Triebwerke zum richtigen Zeitpunkt abzufeuern, um es sicher auf dem Landeplatz zu landen.



Um Ihnen einen Eindruck davon zu vermitteln, wie es aussieht. Dies ist die erfolgreiche Landung des Mondlanders und er feuert Triebwerke nach unten sowie nach links und rechts ab, um sich für die Landung zwischen diesen beiden gelben Flaggen zu positionieren. Oder wenn die Strategie des Verstärkungslandealgorithmus nicht funktioniert, dann könnte es so aussehen, dass der Lander unglücklicherweise auf der Mondoberfläche zerschmettert ist. In dieser Anwendung haben Sie für jeden Zeitschritt vier mögliche Aktionen. Sie könnten entweder nichts tun. In diesem Fall ziehen Sie die Kräfte der Trägheit und der Schwerkraft zur Mondoberfläche. Oder Sie können ein linkes Triebwerk auslösen, wenn Sie einen kleinen roten Punkt auf der linken Seite sehen, der die linke Seite auslöst. Sie neigen dazu, den Mondlander nach rechts zu schieben, oder Sie können das Haupttriebwerk zünden, das hier unten nach unten drückt. Oder Sie können das rechte Triebwerk abfeuern und das zündet das rechte Triebwerk, das Sie nach links drückt, und Ihre Aufgabe ist es, im Laufe der Zeit immer wieder Aktionen auszuwählen. Die Mondlandefähre befindet sich also sicher zwischen diesen beiden Flaggen hier auf dem Landeplatz. Um den Aktionen einen kürzeren Namen zu geben, nenne ich die Aktionen manchmal „nichts“, was nichts tun bedeutet, oder „links“, was „weiter links Triebwerk“ bedeutet, oder „Haupt“, was bedeutet, dass das Haupttriebwerk nach unten oder rechts abgefeuert wird. Deshalb nenne ich die Aktionen nichts mehr. Maine und schreibe später kurz in diesem Video. Wie wäre es mit der Sichtweise des Zustandes darauf? Mtp sind die Zustände seine Position X und Y. Also wie weit links oder rechts und wie hoch oben ist es sowie Geschwindigkeit xy, wie schnell es sich in horizontaler und vertikaler Richtung bewegt und dann auch der Winkel. Wie weit ist die Mondlandefähre also nach links bzw. nach rechts geneigt? Ist die Winkelgeschwindigkeit nicht vom Schicksal bestimmt? Und schließlich, weil ein kleiner Unterschied in der Positionierung einen großen Unterschied darüber macht, ob es gelandet wird oder nicht. Wir werden zwei weitere Variablen im Zustandsvektor haben, die wir l und r nennen. Dies entspricht der Frage, ob das linke Bein geerdet ist, also ob das linke Bein auf dem Boden sitzt oder nicht, und r, was der Frage entspricht, ob das rechte Bein auf dem Boden sitzt oder nicht. Also während xy x.theta Theta. Unsere Zahlen l und r haben binäre Werte und können nur die Werte Null oder Eins annehmen, je nachdem, ob das linke und das rechte Bein den Boden berühren. Endlich seine Belohnungsfunktion für den Mondlander.

Reward Function

- Getting to landing pad: 100 – 140
- Additional reward for moving toward/away from pad.
- Crash: -100
- Soft landing: +100
- Leg grounded: +10
- Fire main engine: -0.3
- Fire side thruster: -0.03

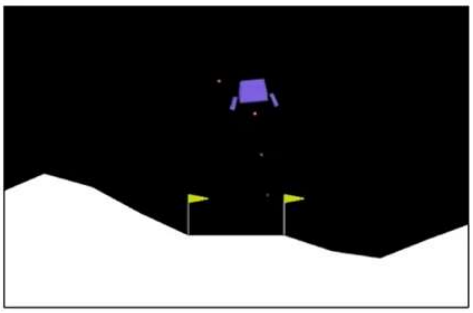


Wenn es den Landeplatz erreicht, erhält man keine Belohnung zwischen 100 und 140, je nachdem, wie gut es geflogen ist, als man die Mitte des Landeplatzes erreicht hat. Wir geben ihm außerdem eine zusätzliche Belohnung für die Bewegung auf das Pad zu oder von ihm weg, so dass es, wenn es sich dem Pad nähert, eine positive Belohnung erhält und sich wegbewegt und davondriftet. Es erhält eine negative Belohnung. Wenn es abstürzt, erhält es eine große Belohnung von -100, es erreicht eine sanfte Landung, also eine Landung. Es kommt zu einem weiteren Absturz, es erhält eine Belohnung von +100 für jedes Bein, das linke Bein oder das rechte Glied, bei dem es am Boden bleibt. Es erhält eine Belohnung von +10 und um es schließlich zu ermutigen, nicht zu viel Treibstoff und Feuer zu verschwenden, sind Triebwerke nicht unbedingt notwendig. Jedes Mal, wenn es das Haupttriebwerk zündet, geben wir ihm eine Belohnung von -0,3 und jedes Mal, wenn es die Triebwerke auf der linken oder rechten Seite zündet, geben wir ihm eine Belohnung von -0,03. Beachten Sie, dass es sich hierbei um eine mäßig komplexe Belohnungsfunktion handelt. Die Designer der Mondlander-Anwendung haben sich tatsächlich Gedanken darüber gemacht, welches Verhalten Sie genau wünschen, und es in der Belohnungsfunktion kodifiziert. Um mehr Verhaltensweisen anzuregen, die Sie sich wünschen, und Angst vor Verhaltensweisen wie einem Absturz zu haben, die Sie nicht wollen. Wenn Sie Ihre eigene Reinforcement-Learning-Anwendung erstellen, müssen Sie in der Regel etwas darüber nachdenken, genau anzugeben, was Sie wollen oder nicht wollen, und dies in der Belohnungsfunktion zu kodifizieren. Aber die Angabe der Belohnungsfunktion sollte sich dennoch als viel einfacher erweisen, um die genau richtige Aktion anzugeben, die von jedem einzelnen Zustand aus durchgeführt werden soll. Was für diese und viele andere Anwendungen des verstärkenden Lernens viel schwieriger ist. Das Problem der Mondlandefähre ist also wie folgt.

Lunar Lander Problem

Learn a policy π that, given

$$s = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \theta \\ \dot{\theta} \\ l \\ r \end{bmatrix}$$



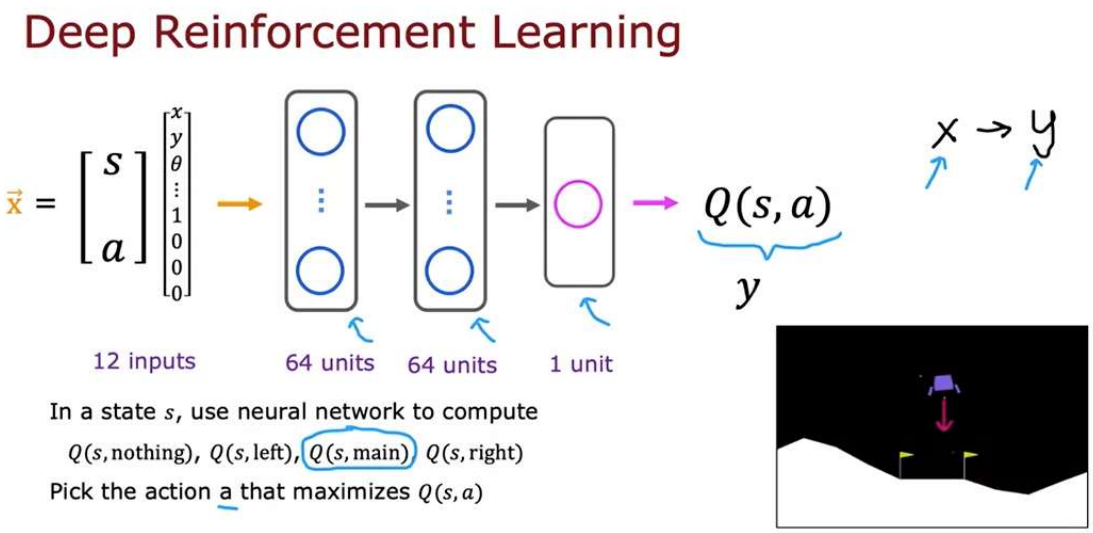
picks action $a = \pi(s)$ so as to maximize the return.

$$\gamma = 0.985$$

Unser Ziel ist es, eine Politik-Pi zu lernen. Wenn ein Zustand S wie hier beschrieben gegeben wird, wird eine Aktion ausgewählt, die dem Pi von S entspricht. Um die Rendite der Summe der ermäßigten Belohnungen zu maximieren. Und normalerweise würde man für den Mondlander einen ziemlich großen Wert für Gamma Ra verwenden. Tatsächlich würde man für die Kamera einen Wert von 0,985 verwenden, also ziemlich nahe bei eins. Und wenn Sie einen Richtlinien-Pi erlernen können, der dies tut, dann landen Sie erfolgreich in dieser aufregenden Mondlandeanwendung, und wir sind jetzt endlich bereit, einen Lernalgorithmus zu entwickeln, der Deep Learning oder neuronale Netze verwendet, um eine Richtlinie zu entwickeln Lande den Mondlander.

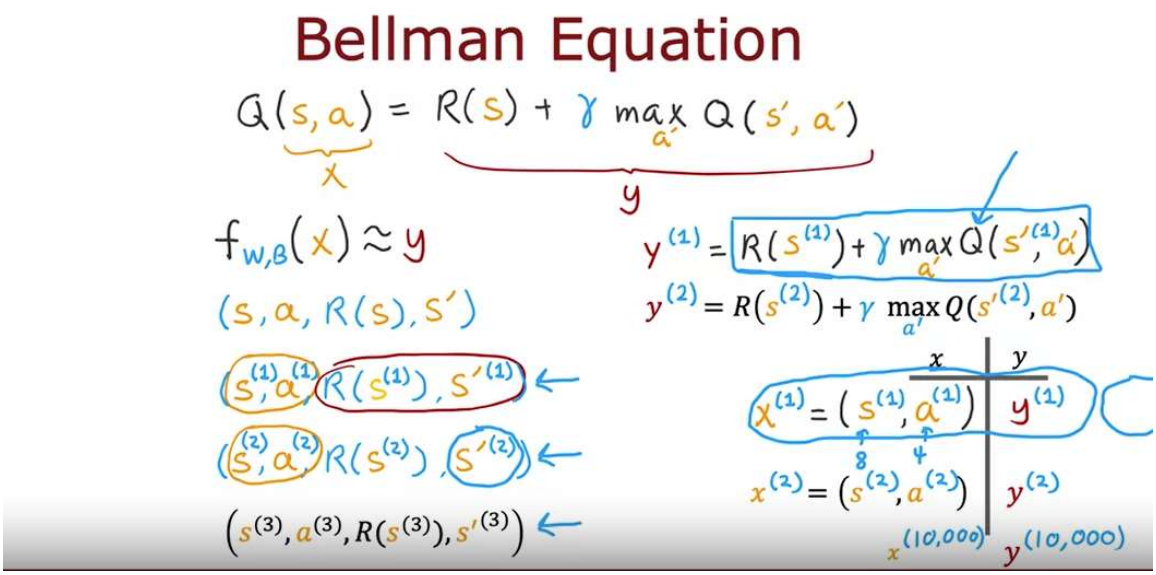
Erlernen der State Value Funktion

Sehen wir uns an, wie wir Reinforcement Learning zur Steuerung des Mondlanders oder für andere Reinforcement-Learning-Probleme nutzen können. Die Schlüsselidee besteht darin, dass wir ein neuronales Netzwerk trainieren, die Zustandsaktionswertfunktion Q von s, a zu berechnen oder anzunähern, und dass wir dadurch wiederum gute Aktionen auswählen können. Mal sehen, wie es funktioniert. Das Herzstück des Lernalgorithmus besteht darin, dass wir ein neuronales Netzwerk trainieren, das den aktuellen Zustand und die aktuelle Aktion eingibt und Q von s, a berechnet oder annähert. Insbesondere für die Mondlandeeinheit werden wir die Zustände und alle Maßnahmen ergreifen und sie zusammenfügen. Konkret handelte es sich bei dem Zustand um die Liste von acht Zahlen, die wir zuvor gesehen haben. Sie haben also xy, der Zustand.



Schließlich haben wir vier mögliche Aktionen: nichts, links, Haupt, eine Hauptmaschine und rechts. Wir können jede dieser vier Aktionen mithilfe eines One-Hot-Feature-Vektors codieren. Wenn die Aktion die erste Aktion wäre, könnten wir sie mit 1, 0, 0, 0 kodieren, oder wenn es die zweite Aktion wäre, um den linken Cluster zu finden, könnten wir sie mit 0, 1, 0, 0 kodieren. Diese Liste mit 12 Zahlen, acht Zahlen für den Zustand und dann vier Zahlen, eine One-Hot-Kodierung der Aktion, das sind die Eingaben, die wir in das neuronale Netzwerk haben, und ich werde das X nennen. Wir nehmen dann diese 12 Zahlen und Füttere sie einem neuronalen Netzwerk mit beispielsweise 64 Einheiten in der ersten verborgenen Schicht, 64 Einheiten in der zweiten verborgenen Schicht und dann einem einzelnen Ausgang in der Ausgabeschicht. Die Aufgabe des neuronalen Netzwerks ist die Ausgabe Q von s, a. Die Zustandsaktionswertfunktion für das Mondlandegerät unter Berücksichtigung der Eingaben s und a. Da wir in Kürze Trainingsalgorithmen für neuronale Netze verwenden werden, werde ich diesen Wert Q von s, a auch als den Zielwert Y bezeichnen, auf dessen Annäherung das neuronale Netz trainiert wurde. Beachten Sie, dass ich gesagt habe, dass sich verstärkendes Lernen von überwachtem Lernen unterscheidet, aber was wir tun werden, ist nicht, einen Zustand einzugeben und eine Aktion auszugeben. Was wir tun werden, ist, ein Zustands-Aktionspaar einzugeben und es versuchen zu lassen, Q von s, a auszugeben, und die Verwendung eines neuronalen Netzwerks innerhalb des Reinforcement-Learning-Algorithmus auf diese Weise wird sich als ziemlich gut herausstellen. Wir werden die Details gleich sehen, also machen Sie sich keine Sorgen, wenn es noch keinen Sinn ergibt. Aber wenn Sie ein neuronales Netzwerk mit geeigneten Parametern in den verborgenen Schichten und in der oberen Schicht trainieren können, um eine gute Schätzung des Q von s, a zu erhalten, dann können Sie das immer dann tun, wenn sich Ihr Mondlandegerät in einem bestimmten Zustand befindet Verwenden Sie dann das neuronale Netzwerk, um Q von s, a zu berechnen. Für alle vier Aktionen können Sie Q von s, nichts, Q von s, links, Q von s, Haupt, Q von s, rechts berechnen und schließlich die entsprechende Aktion a auswählen, je nachdem, welche davon den höchsten Wert hat. Wenn beispielsweise Q von s, main von diesen vier Werten der größte ist, dann würden Sie sich dafür entscheiden, das Haupttriebwerk der Mondlandefähre anzuzünden. Es stellt sich die Frage: Wie trainiert man ein neuronales Netzwerk, um Q von s, a auszugeben? Es stellt sich heraus, dass der Ansatz darin besteht, die Bellman-Gleichungen zu verwenden, um einen Trainingssatz mit vielen Beispielen x und y zu erstellen. Anschließend verwenden wir überwachtes Lernen genau so, wie Sie es im zweiten Kurs gelernt haben, als wir über neuronale Netze gesprochen haben. Um mithilfe des überwachten Lernens zu lernen, wird eine Zuordnung von x nach y durchgeführt, d. h. eine Zuordnung vom Zustandsaktionspaar zu diesem Zielwert Q von s, a. Aber wie erhält man den Trainingssatz mit

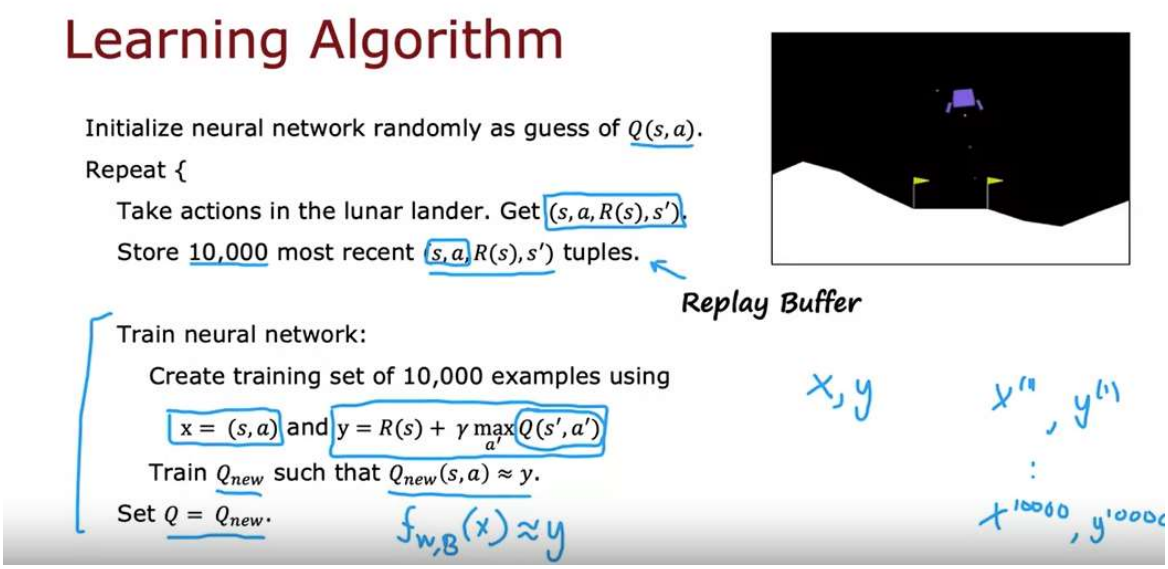
Werten für x und y, auf dem man dann ein neuronales Netzwerk trainieren kann? Lass uns einen Blick darauf werfen.



Hier ist die Bellman-Gleichung: Q von s, a ist gleich R von s plus Gamma, max einer Primzahl, Q von s Primzahl, eine Primzahl. Auf der rechten Seite soll das Q von s, a gleich sein, also nenne ich diesen Wert auf der rechten Seite y und die Eingabe in das neuronale Netzwerk ist ein Zustand und eine Aktion, also I Ich werde das x nennen. Die Aufgabe des neuronalen Netzwerks besteht darin, x einzugeben, also das Zustands-Aktionspaar einzugeben, und zu versuchen, den Wert auf der rechten Seite genau vorherzusagen. Beim überwachten Lernen trainierten wir ein neuronales Netzwerk, um eine Funktion f zu lernen, die von einer Reihe von Parametern W und B abhängt, den Parametern der verschiedenen Schichten des neuronalen Netzwerks, und es war die Aufgabe des neuronalen Netzwerks, x einzugeben. Hoffentlich setze ich etwas in die Nähe des Zielwerts y. Die Frage ist, wie wir einen Trainingssatz mit den Werten x und y erstellen können, aus dem ein neues Netzwerk lernen kann. Folgendes werden wir tun. Wir werden die Mondlandefähre nutzen und einfach versuchen, darin verschiedene Aktionen durchzuführen. Wenn wir noch keine gute Richtlinie haben, werden wir nach dem Zufallsprinzip Maßnahmen ergreifen, weiter, links, weiter, rechts, weiter, Hauptmaschine, nichts tun. Indem wir einfach verschiedene Dinge im Mondlander-Simulator ausprobieren, werden wir viele Beispiele dafür beobachten, wie wir uns in einem bestimmten Zustand befinden und etwas unternommen haben, vielleicht eine gute Aktion, vielleicht eine schreckliche Aktion, so oder so. Dann erhielten wir eine Belohnung R von s dafür, dass wir uns in diesem Zustand befanden, und als Ergebnis unserer Aktion gelangten wir in einen neuen Zustand, S'. Wenn Sie im Mondlander verschiedene Aktionen ausführen, sehen Sie dieses S, a, R von s, S', und wir nennen sie im Python-Code oft Tupel. Wenn Sie sich zum Beispiel einmal in einem Zustand S befinden und diesem einfach einen Index geben, wir nennen ihn S^1, und Sie zufällig eine Aktion a^1 ausführen, könnte dies nichts Links, Hauptstöße dort oder rechts sein. Infolgedessen haben Sie eine Belohnung erhalten und möchten in einem Zustand S^'1 aufsteigen. Vielleicht befinden Sie sich zu unterschiedlichen Zeiten in einem anderen Zustand S2, Sie haben eine andere Aktion durchgeführt, es könnte eine gute Aktion sein, es könnte eine schlechte Aktion sein, es könnte eine der vier Aktionen sein, und Sie haben die Belohnung erhalten, und dann möchten Sie mit S aufsteigen '2 und so weiter mehrmals. Vielleicht haben Sie das 10.000 Mal oder sogar mehr als 10.000 Mal gemacht, also müssten Sie den Weg nicht nur mit S^1, a^1 usw., sondern bis zu S^10.000, a^10.000 sparen. Es stellt sich heraus, dass jedes dieser Tupel ausreicht, um ein einzelnes Trainingsbeispiel x^1, y^1 zu erstellen. Im Einzelnen erfahren Sie hier, wie Sie es tun. Dieses erste Tupel enthält vier Elemente. Die ersten beiden werden zur Berechnung von x^1 und die zweiten beiden zur Berechnung von y^1 verwendet. Insbesondere wird

x^1 zusammen nur S^1, a^1 sein. S^1 wäre acht Zahlen, der Zustand der Mondlandefähre, a^1 wäre vier Zahlen, die One-Hot-Kodierung der jeweiligen Aktion, und y^1 würde mithilfe der rechten Seite der Bellman-Gleichung berechnet. Insbesondere besagt die Bellman-Gleichung: Wenn Sie S^1, a^1 eingeben, soll Q von S^1, a^1 auf dieser rechten Seite gleich R von S^1 plus Gamma max sein a' von Q von S^1'a' prim. Beachten Sie, dass diese beiden Elemente des Tupels auf der rechten Seite Ihnen genügend Informationen liefern, um dies zu berechnen. Wissen Sie, was R von S^1 ist? Das ist die Belohnung, die Sie sich den Weg hierher erspart haben. Plus den Abzinsungsfaktor Gamma mal Max über alle Aktionen, a' von Q von S'^1, das ist ein Zustand, den Sie in diesem Beispiel erreicht haben, und nehmen Sie dann das Maximum über alle möglichen Aktionen, a'. Ich werde es y^1 nennen. Wenn Sie dies berechnen, wird dies eine Zahl wie 12,5 oder 17 oder 0,5 oder eine andere Zahl sein. Wir speichern diese Zahl hier als y^1, sodass dieses Paar x^1, y^1 das erste Trainingsbeispiel in diesem niedrigen Datensatz wird, den wir berechnen. Nun fragen Sie sich vielleicht, woher Q of S', a' oder Q of S'^1, a' kommt. Nun, zunächst wissen wir nicht, was die Q-Funktion ist. Es stellt sich jedoch heraus, dass Sie, wenn Sie nicht wissen, was die Q-Funktion ist, zunächst eine völlig zufällige Vermutung anstellen können. Was ist die Q-Funktion? Auf der nächsten Folie werden wir sehen, dass [unverständlich] trotzdem funktioniert. Aber bei jedem Schritt wird Q hier nur eine Vermutung sein. Sie werden mit der Zeit besser, je nachdem, was die tatsächliche Q-Funktion ist. Schauen wir uns das zweite Beispiel an. Wenn Sie eine zweite Erfahrung gemacht haben, bei der Sie sich im Zustand S^2 befinden, um zu a^2 zu gelangen, diese Belohnung erhalten haben und dann in diesen Zustand gelangt sind. Dann würden wir in diesem Datensatz ein zweites Trainingsbeispiel erstellen, x^2, wobei die Eingabe jetzt S^2, a^2 ist, sodass die ersten beiden Elemente zur Berechnung der Eingabe x verwendet werden und dann y^2 gleich ist R von s^2 plus Gammamax von a' Q von S' zu a', und was auch immer diese Zahl ist, y^2. Wir fügen dies hier in unseren kleinen, aber wachsenden Trainingssatz ein und so weiter und so weiter, bis Sie am Ende vielleicht 10.000 Trainingsbeispiele mit diesen x-, y-Paaren haben. Was wir später sehen werden, ist, dass wir tatsächlich diesen Trainingssatz verwenden, bei dem die x-Werte Eingaben mit 12 Features und die y-Werte nur Zahlen sind. Wir trainieren ein neues Netzwerk beispielsweise mit dem mittleren quadratischen Fehlerverlust, um zu versuchen, y als Funktion der Eingabe x vorherzusagen. Was ich hier beschreibe, ist nur ein Teil des Lernalgorithmus, den wir verwenden werden. Lassen Sie uns auf der nächsten Folie alles zusammenfassen und sehen, wie alles in einem einzigen Algorithmus zusammenkommt. Werfen wir einen Blick darauf, wie ein vollständiger Algorithmus zum Erlernen der Q-Funktion aussieht. Zuerst nehmen wir unser neuronales Netzwerk und initialisieren alle Parameter des neuronalen Netzwerks zufällig. Zunächst haben wir keine Ahnung, ob es sich um eine Q-Funktion handelt. Wählen wir einfach völlig zufällige Werte der Breiten aus. Wir gehen davon aus, dass dieses neuronale Netzwerk unsere erste zufällige Schätzung für die Q-Funktion ist. Das ist ein bisschen so, als würden Sie die lineare Regression trainieren und alle Parameter zufällig initialisieren und dann den Gradientenabstieg verwenden, um die Parameter zu verbessern. Es ist in Ordnung, es vorerst zufällig zu initialisieren. Wichtig ist, ob der Algorithmus die Parameter langsam verbessern kann, um eine bessere Schätzung zu erhalten. Als nächstes werden wir wiederholt Folgendes tun; Wir werden Aktionen in der Mondlandefähre durchführen, also schwebe zufällig umher, ergreife einige gute Aktionen, ergreife einige schlechte Aktionen. Es ist so oder so in Ordnung. Aber man bekommt viele dieser Tupel, als es in einem Zustand war, man nahm einige davon, sammelte tatsächlich R von S und gelangte zur Primzahl eines Zustandes. Wir werden die 10.000 aktuellsten Beispiele dieser Tupel speichern. Wenn Sie diesen Algorithmus ausführen, werden Sie viele Schritte im Mondlander sehen, vielleicht Hunderttausende von Schritten. Um jedoch sicherzustellen, dass wir am Ende nicht zu viel Computerspeicher verbrauchen, ist es üblich, sich einfach die 10.000 aktuellsten Tupel zu merken, die wir im MTP beim Ausführen von Aktionen gesehen haben. Diese Technik, bei der nur die neuesten Beispiele gespeichert werden, wird im Reinforcement-Learning-Algorithmus manchmal als Wiedergabepuffer bezeichnet. Im Moment fliegen wir die Mondlandefähre einfach zufällig,

manchmal stürzt sie ab, manchmal nicht, und wir erhalten diese Tupel als Erfahrung für unseren Lernalgorithmus. Gelegentlich trainieren wir dann das neuronale Netzwerk. Um das neuronale Netzwerk zu trainieren, gehen wir wie folgt vor. Wir schauen uns diese 10.000 zuletzt gespeicherten Tupel an und erstellen einen Trainingssatz mit 10.000 Beispielen.

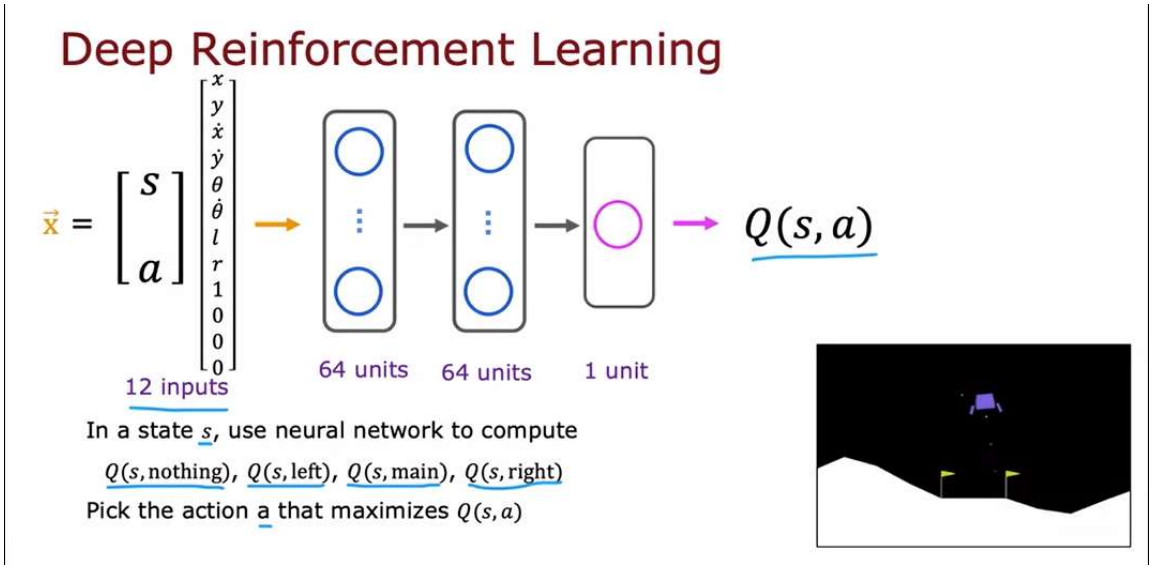


Der Trainingssatz benötigt viele x- und y-Paare. Für unsere Trainingsbeispiele ist x das s, a aus diesem Teil des Tupels. Es wird eine Liste mit 12 Zahlen geben, die 8 Zahlen für den Status und die 4 Zahlen für die One-Hot-Kodierung der Aktion. Der Zielwert, den ein neuronales Netzwerk vorhersagen soll, wäre y gleich R von S plus Gamma max einer Primzahl, Q von s Primzahl einer Primzahl. Wie erhalten wir diesen Wert von Q? Nun, zunächst einmal handelt es sich um dieses neuronale Netzwerk, das wir zufällig initialisiert haben. Es ist vielleicht keine sehr gute Vermutung, aber es ist eine Vermutung. Nachdem wir diese 10.000 Trainingsbeispiele erstellt haben, verfügen wir über die Trainingsbeispiele x1, y1 bis x10.000, y10.000. Wir trainieren ein neuronales Netzwerk und ich werde das neue neuronale Netzwerk Q neu nennen, sodass Q neu von s, a lernt, sich y anzunähern. Dies ist genau das Training dieses neuronalen Netzwerks, um f mit den Parametern w und b auszugeben, x einzugeben und zu versuchen, den Zielwert y anzunähern. Nun sollte dieses neuronale Netzwerk etwas besser abschätzen können, wie die Q-Funktion oder die Zustandsaktionswertfunktion aussehen sollte. Was wir tun werden, ist, dass wir Q nehmen und es auf dieses neue neuronale Netzwerk setzen, das wir gerade gelernt haben. Viele der Ideen in diesem Algorithmus gehen auf Mnih et al. zurück. Es stellt sich heraus, dass, wenn Sie diesen Algorithmus ausführen, bei dem Sie mit einer wirklich zufälligen Schätzung der Q-Funktion beginnen, Sie die Bellman-Gleichungen verwenden, um wiederholt zu versuchen, die Schätzungen der Q-Funktion zu verbessern. Wenn Sie dies dann immer wieder tun, viele Maßnahmen ergreifen und ein Modell trainieren, können Sie Ihre Schätzung für die Q-Funktion verbessern. Für das nächste Modell, das Sie trainieren, haben Sie jetzt eine etwas bessere Schätzung der Q-Funktion. Dann wird das nächste Modell, das Sie trainieren, noch besser sein. Wenn Sie aktualisieren, ist Q gleich Q neu. Wenn Sie dann das nächste Mal ein Modell Q von s Primzahlen trainieren, ist eine Primzahl eine noch bessere Schätzung. Wenn Sie diesen Algorithmus bei jeder Iteration Q von s Primzahl ausführen, wird eine Primzahl hoffentlich zu einer noch besseren Schätzung der Q-Funktion, sodass dies, wenn Sie den Algorithmus lange genug ausführen, tatsächlich zu einer ziemlich guten Schätzung des wahren Werts von Q wird von s, a, so dass man daraus dann hoffentlich gute Aktionen oder den MTP auswählen kann. Der Algorithmus, den Sie gerade gesehen haben, wird manchmal als DQN-Algorithmus bezeichnet, was für Deep Q-Network steht, weil Sie Deep Learning und ein neuronales Netzwerk verwenden, um ein Modell zum Erlernen der Q-Funktionen zu trainieren. Daher DQN oder DQ unter Verwendung eines neuronalen Netzwerks.

Wenn Sie den Algorithmus so verwenden, wie ich ihn beschrieben habe, funktioniert er auf dem Mondlander, okay. Vielleicht dauert die Konvergenz lange, vielleicht landet es nicht perfekt, aber es wird funktionieren. Es stellt sich jedoch heraus, dass der Algorithmus mit ein paar Verfeinerungen viel besser funktionieren kann. Schauen wir uns in den nächsten Videos einige Verbesserungen des Algorithmus an, den Sie gerade gesehen haben.

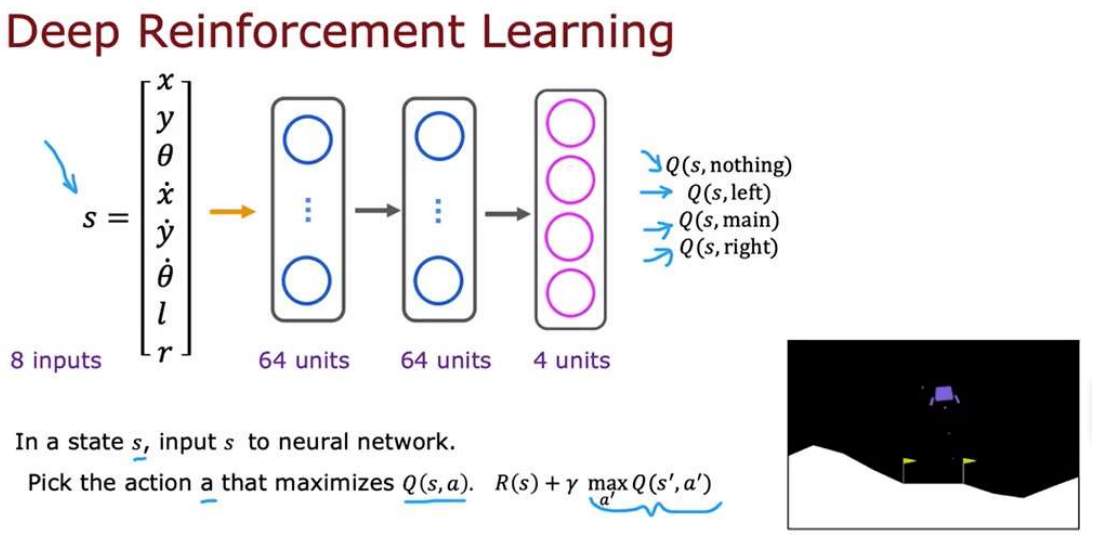
Verfeinerung des Algorithmus: Verbesserte neuronale Netzwerkarchitektur

Im letzten Video haben wir eine neuronale Netzwerkarchitektur gesehen, die den Zustand und die Aktion eingibt und versucht, die Q-Funktion Q von s, a auszugeben. Es stellt sich heraus, dass es eine Änderung in der Architektur des neuronalen Netzwerks gibt, die diesen Algorithmus wesentlich effizienter macht. Die meisten Implementierungen von DQN nutzen tatsächlich diese effizientere Architektur, die wir in diesem Video sehen werden. Lass uns einen Blick darauf werfen. Dies war die neuronale Netzwerkarchitektur, die wir zuvor gesehen haben, bei der 12 Zahlen eingegeben und Q von s, a ausgegeben wurden.



Immer wenn wir uns in einem Zustand s befinden, müssten wir im neuronalen Netzwerk viermal separat eine Inferenz durchführen, um diese vier Werte zu berechnen und so die Aktion a auszuwählen, die uns den größten Q-Wert liefert. Dies ist ineffizient, da wir unsere Schlussfolgerung aus jedem einzelnen Zustand viermal ziehen müssen. Stattdessen erweist es sich als effizienter, ein einzelnes neuronales Netzwerk so zu trainieren, dass es alle vier dieser Werte gleichzeitig ausgibt. So sieht es aus. Hier ist eine modifizierte neuronale Netzwerkarchitektur, bei der die Eingabe aus acht Zahlen besteht, die dem Zustand der Mondlandefähre entsprechen. Anschließend durchläuft es das neuronale Netzwerk mit 64 Einheiten in der ersten verborgenen Schicht und 64 Einheiten in der zweiten verborgenen Schicht. Jetzt verfügt die Ausgabeeinheit über vier Ausgabeeinheiten, und die Aufgabe des neuronalen Netzwerks besteht darin, dass die vier Ausgabeeinheiten Q von s, nichts, Q von s, links, Q von s, Haupt und q von s, rechts ausgeben. Die Aufgabe des neuronalen Netzwerks besteht darin, gleichzeitig den Q-Wert für alle vier möglichen Aktionen zu berechnen, wenn wir uns im Zustand s befinden. Dies erweist sich als effizienter, da wir bei gegebenem Zustand s die Inferenz nur einmal ausführen und alle vier dieser Werte erhalten und dann sehr schnell die Aktion a auswählen können, die Q von s, a, maximiert. Sie bemerken auch, dass es in Bellmans Gleichungen einen Schritt gibt, in dem wir max über eine Primzahl Q von s berechnen müssen. Primzahl eine Primzahl, dies maximiert ein Gamma und dann gab es hier oben plus R von s. Dieses duale Netzwerk macht es auch viel effizienter, dies zu berechnen, da wir Q von s Primzahlen erhalten, eine Primzahl

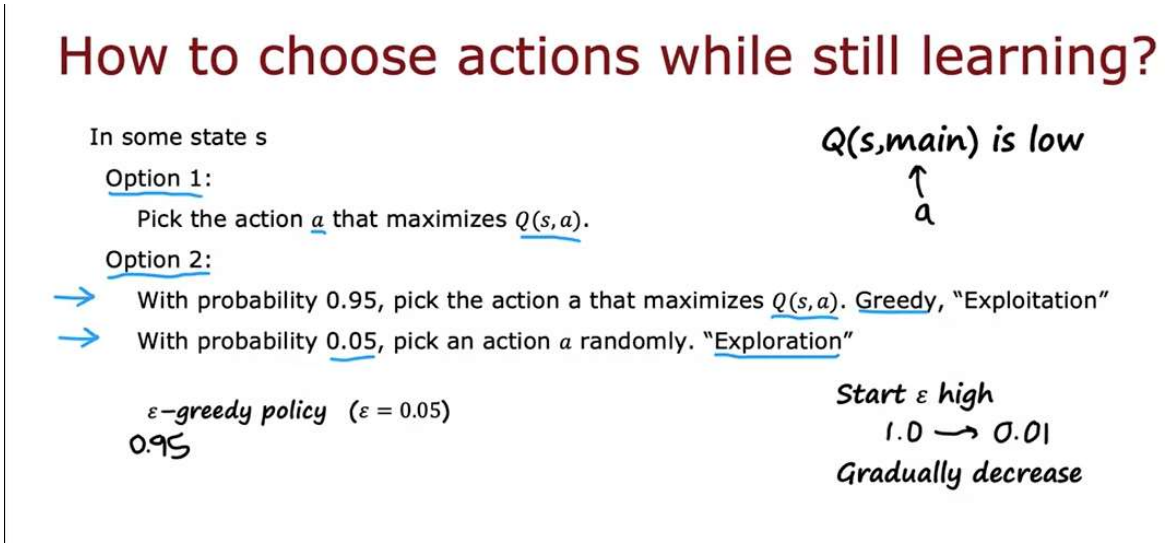
für alle Aktionen gleichzeitig eine Primzahl. Sie können dann einfach den Maximalwert auswählen, um diesen Wert für die rechte Seite der Bellman-Gleichungen zu berechnen.



Diese Änderung an der neuronalen Netzwerkarchitektur macht das RN viel effizienter, und daher werden wir diese Architektur im Übungslabor verwenden. Als nächstes gibt es noch eine weitere Idee, die dem Algorithmus sehr helfen wird, nämlich eine sogenannte Epsilon-Greedy-Richtlinie, die sich darauf auswirkt, wie Sie Aktionen auswählen, selbst während Sie noch lernen. Werfen wir einen Blick auf das nächste Video und was das bedeutet.

Verfeinerung des Algorithmus: ϵ -gierige Politik

Mit dem Lernalgorithmus, den wir entwickelt haben, müssen Sie, auch wenn Sie noch lernen, Q(s,a) zu approximieren, einige Aktionen im Mondlander durchführen. Wie wählen Sie diese Aktionen aus, während Sie noch lernen? Der gebräuchlichste Weg hierfür ist die Verwendung einer sogenannten Epsilon-Greedy-Richtlinie. Werfen wir einen Blick darauf, wie das funktioniert. Hier ist der Algorithmus, den Sie zuvor gesehen haben. Einer der Schritte im Algorithmus besteht darin, Aktionen im Mondlander durchzuführen.



Wenn der Lernalgorithmus noch läuft, wissen wir nicht wirklich, welche Maßnahmen in jedem Zustand die beste sind. Wenn wir das täten, wären wir bereits mit dem Lernen fertig. Aber selbst wenn wir noch lernen und noch keine sehr gute Schätzung von Q(s,a) haben, wie ergreifen wir in

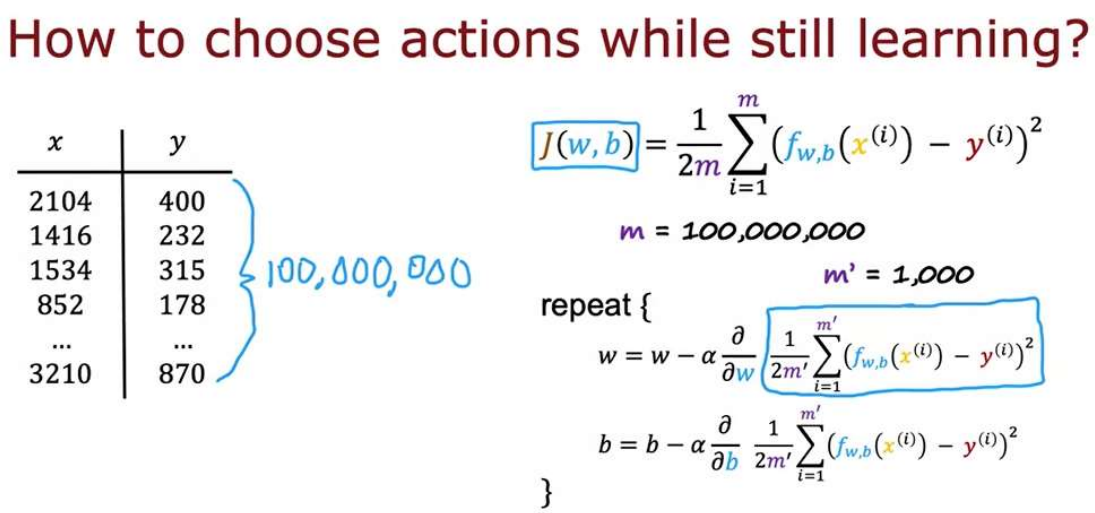
diesem Schritt des Lernalgorithmus Maßnahmen? Schauen wir uns einige Optionen an. Wenn Sie sich in bestimmten BundesZustanden befinden, möchten wir möglicherweise keine völlig willkürlichen Maßnahmen ergreifen, da dies oft eine schlechte Maßnahme ist. Eine natürliche Option wäre, immer im Zustand s eine Aktion a auszuwählen, die Q(s,a) maximiert. Wir können sagen: Auch wenn Q(s,a) keine gute Schätzung der Q-Funktion ist, geben wir einfach unser Bestes und verwenden unsere aktuelle Schätzung von Q(s,a) und wählen die Aktion a aus, die sie maximiert. Es stellt sich heraus, dass dies zwar gut funktioniert, aber nicht die beste Option ist. Stattdessen erfahren Sie hier, was üblicherweise getan wird. Hier ist Option 2, die in den meisten Fällen, sagen wir mit einer Wahrscheinlichkeit von 0,95, darin besteht, die Aktion auszuwählen, die Q(s,a) maximiert. Meistens versuchen wir, anhand unserer aktuellen Schätzung von Q(s,a) eine gute Aktion auszuwählen. Aber in einem kleinen Bruchteil der Fälle, sagen wir fünf Prozent, wählen wir eine Aktion zufällig aus. Warum wollen wir gelegentlich eine Aktion zufällig auswählen? Nun, hier ist der Grund. Angenommen, es gibt einen seltsamen Grund dafür, dass Q(s,a) zufällig initialisiert wurde, sodass der Lernalgorithmus denkt, dass das Zünden des Haupttriebwerks niemals eine gute Idee ist. Möglicherweise wurden die Parameter des neuronalen Netzwerks so initialisiert, dass Q(s, main) immer sehr niedrig ist. Wenn das der Fall ist, wird das neuronale Netzwerk niemals versuchen, das Haupttriebwerk zu zünden, da es versucht, die Aktion a auszuwählen, die Q(s,a) maximiert. Da es niemals versucht, das Hauptstrahlruder zu zünden, wird es auch nie lernen, dass das Zünden des Hauptstrahlruders manchmal tatsächlich eine gute Idee ist. Wenn das neuronale Netzwerk aufgrund der zufälligen Initialisierung anfangs irgendwie in dem Bewusstsein stecken bleibt, dass einige Dinge einfach zufällig eine schlechte Idee sind, dann bedeutet Option 1, dass es diese Aktionen niemals ausprobiert und herausfindet, dass vielleicht tatsächlich eine gute Idee ist Idee, diese Aktion zu ergreifen, wie zum Beispiel manchmal die Haupttriebwerke abzufeuern. Bei Option 2 haben wir bei jedem Schritt eine geringe Wahrscheinlichkeit, verschiedene Aktionen auszuprobieren, damit das neuronale Netzwerk lernen kann, seine eigenen möglichen Vorurteile darüber zu überwinden, was eine schlechte Idee sein könnte, die sich aber als nicht der Fall erweist. Diese Idee, Aktionen zufällig auszuwählen, wird manchmal als Erkundungsschritt bezeichnet. Denn wir werden etwas ausprobieren, was vielleicht nicht die beste Idee ist, aber wir werden einfach eine Aktion unter bestimmten Umständen ausprobieren, eine Aktion in einer Situation erforschen und mehr darüber erfahren, in der wir vielleicht nicht so viel hatten Erfahrung vor. Wenn wir eine Aktion ausführen, die Q(s,a) maximiert, wird dies manchmal als gierige Aktion bezeichnet, da wir versuchen, unsere Rendite dadurch tatsächlich zu maximieren. Oder in der Literatur zum Reinforcement Learning hört man dies manchmal auch als Ausbeutungsschritt. Ich weiß, dass Ausbeutung keine gute Sache ist, niemand sollte jemals jemand anderen erforschen. Aber historisch gesehen war dies der Begriff, der beim verstärkenden Lernen verwendet wurde, um zu sagen: Nutzen wir alles, was wir gelernt haben, um das Beste zu geben, was wir können. In der Fachliteratur zum Reinforcement Learning hört man manchmal Leute über den Kompromiss zwischen Erkundung und Ausbeutung sprechen. Dabei geht es darum, wie oft man willkürlich oder vielleicht nicht optimal handelt, um mehr zu lernen, statt zu versuchen, das eigene Potenzial zu maximieren Kehren Sie zurück, indem Sie beispielsweise die Aktion ausführen, die Q (s,a) maximiert. Dieser Ansatz, also Option 2, hat einen Namen und wird als Epsilon-Greedy-Richtlinie bezeichnet, wobei Epsilon hier 0,05 beträgt und die Wahrscheinlichkeit ist, eine Aktion zufällig auszuwählen. Dies ist die gebräuchlichste Methode, um Ihren Reinforcement-Learning-Algorithmus ein wenig erforschen zu lassen, auch wenn er gelegentlich oder vielleicht die meiste Zeit gierige Aktionen ausführt. Übrigens haben viele Leute angemerkt, dass der Name „Epsilon-Gier-Politik“ verwirrend ist, weil man tatsächlich in 95 Prozent der Fälle gierig ist, nicht in fünf Prozent der Fälle. Also vielleicht 1 minus Epsilon-Gier-Richtlinie, weil sie zu 95 Prozent gierig und zu 5 Prozent explorativ ist, das ist tatsächlich eine genauere Beschreibung des Algorithmus. Aber aus historischen Gründen ist der Name „Epsilon-gierige Politik“ hängengeblieben. Dies ist der Name, den die Leute verwenden, um sich auf die Richtlinie zu beziehen, die tatsächlich Epsilon-Bruchteile der

Zeit erforscht, und nicht diesen gierigen Epsilon-Bruchteil der Zeit. Schließlich besteht einer der Tricks, die manchmal beim verstärkenden Lernen angewendet werden, darin, mit einem hohen Epsilon-Wert zu beginnen. Anfangs führen Sie viele zufällige Aktionen auf einmal aus und verringern diese dann schrittweise, sodass Sie mit der Zeit weniger wahrscheinlich zufällige Aktionen ausführen und eher Ihre sich verbessernden Schätzungen der Q-Funktion nutzen, um gute Aktionen auszuwählen. Beispielsweise könnten Sie bei der Übung zur Mondlandefähre mit einem sehr, sehr hohen Epsilon beginnen, vielleicht entspricht Epsilon sogar 1,0. Sie wählen die Aktionen zunächst völlig zufällig aus und verringern den Wert dann schrittweise bis auf beispielsweise 0,01, sodass Sie schließlich in 99 Prozent der Fälle gierige Aktionen ausführen und nur in einem sehr kleinen Prozent der Fälle zufällig handeln. Wenn dies kompliziert erscheint, machen Sie sich darüber keine Sorgen. Wir stellen Ihnen im Jupiter-Labor den Code zur Verfügung, der Ihnen zeigt, wie das geht. Wenn Sie den Algorithmus, wie wir ihn beschrieben haben, mit der effizienteren neuronalen Netzwerkarchitektur und einer Epsilon-gierigen Explorationsrichtlinie implementieren würden, würden Sie feststellen, dass sie auf dem Mondlander ziemlich gut funktionieren. Eines der Dinge, die mir beim Reinforcement-Learning-Algorithmus aufgefallen sind, ist, dass sie im Vergleich zum überwachten Lernen hinsichtlich der Auswahl der Hyperparameter heikler sind. Wenn Sie beispielsweise beim überwachten Lernen die Lernrate etwas zu niedrig einstellen, dauert das Lernen des Algorithmus möglicherweise länger. Vielleicht dauert das Training dreimal so lange, was ärgerlich ist, aber vielleicht nicht so schlimm. Beim verstärkenden Lernen stellen Sie jedoch fest, dass das Lernen nicht dreimal so lange dauert, wenn Sie den Wert von Epsilon nicht ganz so gut einstellen oder andere Parameter nicht ganz so gut einstellen. Das Erlernen kann zehnmal oder hundertmal so lange dauern. Ich denke, dass Reinforcement-Learning- Algorithmen, weil sie weniger ausgereift sind als Supervised-Learning-Algorithmen, bei der Auswahl solcher Parameter viel heikler sind, und ehrlich gesagt ist es manchmal tatsächlich frustrierender, diese Parameter mit Reinforcement-Learning-Algorithmen abzustimmen als mit einem Supervised-Algorithmus Lernalgorithmus. Aber noch einmal: Wenn Sie sich wegen des Übungslabors, der Programmübung, Sorgen machen, geben wir Ihnen einen Eindruck von guten Parametern, die Sie in der Programmübung verwenden können, damit Sie hoffentlich in der Lage sein sollten, dies zu tun und den Mondlander erfolgreich zu erlernen ohne allzu viele Probleme. Im nächsten optionalen Video möchte ich, dass wir ein paar weitere Algorithmusverfeinerungen, Mini-Batching und auch die Verwendung von Soft-Updates vorantreiben. Auch ohne diese zusätzlichen Verfeinerungen funktioniert der Algorithmus einwandfrei, es handelt sich jedoch um zusätzliche Verfeinerungen, die dafür sorgen, dass der Algorithmus viel schneller läuft. Es ist in Ordnung, wenn Sie dieses Video überspringen. Wir haben im Übungslabor alles bereitgestellt, was Sie brauchen, um es hoffentlich erfolgreich abzuschließen. Aber wenn Sie daran interessiert sind, mehr über diese Details zweier benannter Reinforcement-Learning-Algorithmen zu erfahren, dann kommen Sie mit und sehen wir uns im nächsten Video Mini-Batching und Soft-Updates an.

Verfeinerung des Algorithmus: Mini-Batch und Soft-Updates (optional)

In diesem Video sehen wir uns zwei weitere Verfeinerungen des Reinforcement-Learning-Algorithmus an, den Sie gesehen haben. Die erste Idee nennt sich die Verwendung von Mini-Batches. Es stellt sich heraus, dass diese Idee sowohl Ihren Reinforcement-Learning-Algorithmus beschleunigen kann als auch auf überwachtes Lernen anwendbar ist. Sie können Ihnen auch dabei helfen, Ihren überwachten Lernalgorithmus zu beschleunigen, z. B. beim Training eines neuronalen Netzwerks oder beim Training einer linearen Regression oder eines logistischen Regressionsmodells. Die zweite Idee, die wir uns ansehen werden, sind Soft-Updates, die, wie sich herausstellt, Ihrem Reinforcement-Learning-Algorithmus dabei helfen werden, bessere Arbeit zu leisten und eine gute Lösung zu finden. Werfen wir einen Blick auf Mini-Batches und Soft-Updates. Um Mini-Batches zu verstehen, werfen wir zunächst einen Blick auf das überwachte Lernen. Hier ist der Datensatz zu

Wohnungsgrößen und -preisen, den Sie bereits im ersten Kurs dieser Spezialisierung zur Verwendung der linearen Regression zur Vorhersage von Immobilienpreisen gesehen haben. Dort hatten wir diese Kostenfunktion für die Parameter w und b erstellt, sie war 1 über 2m, Summe der Vorhersage minus dem tatsächlichen Wert y^2. Der Gradient in diesem Algorithmus bestand darin, w wiederholt zu aktualisieren, indem w minus [unverständliches] Alpha mal der partiellen Ableitung der Kosten J von wb nach w, und b in ähnlicher Weise wie folgt zu aktualisieren. Lassen Sie mich einfach diese Definition von J von wb übernehmen und sie hier ersetzen. Als wir uns dieses Beispiel ansahen, als wir vor langer Zeit anfingen, über lineare Regression und überwachtes Lernen zu sprechen, war die Größe des Trainingssatzes m ziemlich klein. Ich glaube, wir hatten 47 Trainingsbeispiele. Aber was ist, wenn Sie ein sehr großes Trainingsset haben? Angenommen, m entspricht 100 Millionen. Es gibt viele Länder, einschließlich der Vereinigten Zustanden, mit über 100 Millionen Wohneinheiten, und eine nationale Volkszählung wird Ihnen einen Datensatz dieser Größenordnung oder Größe liefern. Das Problem bei diesem Algorithmus besteht bei einem so großen Datensatz darin, dass jeder einzelne Schritt des Gradientenabstiegs die Berechnung dieses Durchschnitts über 100 Millionen Beispiele erfordert, was sich als sehr langsam herausstellt. Jeder Schritt des Gradientenabstiegs bedeutet, dass Sie diese Summe oder diesen Durchschnitt über 100 Millionen Beispiele berechnen würden.



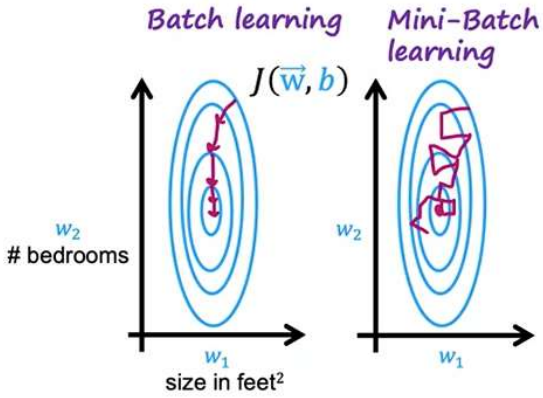
Dann machen Sie einen kleinen Gradientenabstiegsschritt und gehen zurück und müssen Ihren gesamten 100-Millionen-Beispieldatensatz erneut scannen, um die Ableitung im nächsten Schritt zu berechnen. Sie machen einen weiteren kleinen Gradientenabstiegsschritt und so weiter und so fort. Wenn die Größe des Trainingssatzes sehr groß ist, erweist sich dieser Gradientenabstiegsalgorithmus als recht langsam. Die Idee des Mini-Batch-Gradientenabstiegs besteht darin, nicht alle 100 Millionen Trainingsbeispiele bei jeder einzelnen Iteration durch diese Schleife zu verwenden. Stattdessen können wir eine kleinere Zahl wählen, ich nenne sie m prim gleich, sagen wir, 1.000. Anstatt bei jedem Schritt alle 100 Millionen Beispiele zu verwenden, würden wir eine Teilmenge von 1.000 oder m Primbeispielen auswählen. Dieser innere Term wird 1 über 2m Primzahl ist Summe über Summe m Primzahlbeispiele. Jetzt erfordert jede Iteration durch den Gradientenabstieg nur die Betrachtung der 1.000 statt 100 Millionen Beispiele, und jeder Schritt nimmt viel weniger Zeit in Anspruch und führt einfach zu einem effizienteren Algorithmus. Was der Mini-Batch-Gradientenabstieg bewirkt, ist die erste Iteration durch den Algorithmus, möglicherweise wird diese Teilmenge der Daten betrachtet. Bei der nächsten Iteration wird möglicherweise diese Teilmenge der Daten untersucht und so weiter. Für die dritte Iteration und so weiter, sodass jede Iteration nur eine Teilmenge der Daten betrachtet, sodass jede Iteration viel schneller ausgeführt wird. Um zu sehen, warum dies ein sinnvoller Algorithmus sein könnte, finden Sie hier den Wohnungsdatensatz. Wenn wir uns bei der

ersten Iteration nur, sagen wir, fünf Beispiele ansehen würden, wäre dies nicht der gesamte Datensatz, aber er ist ein wenig repräsentativ für die Zeichenfolgenlinie, die Sie am Ende möglicherweise anpassen möchten, und macht daher einen Gradientenabstiegsschritt erforderlich, um den Algorithmus zu verbessern Passt diese fünf Beispiele, ist das in Ordnung. Aber dann nehmen Sie bei der nächsten Iteration andere fünf Beispiele wie das hier gezeigte. Sie machen einen Gradientenabstiegsschritt mit diesen fünf Beispielen, und bei der nächsten Iteration verwenden Sie andere fünf Beispiele und so weiter und so weiter.

Sie können diese Beispielliste von oben nach unten durchgehen. Das wäre eine Möglichkeit. Eine andere Möglichkeit wäre, bei jeder einzelnen Iteration einfach fünf völlig andere Beispiele zur Verwendung auszuwählen. Sie erinnern sich vielleicht an den Batch-Gradientenabstieg, wenn dies die Konturen der Kostenfunktion J sind. Dann würde der Batch-Gradientenabstieg sagen: Beginnen Sie hier und machen Sie einen Schritt, machen Sie einen Schritt, machen Sie einen Schritt, machen Sie einen Schritt, machen Sie einen Schritt. Jeder Schritt des Gradientenabstiegs führt dazu, dass sich die Parameter hier in der Mitte zuverlässig dem globalen Minimum der Kostenfunktion nähern. Im Gegensatz dazu bewirken ein Mini-Batch-Gradientenabstieg oder ein Mini-Batch-Lernalgorithmus so etwas. Wenn Sie hier beginnen, werden in der ersten Iteration nur fünf Beispiele verwendet. Es wird in die richtige Richtung treffen, aber möglicherweise nicht in die beste Richtung für den Gefälleabstieg. Bei der nächsten Iteration machen sie vielleicht das, bei der nächsten Iteration das und das, und manchmal ist es einfach Zufall, dass die fünf Beispiele, die Sie ausgewählt haben, eine unglückliche Wahl sind und sogar in die falsche Richtung gehen, weg vom globalen Minimum, und so weiter und so weiter her. Aber im Durchschnitt tendiert der Mini-Batch-Gradientenabstieg in Richtung des globalen Minimums, was nicht zuverlässig und etwas laut ist, aber jede Iteration ist viel rechenintensiver, und so erweist sich das Mini-Batch-Lernen oder der Mini-Batch-Gradientenabstieg als viel schnellerer Algorithmus wenn Sie einen sehr großen Trainingssatz haben.

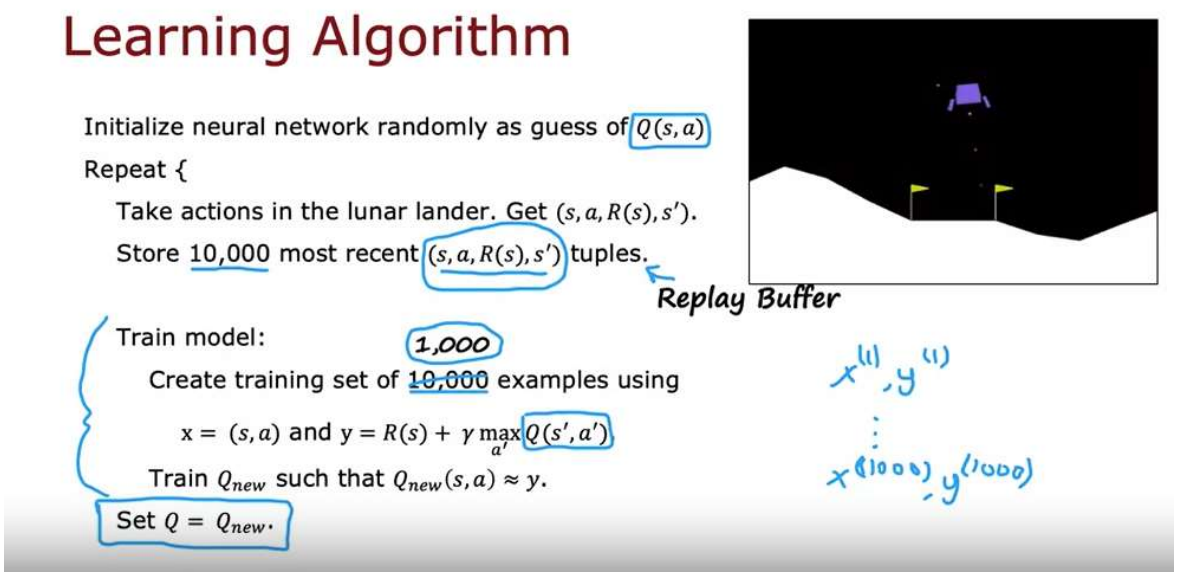
Mini-batch

x	y
2104	400
1416	232
1534	315
852	178
...	...
3210	870

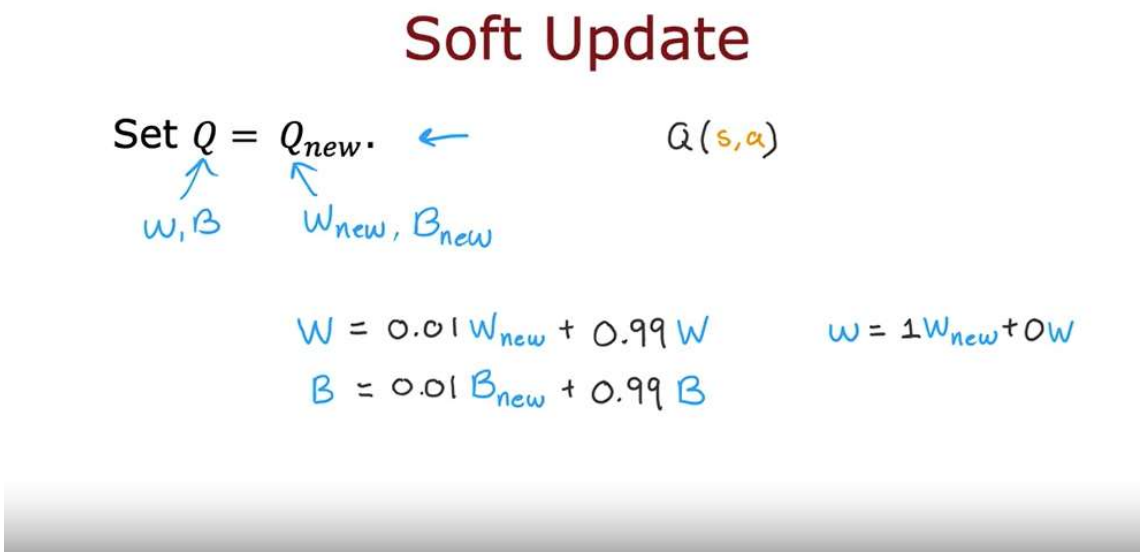


Tatsächlich wird beim überwachten Lernen, bei dem Sie über einen sehr großen Trainingssatz verfügen, häufiger Mini-Batch-Lernen oder Mini-Batch-Gradientenabstieg oder eine Mini-Batch-Version mit anderen Optimierungsalgorithmen wie Atom verwendet als Batch-Gradientenabstieg. Zurück zu unserem Reinforcement-Learning-Algorithmus: Dies ist der Algorithmus, den wir zuvor gesehen haben. Die Mini-Batch-Version davon würde lauten: Selbst wenn Sie die 10.000 aktuellsten Tupel im Wiedergabepuffer gespeichert haben, könnten Sie sich dafür entscheiden, nicht jedes Mal, wenn Sie ein Modell trainieren, alle 10.000 zu verwenden. Stattdessen könnten Sie einfach die Teilmenge nehmen. Sie könnten nur 1.000 Beispiele dieser s, a, R von s, s-Primärtupeln auswählen und damit nur 1.000 Trainingsbeispiele zum Trainieren des neuronalen Netzwerks erstellen. Es stellt sich heraus, dass dadurch jede Iteration des Trainierens eines Modells etwas lauter, aber viel schneller wird, was insgesamt dazu führt, dass dieser Verstärkungslernalgorithmus beschleunigt wird. Auf diese Weise kann Mini-Batching sowohl einen überwachten Lernalgorithmus wie die lineare Regression als auch diesen Reinforcement-Learning-Algorithmus beschleunigen, bei dem Sie eine Mini-Batch-Größe von beispielsweise 1.000 Beispielen verwenden können, selbst wenn Sie 10.000 dieser Tupel darin speichern Ihr Wiedergabepuffer. Schließlich gibt es noch eine weitere Verfeinerung des Algorithmus, die ihn zuverlässiger konvergieren lässt: Ich habe hier diesen Schritt von „Setze Q gleich Q_new“ beschrieben. Es stellt sich jedoch heraus, dass dies zu einer sehr abrupten Änderung von Q führen kann. Wenn Sie ein neues neuronales Netzwerk auf ein neues trainieren, handelt es sich möglicherweise nur zufällig um kein sehr gutes neuronales Netzwerk. Vielleicht ist es sogar ein bisschen schlechter als das alte, dann haben Sie einfach Ihre Q-Funktion mit einem potenziell schlimmeren verrauschten neuronalen Netzwerk überschrieben. Die Soft-Update-Methode hilft zu verhindern, dass sich Q_new durch nur einen einzigen unglücklichen Schritt verschlechtert. Insbesondere verfügt das neuronale Netzwerk Q über einige Parameter, W und B, alle Parameter für alle Schichten im neuronalen Netzwerk. Wenn Sie das neue neuronale Netzwerk trainieren, erhalten Sie einige Parameter W_new und B_new. Im ursprünglichen Algorithmus S [unverständlich] auf dieser Folie würden Sie W gleich W_new und B gleich B_new setzen. Das ist es, was die Menge Q gleich Q_new bedeutet. Mit dem Soft-Update setzen wir stattdessen W gleich 0,01 mal W_new plus 0,99 mal W. Mit anderen Worten, wir machen W zu 99 Prozent der alten Version von W plus einem Prozent der neuen Version W_new. Dies wird als Soft-Update bezeichnet, da wir jedes Mal, wenn wir ein neues neuronales Netzwerk W_new trainieren, nur einen kleinen Teil des neuen Werts akzeptieren. Ebenso ist B gleich 0,01 mal B_new plus 0,99 mal B. Diese Zahlen, 0,01 und 0,99, sind Hyperparameter, die Sie festlegen können, aber sie steuern, wie aggressiv Sie W nach

W_new verschieben, und es wird erwartet, dass diese beiden Zahlen eins ergeben. Ein Extrem wäre, wenn Sie W gleich eins mal W_new plus 0 mal W setzen würden.



In diesem Fall kehren Sie zum ursprünglichen Algorithmus zurück, bei dem Sie nur W_new auf W kopieren. Ein Soft-Update ermöglicht Ihnen dies jedoch Nehmen Sie eine allmählichere Änderung an Q oder an den neuronalen Netzwerkparametern W und B vor, die sich auf Ihre aktuelle Schätzung für die Q-Funktion Q von s, a auswirken. Es stellt sich heraus, dass die Verwendung der Soft-Update-Methode dazu führt, dass der Reinforcement-Learning-Algorithmus zuverlässiger konvergiert.



Dadurch ist es weniger wahrscheinlich, dass der Reinforcement-Learning-Algorithmus schwankt oder abweicht oder andere unerwünschte Eigenschaften aufweist. Mit diesen beiden letzten Verfeinerungen des Algorithmus, dem Mini-Batching, das sich tatsächlich auch sehr gut für die Überwachung des Lernens und nicht nur für das verstärkende Lernen eignet, sowie der Idee von Soft Updates, sollten Sie in der Lage sein, Ihren Mondalgorithmus wirklich zum Laufen zu bringen gut auf dem Mondlandegerät. Der Lunar Lander ist tatsächlich eine ziemlich komplexe, ziemlich herausfordernde Anwendung, sodass Sie ihn zum Laufen bringen und sicher auf dem Mond landen können. Ich finde das wirklich cool und wünsche euch viel Spaß beim Spielen mit dem Übungslabor. Nun haben wir viel über Reinforcement Learning gesprochen. Bevor wir zum Abschluss kommen, möchte ich Ihnen meine Gedanken zum Stand des verstärkenden Lernens mitteilen, damit Sie beim

Erstellen von Anwendungen mithilfe verschiedener Techniken des maschinellen Lernens über überwachte und unüberwachte Techniken des verstärkenden Lernens einen Rahmen haben, den Sie verstehen können wo Reinforcement Learning heute in die Welt des maschinellen Lernens passt. Schauen wir uns das im nächsten Video an.

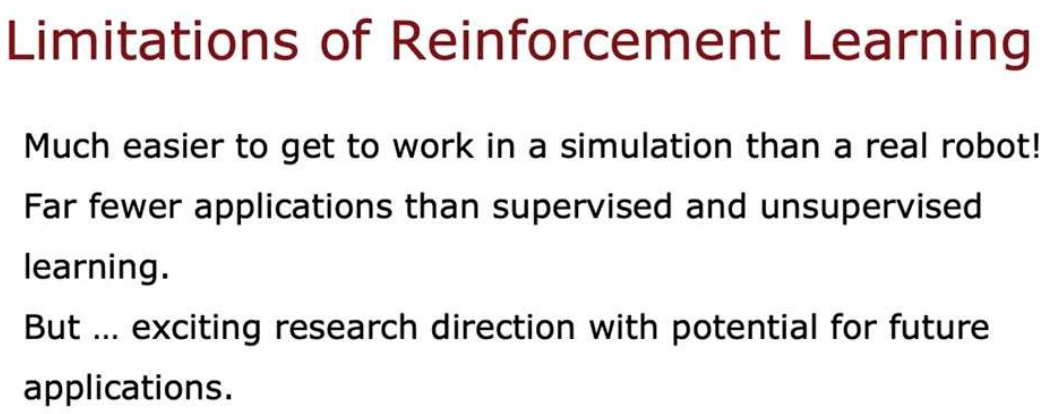
Der Stand des verstärkenden Lernens

Reinforcement Learning ist eine spannende Reihe von Technologien. Als ich an meiner Doktorarbeit arbeitete, war Reinforcement Learning tatsächlich das Thema meiner Dissertation. Daher war und bin ich von diesen Ideen begeistert. Trotz der ganzen Forschungsdynamik und Aufregung, die hinter Reinforcement Learning steckt, glaube ich, dass es einen gewissen oder vielleicht manchmal sogar großen Hype um das Thema gibt. Ich hoffe, Ihnen ein praktisches Gefühl dafür vermitteln zu können, wo Reinforcement Learning heute im Hinblick auf seinen Nutzen für Anwendungen steht. Einer der Gründe für den Hype um Reinforcement Learning liegt darin, dass sich viele der Forschungspublikationen offenbar mit simulierten Umgebungen befassen. Und da ich selbst sowohl in Simulationen als auch an echten Robotern gearbeitet habe, kann ich Ihnen sagen, dass es viel einfacher ist, ein Album zum Reinforcement Learning in einer Simulation oder einem Videospiel zum Laufen zu bringen als in einem echten Roboter. Daher haben viele Entwickler angemerkt, dass es sich selbst nachdem sie es in der Simulation zum Laufen gebracht hatten, als überraschend schwierig herausstellte, etwas in der realen Welt oder am echten Roboter zum Laufen zu bringen. Wenn Sie diese Algorithmen also auf eine reale Anwendung anwenden, ist dies eine Einschränkung, die Sie hoffentlich beachten, um sicherzustellen, dass das, was Sie tun, bei der realen Anwendung funktioniert. Zweitens gibt es trotz aller Berichterstattung in den Medien über Reinforcement Learning heute weitaus weniger Anwendungen von Reinforcement Learning als überwachtes und unüberwachtes Lernen. Wenn Sie also eine praktische Anwendung erstellen, ist die Wahrscheinlichkeit, dass Sie überwachtes oder unüberwachtes Lernen nützlich oder das richtige Werkzeug für den Job finden, viel höher als die Wahrscheinlichkeit, dass Sie am Ende verstärkendes Lernen verwenden würden. Ich habe selbst ein paar Mal Verstärkungslernen eingesetzt, insbesondere bei Robotersteuerungsanwendungen, aber in meiner alltäglichen angewandten Arbeit nutze ich Supervising und überwachtes Lernen viel häufiger. Derzeit gibt es viele spannende Forschungsarbeiten zum Thema Reinforcement Learning, und ich denke, dass das Potenzial des Reinforcement Learning für zukünftige Anwendungen sehr groß ist. Und Reinforcement Learning bleibt nach wie vor eine der wichtigsten Säulen des maschinellen Lernens. Wenn Sie es also als Rahmen für die Entwicklung Ihrer eigenen Algorithmen für maschinelles Lernen nutzen, hoffe ich, dass Sie damit auch beim Aufbau funktionierender Systeme für maschinelles Lernen effektiver werden. Ich hoffe also, dass Ihnen die Materialien dieser Woche zum Thema Verstärkungslernen gefallen haben, und insbesondere wünsche ich Ihnen viel Spaß dabei, die Mondlandefähre selbst zum Landen zu bringen. Ich hoffe, dass es eine zufriedenstellende Erfahrung sein wird, wenn Sie einen Algorithmus implementieren und dann sehen, wie die Mondlandefähre aufgrund des von Ihnen geschriebenen Codes sicher auf dem Mond landet. Damit sind wir am Ende dieser Spezialisierung angelangt. Kommen wir zum letzten Video, in dem wir abschließen

Zusammenfassung und vielen Dank

Willkommen zum Abschlussvideo dieser Spezialisierung auf maschinelles Lernen. Wir haben viele Videos zusammen gesehen und dieses ist das letzte. Lassen Sie uns die Hauptthemen zusammenfassen, die wir besprochen haben, und dann sage ich gerne ein paar Worte und dann schließen wir den Unterricht ab. Rückblickend denke ich, dass wir viel gemeinsam durchgemacht haben. Der erste Kurs, den wir absolvierten, befasste sich mit überwachtem maschinellem Lernen, einschließlich Regression und Klassifizierung. Hier haben Sie etwas über lineare Regression,

logistische Regression, Kostenfunktionen und den Gradientenabstiegsalgorithmus gelernt. Im zweiten Kurs haben wir uns dann mit fortgeschritteneren Lernalgorithmen befasst, darunter neuronale Netze und Entscheidungsbaum-Ensembles, und sind auch Ratschläge für maschinelles Lernen wie Bias und Varianz sowie die Verwendung einer Train-Hold-Out-Kreuzvalidierung und Testsätze usw. durchgegangen wie Sie Ihren Lernalgorithmus effizient verbessern können. Dann ging es im dritten Kurs um unbeaufsichtigte Lernempfehlungen und Reinforcement Learning, wo wir über Clustering-Algorithmen, Anomalie-Erkennungsalgorithmen, kollaborative Filterung und inhaltsbasierte Filterung sprachen, und in der vergangenen Woche dann über Reinforcement Learning. Mit diesem breiten Werkzeugsatz sind Sie nun gut qualifiziert, eine Vielzahl möglicher Anwendungen für maschinelles Lernen zu entwickeln.



Herzlichen Glückwunsch, dass du es bis zu diesem letzten Video geschafft hast. Wenn Sie diese Spezialisierung vollständig durchgearbeitet haben, verfügen Sie nun über eine sehr solide Grundlage im maschinellen Lernen. Ich denke, Sie haben einen guten Start hingelegt, um ein Experte für maschinelles Lernen zu werden. Wie Sie wissen, hat maschinelles Lernen als leistungsstarkes Werkzeug, das täglich von Milliarden Menschen durch Websuche, Produktempfehlungen, Spracherkennung und viele andere Anwendungen genutzt wird, enorme Auswirkungen auf die Gesellschaft. Man hofft sogar, das menschliche Wissen zu verbessern, indem man hofft, dass wissenschaftliche Entdeckungen einen Mehrwert in Milliardenhöhe schaffen und neue Anwendungen ermöglichen, die noch vor wenigen Jahren undenkbar waren. Aber ich denke, die beste Anwendung des maschinellen Lernens muss noch erfunden werden, und das bringt uns zu Ihnen. Sie sind nun gut qualifiziert, die Werkzeuge des maschinellen Lernens zu nutzen, selbst Anwendungen zu erstellen und Dinge zu erschaffen. Ich hoffe, dass Sie diese Fähigkeiten nutzen, um das Leben anderer Menschen zu verbessern. Bevor ich diesen Kurs abschließe, möchte ich Ihnen noch etwas sagen. Es hat mir Spaß gemacht, diesen Kurs zu unterrichten, aber vor nicht allzu langer Zeit war ich selbst Student und weiß daher, wie zeitaufwändig es ist, dieses Zeug zu lernen. Ich weiß, dass Sie ein vielbeschäftigter Mensch sind und viele andere Dinge in Ihrem Leben passieren, also haben Sie sich die Zeit genommen, sich die Videos anzusehen, die Tests und Übungen durchzugehen. Ich weiß, dass Sie viel Zeit und viel von sich selbst in diesen Kurs gesteckt haben. Ich möchte Ihnen nur ganz herzlich dafür danken, dass Sie in diesem Kurs studiert haben. Ich bin Ihnen sehr dankbar und schätze die Zeit, die Sie mit mir und der Spezialisierung verbracht haben. Danke schön.

Andrew Ng und Chelsea Finn über KI und Robotik

Vielen Dank, dass Sie hier an der Stanford University zu mir gekommen sind, mit Professor Chelsea Finn, die sowohl in der Informatik- als auch in der Elektrotechnikabteilung tätig ist. Chelsea ist ein Forschungslabor, das Spitzenarbeit leistet und maschinelles Lernen, insbesondere Reinforcement

Learning, auf eine Reihe von Robotikanwendungen anwendet. Ich bin wirklich froh, Sie hier zu haben, Chelsea. >> Ich freue mich, hier zu sein. >> Ich weiß also, dass Sie, bevor Sie sich für die Informatik und maschinelles Lernen entschieden haben, tatsächlich über viele Karrieremöglichkeiten nachgedacht haben, die von Biologie über Luft- und Raumfahrt bis hin zu Informatik und Elektromotoren reichen. Warum haben Sie diese Wahl getroffen? Ja, ich habe mich schon immer zum Ingenieurwesen hingezogen gefühlt, weil ich es wirklich mag, Rätsel zu lösen und Probleme zu lösen. Und als ich für mein Grundstudium am MIT ankam, dachte ich über verschiedene Ingenieurstudiengänge nach. Und was mich wirklich an der Informatik interessiert hat, war, dass sie einem viel Flexibilität in Bezug auf die verschiedenen Wege gibt, die man einschlagen kann. Im Wesentlichen befähigt die Informatik einen, viele verschiedene Dinge mit Software und Code zu bauen, und damit könnte ich schließlich in die Biologie einsteigen. Wenn ich wollte, könnte ich in die Robotik einsteigen, da gibt es in der Informatik jede Menge wirklich spannende Dinge zu tun. So landete ich in der Informatik, und nachdem ich mich dann dafür entschieden hatte, gefiel mir die Mathematik, die hinter der KI steckt, wie Wahrscheinlichkeitsrechnung und Statistik, sehr gut. Und die Herausforderung, wie wir einen Computer bauen können, der so sehen kann wie Menschen, hat mich auch wirklich fasziniert. Das kann Dinge und Bilder wahrnehmen und in der Welt Maßnahmen ergreifen, ja. >> Es war also fast so, als hättest du die Wahl, wenn du Informatik studierst, könntest du alles machen. Und das macht es so, ich schätze, was Mason gesagt hat: Software frisst die Welt auf. Und ich glaube, Jen Huang hat gesagt, dass KI Software frisst, und es fühlt sich so an, als ob man, wenn man sich mit maschinellem Lernen beschäftigt, seine Nase in alle Ecken der Welt stecken und lustige Dinge tun kann. >> Absolut, ich denke, maschinelles Lernen ist ein wirklich spannender Bereich. Und ich habe erwähnt, dass die Informatik sehr flexibel ist und viele Türen öffnet, und ich denke, dass maschinelles Lernen dem sehr ähnlich ist. Es gibt so viele verschiedene Dinge, die Sie mit maschinellem Lernen machen können, und so viele verschiedene Anwendungen. Ich bin wirklich gespannt auf Anwendungen in der Robotik. Aber wir haben viele verschiedene Anwendungen der Sprachverarbeitung in der Biologie und in anderen Bereichen gesehen. >> Ihre Spezialisierung, von den vielen Dingen, die Sie tun, liegt also zu einem großen Teil in der Robotik. Und wenn sich die Leute die populären Medien oder sozialen Medien ansehen, sehen Sie zum Beispiel diese Videos von humanoiden Robotern, die auf und ab springen und unglaubliche Dinge tun. Wie soll man darüber nachdenken, was Roboter heute wirklich können und was nicht? Denn es gibt diese erstaunlichen Demovideos, die scheinbar fast alles können. Doch wie soll man entscheiden, was tatsächlich machbar ist? >> Ja, die Videos, die wir online von Boston Dynamics sehen, zum Beispiel von Robotern, die Rückwärtssaltos machen oder Parkour machen, und natürlich. Sind wirklich beeindruckend, weil sie wirklich kompliziert aussehen. In gewisser Weise ist es wirklich einfach, den Roboter zu vermenschlichen und zu denken, dass es sehr beeindruckend wäre, wenn man eine Person sehen würde, die das tut. Und sie würden wahrscheinlich noch viele andere wirklich beeindruckende Dinge tun können. Und der Haken ist, dass es in diesen Videos und vielen beeindruckenden Demonstrationen der Robotik um Robotik geht. Der Roboter wurde für diese spezielle Umgebung eingerichtet und so abgestimmt, dass er mit dieser Einrichtung gut funktioniert. Und wenn Sie etwas an der Umgebung ändern, am Setup oder an dem, was es tun soll, dann funktioniert die Demo nicht mehr. Dann fällt der Roboter flach. Die wirklich große Herausforderung bei der Robotik besteht also darin, Roboter dazu zu bringen, zu verallgemeinern und viele verschiedene Szenarien mit vielen unterschiedlichen Objekten und vielen unterschiedlichen Umgebungen bewältigen zu können. Und derzeit können Roboter Dinge in sehr kontrollierten Umgebungen wie in Fabriken erledigen, und sie sind in Fabriken sehr nützlich. Und was mich wirklich begeistert und was in der Robotik wirklich schwierig ist, ist, ihnen im Allgemeinen die Flexibilität zu geben, die die Fähigkeiten haben, die Menschen beherrschen. >> Nur um ein wenig mehr auf die konkrete Umgebungssache einzugehen, bevor wir uns auf das Allgemeine konzentrieren. Wenn jemand die nächste coole Robotik-Demo sieht, wie sollte er darüber denken oder welche Fragen

sollte er sich stellen? >> Ja, also denke ich, dass die erste Frage, die jemand stellen kann, ist: Was passiert, wenn sich etwas in der Umgebung ändert? Was passiert, wenn Sie einen Block verschieben oder den Roboter beim Start in eine etwas andere Position bringen ? Oft gibt es solche Tests der Robustheit oder Widerstandsfähigkeit des Systems gegenüber diesen Schwankungen nicht. Und wenn Sie anfangen, diese Demos zu sehen, in denen Sie tatsächlich den Roboter anstupsen oder die Umgebung ein wenig ändern, funktioniert es immer noch. Dann denke ich, dass wir wirklich beeindruckt sein können. >> Cool, ein Großteil Ihrer Arbeit bestand also speziell darin, Roboter robust zu machen, um sie zu verallgemeinern oder auf viele verschiedene Umgebungen anzuwenden. Erzählen Sie mir einfach mehr über diese Arbeit und wie Sie dieses Problem angehen. >> Auf jeden Fall, denn in anderen Bereichen des maschinellen Lernens haben wir große Erfolge erzielt, wenn wir einen sehr großen Datensatz verwenden. Nehmen Sie beispielsweise die gesamte Wikipedia und trainieren Sie ein neuronales Netzwerk, um die Daten in diesem sehr großen und breiten Datensatz modellieren zu können. Und was mich in der Robotik begeistert, ist, ob wir diesen Erfolg einigermaßen wiederholen können. Wenn wir einen ähnlich vielfältigen Datensatz sammeln und es Robotern ermöglichen können, aus diesem Datensatz zu lernen. Das ist aus mehreren Gründen eine Herausforderung. Zum einen verfügen wir noch nicht über einen vorhandenen Datensatz und über Wikipedia zur Robotermotorsteuerung. Im Internet gibt es keine Daten darüber, wie man beispielsweise die Finger eines Roboters genau bewegt, um Schuhe zuzubinden. Wir verfügen also noch nicht über Unmengen unterschiedlicher Daten, aber glücklicherweise sollten Roboter grundsätzlich in der Lage sein, selbst Daten zu sammeln. Und das ist eine Chance, denn es bedeutet, dass wir möglicherweise eine große Menge unterschiedlicher Daten sammeln können, indem wir den Robotern erlauben, ihre eigenen Daten zu sammeln. Aber es ist auch eine Herausforderung, denn es bedeutet, dass wir herausfinden müssen, wie wir es Robotern ermöglichen können, nützliche und vielfältige Daten zu sammeln. Daher gibt es viele verschiedene Dinge, die wir im Hinblick auf die Lösung dieses Problems erforschen. Einige davon ermöglichen es Robotern, ihre eigenen Daten in vielen verschiedenen Umgebungen zu sammeln. Einige demonstrieren einem Roboter, wie er eine Aufgabe ausführt. Wir haben auch ein wenig untersucht, wie man Daten aus dem Internet nutzen kann, beispielsweise Videos von Menschen, um einem Roboter zu zeigen, wie er eine Aufgabe erledigt. Und die Einbeziehung all dieser unterschiedlichen Datenquellen, um Robotern dies zu ermöglichen, sobald sie hoffentlich sehr unterschiedliche Datensätze wie diesen gesehen haben. Möglicherweise können sie nicht nur auf das eine Szenario verallgemeinern, das sie in einer Laborumgebung gesehen haben, sondern langfristig möglicherweise auch auf die reale Welt. >> Eines der Dinge, die wir gesehen haben, ist maschinelles Lernen und Verstärkungslernen. Unsere Räume eignen sich hervorragend zum Spielen von Videospielen. Weil man praktisch unendlich viele Daten sammeln kann, wenn man ein Videospiel mit sich selbst spielt. Was halten Sie von der Rolle der Simulation? Ja, wir haben definitiv viele Erfolge von KI-Systemen gesehen, die sich bei Videospielen sehr gut bewährt haben. Und die Simulation ist in gewisser Weise ähnlich, da wir auch in der Simulation viele Daten sammeln können. Eine Herausforderung bei der Simulation besteht darin, dass die Physik in der Simulations-Engine nicht genau mit der realen Welt übereinstimmt. Wir kennen im Allgemeinen die Gesetze der Physik, aber die Reibung auf einem Tisch ist beispielsweise möglicherweise nicht genau bekannt. Oder die Reibung zwischen den Rillen eines Verschlusses und einer Wasserflasche, wenn Sie versuchen, eine Wasserschale zu modellieren. Das Modellieren ist sehr schwierig und fehleranfällig. Und dann, wenn Sie versuchen, in diesem Simulator eine Richtlinie zu trainieren, um beispielsweise eine Wasserflasche zu verschließen, und sie dann in der realen Welt einzusetzen. Aufgrund dieser Fehler in der Simulation funktioniert es tatsächlich nicht. Daher halte ich es für eine vielversprechende Datenquelle. Aber wegen dieser Ungenauigkeiten und dieser Herausforderung, etwas, was in der Simulation gelernt wurde, auf die reale Welt zu übertragen. Ich denke nicht, dass dies der einzige Ansatz ist, auf den wir setzen sollten. Ich denke, dass wir versuchen sollten, viele reale Daten zu nutzen. >> Und ich möchte jemanden, der das sieht, überraschen:

Warum ist es so heiß, die reale Welt zu simulieren? Es sind nur viele Flaschen, zwei Plastikteile, und irgendwie können wir nicht herausfinden, was mit zwei Plastikteilen los ist. >> Ja. Ich denke also, dass es viele Herausforderungen gibt. Ich meine, zunächst geht es nur um die Modellierung der Low-Level-Physik. Viele Simulations-Engines da draußen modellieren die Zeit zu einem bestimmten Zeitpunkt, und in Wirklichkeit ist die Zeit viel flüssiger, als wenn sie mit etwa 100 Hertz oder so ähnlich arbeitet. Und um diese Dinge tatsächlich genau zu modellieren, müssen Sie möglicherweise eine sehr feine Zeitunterscheidung modellieren, und das wird sehr teuer. Und dann ist Ihr Simulator möglicherweise tatsächlich langsamer als die tatsächliche Ausführung von Dingen in der realen Welt. Und dann ist es im Vergleich zur Datenerhebung in der realen Welt nicht mehr sinnvoll. Und zweitens ist die reale Welt sehr vielfältig. Und so tatsächlich den Inhalt aller Arten von Objekten erstellen, denen man auch im Alltag begegnet. Wenn Sie zu Mittag essen oder einen Tisch aufräumen, ist die Erstellung all dieser Inhalte eigentlich eine wirklich manuelle Anstrengung. Und normalerweise möchte man in maschinellen Lernsystemen auf Dinge verzichten, die viel manuellen Aufwand erfordern. Und versuchen Sie, Dinge zu automatisieren und zu automatisieren, damit sie skaliert werden können. >> Ja. Und tatsächlich funktionieren Fabriken angesichts der Vielfalt der heutigen Umgebungen hervorragend mit Robotern, in denen die Umgebung relativ kontrolliert ist und im Durchschnitt relativ wenige unerwartete Dinge passieren. Ich weiß, dass es bei einigen Ihrer Arbeiten darum ging, Roboter in die Häuser der Menschen zu schicken und Dinge in deren Küchen zu erledigen. Das ist eine der unkontrolliertesten Umgebungen der Welt, sagen Sie etwas mehr dazu. >> Ja, absolut. Daher denke ich, dass wir in kontrollierten Umgebungen große Erfolge erzielt haben. Und auf lange Sicht würde ich gerne Roboter sehen, die in jede Umgebung wie ein Zuhause oder ein Büro eindringen und den Menschen nützlich sein könnten. Und um das zu erreichen, wollen wir tatsächlich damit beginnen, Daten in diesen Umgebungen zu sammeln. In der Robotikforschung sammeln wir die Daten oft einmal in einer Laborumgebung und lernen auf der Grundlage dieser Daten. Aber wenn wir nur reale Daten in einem Labor sammeln, werden wir die Roboter nie wirklich in die reale Welt bringen. Deshalb versuchen wir, mit der Datenerfassung zu beginnen und Roboter tatsächlich in die reale Welt einzusetzen, damit sie die Vielfalt all dieser verschiedenen Häuser sehen können. Anstelle der engen Daten, die Sie im Labor sehen. >> Ich möchte also Ihr Bauchgefühl über eine Sache erfahren, nämlich, dass Sie darüber gesprochen haben, dass Sie einen Roboter dazu bringen wollen, in ein brandneues Zuhause zu gehen und eine Schüssel Müsli zuzubereiten, oder? Anspruchsvoll gegenüber Flüssigkeiten, Müsli-Kühlschrank. Was sagen Sie dazu, wie viele verschiedene Häuser und Küchen ein Roboter sehen muss? Wie viele Trainingsbeispiele braucht ein Roboter also Ihrer Meinung nach, um gute Chancen zu haben, in der nächsten Küche auf dem N plus zuerst zu landen? >> Ja. Ich denke, es hängt wirklich davon ab, wie schwer die Aufgabe ist. Wenn die Aufgabe so einfach ist, wie ich es nicht kenne, vielleicht das Drücken eines Knopfes an der Mikrowelle, dann denke ich, dass Sie dafür möglicherweise nicht ganz so viele Daten benötigen. Aber ich denke, selbst für eine neue Küche braucht man wahrscheinlich immer noch eine ziemlich große Anzahl an Küchen, um sie auf eine neue zu verallgemeinern, vielleicht 20, vielleicht 100. Wenn es sich um eine komplexere Aufgabe handelt, wie zum Beispiel das Zubereiten einer Müslischale, ist das etwas Das ist für die Leute super einfach. Aber für Roboter ist es tatsächlich sehr schwierig, weil viele verschiedene Schritte erforderlich sind. Dafür braucht man meiner Meinung nach mehr Daten. Und wahrscheinlich auch noch mehr Küchen, vielleicht zu Tausenden. Und ich denke auch, dass selbst wenn wir Tausende oder Zehntausende von Datenküchen hätten, das Problem immer noch nicht vollständig gelöst wäre. Wir haben Anwendungen wie das autonome Fahren gesehen, bei denen Unmengen an Daten vorliegen und das Problem noch nicht ganz gelöst ist. Daher denke ich, dass wir auch versuchen müssen, über die Datenverteilung hinauszugehen und es den Robotern zu ermöglichen, sich spontan an eine neue Umgebung anzupassen. >> Nun, Tausende, das ist mehr als mein Bauchgefühl, aber es ist interessant zu sehen, wie es sich entwickelt und so weiter zum Thema, Roboter robuster zu machen. Ich weiß, dass sich ein Großteil Ihrer Forschungsarbeit auch mit Meta-

Lernen oder dem Lernen, wie man lernt, befasste, anstatt dass wir dem Computer sagen, dass dies ein Lernalgorithmus ist, um einen Computer dazu zu bringen, noch besser zu lernen. Können Sie etwas dazu sagen ? >> Ja, absolut. Ein Teil der Motivation besteht also darin, dass wir nicht wollen, dass ein Roboter, wenn er auf eine neue Küche trifft, einfach das ausführt, was er in dieser neuen Küche gelernt hat. Möglicherweise muss es sich tatsächlich spontan anpassen und lernen. Dies kann beispielsweise erforderlich sein, wenn die Tür zu einer Speisekammer geöffnet werden soll und die Tür etwas anders ist. Möglicherweise sind ein wenig Erfahrung und Versuch und Irrtum an dieser Tür erforderlich, um herauszufinden, wie man sie öffnet. Und wir möchten es Systemen ermöglichen, diese Art von Lernprozess während der Testzeit in neuen Szenarien spontan durchzuführen. Und wenn Sie maschinelles Lernen zum Testzeitpunkt einfach von Grund auf mit einer sehr kleinen Datenmenge ausführen, funktioniert das leider nicht besonders gut. Um etwas zu lernen, benötigt man große Datenmengen. Beim Metalllernen geht es also darum, Erfahrungen aus früheren Küchen oder früheren Aufgaben zu sammeln. Und optimieren Sie die Fähigkeit, neue Dinge zu lernen, die Fähigkeit, neue Dinge zu lernen, basierend darauf, dass Sie viele Dinge zuvor gelernt haben. >> So könnte man zum Beispiel die beste Lerngeschwindigkeit einstellen, die sich an eine neue Küchenumgebung anpassen möchte, und viele andere Dinge. >> Ja, absolut. Die einfachste Einstellung besteht also darin, dass Sie nur für einen Parameter optimieren, beispielsweise die Lernrate in einer neuen Situation. Und bei komplexeren Metall-Lernalgorithmen können Sie den gesamten Lernalgorithmus optimieren. >> Wenn Sie also einer der Weltexperten für Meta- Lernen sind, denke ich, dass dies tatsächlich ein großer Teil Ihrer Doktorarbeit ist, wenn ich mich daran erinnere. Was Sie vor langer Zeit mit Peter gemacht haben, der mein Doktorand bei Standard war. Und sagen Sie einfach mehr darüber, wie Meta- Lernen funktioniert. >> Ja. Es gibt also verschiedene Ansätze zum Metalllernen. Aber der Ansatz meiner Doktorarbeit bestand im Wesentlichen darin, ein Lernproblem in ein anderes Lernproblem zu integrieren. Es handelt sich also tatsächlich um diese zweistufige Optimierung, dieses zweistufige Lernproblem, bei dem Sie tatsächlich versuchen, alle Parameter dieses inneren Lernproblems zu optimieren. So können Sie schneller oder effizienter lernen. >> Ja, ich erinnere mich, dass ich einige frühe Artikel gelesen habe, die sehr beeindruckend waren. Also viele Softwarealgorithmen, Algorithmen für maschinelles Lernen, Algorithmen für verstärktes Lernen. Ein Teil Ihrer Arbeit besteht darin, dass Sie auch ein Robotiklabor betreiben, in dem Roboter um Roboterarme herumlaufen. Ich bin neugierig auf jemanden, der noch nicht viel Zeit in einem Robotiklabor verbracht hat. Können Sie uns sagen, wie es ist, wenn ein Student in Ihr Labor kommt, wie es ist, morgens zu erscheinen und zu arbeiten und in einem Robotiklabor zu arbeiten? >> Ja. Im Labor unterscheidet es sich also nicht wesentlich von der Durchführung von Experimenten zum maschinellen Lernen, bei denen Sie Code ausführen, versuchen, ein Modell zu trainieren und dieses Modell dann zu bewerten. Aber zusätzlich zu diesen alltäglichen Dingen gibt es auch eine Schnittstelle zum echten Roboter. Daher beginnen viele Projekte oft mit der Simulation und beginnen dann mit der Iteration von Algorithmen. Verwenden einer Physik-Engine, bei der möglicherweise bestimmte Aspekte der Simulation entworfen werden, um die Hypothesen zu testen, die Sie testen möchten. Und manchmal erfordern Produkte diesen Simulationsschritt auch nicht oder Sie haben die Simulation bereits durchlaufen. Und beim echten Roboter gibt es alles vom Einrichten des Kontrollstapels. So können Sie dem Roboter tatsächlich befehlen, sich an eine bestimmte Position zu bewegen, die Kameras einzurichten und Daten mit dem Roboter zu sammeln. Möglicherweise sammeln Sie Daten, indem Sie beispielsweise einen VR-Controller verwenden, um dem Roboter mitzuteilen, dass er sich zu bestimmten Positionen bewegen soll. Und wenn es dann tatsächlich darum geht, ein Modell zu evaluieren, geht es darum, es tatsächlich auf dem Roboter laufen zu lassen und zu sehen, was der Roboter im wirklichen Leben macht. Zumindest für mich ist es ein ziemlich unterhaltsamer Prozess. Ich finde es sehr lohnend, den Roboter tatsächlich etwas tun zu sehen, anstatt nur ein paar Zahlen oder eine gewisse Genauigkeit Ihres Modells zu sehen. Es erfordert aber auch ein wenig Ausdauer, denn Roboter können kaputt gehen. Möglicherweise ist der

Motor kaputt oder Sie haben die Kamera nicht richtig eingestellt. Es könnte zum Beispiel sein, dass Sie nur Nullen erhalten. Es können viel mehr Fehler auftreten als in einem System ohne Roboter. >> Es gibt also viele Leute, die Erfahrung mit dem Debuggen von Software haben, und dann gibt es noch das Debuggen von Algorithmen für maschinelles Lernen, das unterschiedliche Vorgehensweisen und Ideen hat. Und können Sie dann etwas darüber sagen, wie das Debuggen von Hardware in einem Robotiklabor aussieht? >> Auf jeden Fall. Ich meine, im Allgemeinen möchte man beim Debuggen mögliche Fehlerquellen und mögliche Fehlerquellen ausschließen, und oft vermutet man, dass es an der Software oder der Hardware liegt. Und von da an denke ich, dass viele der Dinge des maschinellen Lernens anwendbar sein werden. Sie möchten Ihre Daten einsehen. Und da viele der Daten tatsächlich im Labor gesammelt werden, stammen sie möglicherweise nicht aus einem Datensatz, dann stimmt möglicherweise etwas nicht. Und vielleicht möchten Sie sich die Bilder, die der Roboter sieht, tatsächlich ansehen und vielleicht einige der Funktionen visualisieren, die Ihr Modell gelernt hat. Ich denke, eine Sache, die wir kürzlich gemacht haben, war, dass wir tatsächlich eine Richtlinie auf mehreren verschiedenen Robotern bereitstellen wollten. Und so haben wir tatsächlich Aktionen an einem Roboter wiederholt und sichergestellt, dass der Roboter eine bestimmte Bewegung reproduziert, die wir reproduzieren wollten. Wie Sie debuggen, hängt stark vom Problem ab. Ich weiß nicht, ob es einen einzelnen Fehler gibt, der am häufigsten auftritt, denn es gibt alle möglichen Fehlerquellen. Aber ja, das gibt uns einen kleinen Eindruck davon, wie es aussieht. >> Und ich dachte, eines der schönen und frustrierenden Dinge an der Robotik im Gegensatz zur Software liegt in der Software. Wenn Ihnen ein Fehler unterläuft, können Sie jederzeit zur früheren Version der Software zurückkehren. Aber dieses Mal, wenn du etwas machst und dein Roboter kaputt geht und du dann gehst, Junge, das sind noch zwei Monate Arbeit oder so, um das wieder aufzubauen. Das ist ein einzigartiges Hardware-Erlebnis. >> Auf jeden Fall. Und weil wir viel an der Software arbeiten, versuchen wir normalerweise, Roboter zu bekommen, die nicht kaputt gehen, wie schöne High-End-Roboter, damit wir uns mehr um die Software und weniger um die Hardware kümmern können. Aber wenn es Fälle gibt, in denen ein Roboter kaputt gehen könnte, versuchen wir, Roboter zu schaffen, bei denen man Teile ganz einfach austauschen kann. Wenn ein Motor kaputt geht, kann man einfach einen neuen Motor einbauen, anstatt das Ganze umbauen zu müssen. Aber es gibt auf jeden Fall viele Herausforderungen, die auf uns zukommen. >> Ja. Und wissen Sie, ich bin früher autonome Hubschrauber geflogen. Glücklicherweise gelingt es uns jedes Mal, wenn ein Helikopter abstürzt, ihn sicher und von Menschen fernzuhalten, dann ist das ein ehrlich gesagt zerstörter Helikopter und dann müssen wir einfach einen neuen bauen, aber es ist schön, wenn man das zu oft macht. >> Ja. Glücklicherweise stürzen die Arme nicht allzu stark ab. >> Ich verstehe. Ja. Ja, vielleicht hätte ich mich stattdessen dafür entscheiden sollen, daran zu arbeiten, nicht wahr? Ein Werkzeug Ihrer Wahl, das Sie in Ihrem Robotiklabor häufig verwendet haben, ist das Reinforcement Learning. Können Sie uns etwas zu den praktischen Lektionen, Tipps und Tricks sagen, wie Sie die erstaunlichen Algorithmen des Verstärkungslernens durch Roboter auf Ihr Herz anwenden können? >> Ja. Reinforcement Learning ist also ein wirklich attraktives Framework für die Robotik, da es einem Roboter im Prinzip ermöglichen würde, durch Ausprobieren etwas völlig selbstständig aus Daten zu lernen, die er selbst gesammelt hat. Aus diesem Grund sind wir sehr gespannt darauf, aber es ist aus mehreren Gründen auch eine große Herausforderung. Beim ersten Verstärkungslernen wird also normalerweise davon ausgegangen, dass Sie ein Belohnungssignal, ein Feedback, erhalten. Und wenn Sie lernen möchten, wie man ein Spiel spielt, beispielsweise Pong oder ein anderes Atari-Spiel, können Sie die Punktzahl einfach als Belohnungsfunktion verwenden und einfach versuchen, Ihre Punktzahl zu maximieren. Aber die reale Welt gibt Ihnen keine Punktzahl. Es gibt dem Roboter keine Punktzahl, wenn er versucht, etwas zu lernen. Stattdessen muss der Roboter möglicherweise seine eigene Belohnungsfunktion erlernen. Möglicherweise muss es lernen, was es bedeutet, eine Aufgabe zu erfüllen und gleichzeitig zu lernen, wie man diese Aufgabe in der physischen Welt erledigt. Das ist also eine große Herausforderung und eine Art und Weise, wie wir sie angehen. Eine

weitere Herausforderung liegt typischerweise in diesem Versuch und Irrtum. >> Und wie lernt man die Belohnung? Wie von einer Demonstration? Wie, wie lernt man die Belohnung? Ja. >> Also haben wir es auf verschiedene Arten gelernt. Wir geben ihm oft zumindest einige Beispiele dafür, wie Erfolg aussieht. Und manchmal führen wir Ihnen auch eine vollständige Demonstration vor, die ein Beispiel dafür ist, was wir von Ihnen erwarten. Und dann können Sie sich mit Techniken wie dem inversen Verstärkungslernen befassen, an dem Sie vor langer Zeit ebenfalls gearbeitet haben, und tatsächlich versuchen, den Verstärkungslernprozess umzukehren und herauszufinden, welche Belohnungsfunktion diesem Verhalten zugrunde liegt. Über das Lernen und die Belohnungsfunktionen hinaus besteht eine weitere Herausforderung in der Autonomie im Prozess des verstärkenden Lernens. Normalerweise stellen Sie es sich als einen Versuch-und-Irrtum-Prozess vor, bei dem Sie versuchen, die Aufgabe zu lösen, und es dann noch einmal versuchen möchten. Und zwischen dem Ausprobieren und erneuten Ausprobieren der Aufgabe müssen Sie irgendwie zu einem Ausgangspunkt zurückkehren. Und wenn Sie zum Beispiel lernen möchten, wie man einen Rückwärtssalto macht, müssen Sie bei einem erneuten Versuch wieder aufstehen, nachdem Sie hingefallen sind. Aber tatsächlich ist es manchmal auch eine ziemliche Herausforderung, zu lernen, dass das Zurücksetzen des Verhaltens eine große Herausforderung darstellt. Und es gibt dieses herausfordernde Zusammenspiel zwischen dem Erlernen, wie man die Aufgabe erledigt, und dem Erlernen, wie man sich von all den Fehlern erholt, die man während des gesamten Prozesses bei der Aufgabe versagt hat. >> Es automatisiert den Forschungsprozess wirklich, sodass Sie wiederholte Experimente durchführen können, ohne jedes Mal einen Roboter herumschleppen zu müssen, um es zurückzustellen. >> Genau. Ja. >> Cool, wirklich cool. Und tatsächlich glaube ich, dass Sie mit Ihrer Anwendung des verstärkenden Lernens zu den ersten Forschern gehören, die mehr oder weniger oder vollständig End-to-End-Robotik betreiben. End-to-End-Lernen zur Verstärkung. Können Sie etwas darüber sagen, wie das in den Anfangstagen war und warum Sie diesen Weg eingeschlagen haben? >> Ja. Zu Beginn meiner Doktorarbeit begann Deep Learning gerade erst populär zu werden. Nachdem Alex Net gezeigt hatte, dass man mit tiefen neuronalen Netzen, die durchgängig für Computer Vision trainiert wurden, sehr gut zurechtkommen kann. Und ich war gerade für meine Doktorarbeit in Berkeley angekommen. Es gab einige Leute, die am verstärkten Lernen mit neuronalen Netzen arbeiteten. Und ich war ziemlich begeistert von dieser Arbeit und wollte mir die früheren Arbeiten ansehen, es war alles blind. Der Roboter benutzte in keiner Weise Kameras. Und ich wollte sehen, ob wir ein neuronales Netzwerk erlernen könnten, das die von der Kamera des Roboters aufgenommenen Bilder direkt auf die auf die Gelenke des Roboters ausgeübten Drehmomente abbildet. Und bei diesem ersten Projekt war es eine Menge Arbeit, dieses erste Ergebnis tatsächlich zu erzielen. Aber wir waren gespannt, als wir sahen, dass wir tatsächlich ein einzelnes neuronales Netzwerk bekommen konnten, das so etwas wie das Aufschrauben eines Verschlusses auf eine Flasche oder das Anbringen eines Kleiderbügels an einer Garderobe lernen konnte, wobei ein einzelnes neuronales Netzwerk Pixel direkt in Drehmomente umwandeln konnte. Und es war ein ziemlich aufregendes Ergebnis. Es war definitiv ganz anders als das, woran der Rest der Robotik-Community arbeitete. Und ich denke, der Artikel, an dem wir gearbeitet haben, wurde tatsächlich zweimal abgelehnt, weil die Leute Deep Learning nicht mochten, weil es den Robotikern nicht gefiel, dass diese neuronalen Netze sehr Blackbox-ähnliche Netze waren und man sie nicht verstehen konnte. Aber schließlich wurde es veröffentlicht und akzeptiert. Und jetzt ist es tatsächlich zu einem viel wichtigeren Paradigma in der Robotik geworden, wo viel mehr Menschen neuronale Netze einsetzen, um die Politik darzustellen, um darzustellen, wie der Roboter Aktionen basierend auf den Kamerabildern auswählt. >> Ich denke, dass Ihre frühen Arbeiten ziemlich visionär waren und eine wichtige Hoffnung gesetzt haben, eine wichtige neue Richtung für das Fachgebiet vorgegeben haben. Was ist Ihre Prognose, wohin das führen wird? Glauben Sie, wie viel Prozent der praktischen Robotikarbeit dieser Zug von Ende zu Ende ausmacht? Wie stark wird dieser Wert Ihrer Meinung nach weiter steigen oder nicht? >> Ja, ich glaube auf jeden Fall, dass es noch deutlich wachsen wird.

Noch immer beschäftigt sich nicht die gesamte Robotik-Forschungsgemeinschaft mit solchen Ansätzen, auch weil es in der Kontrolltheorie nützliche Techniken gibt, die besonders in kontrollierten Szenarien, wie wir sie besprochen haben, nützlich sind. Aber ich denke, dass es sicherlich wachsen wird. Ich denke, dass ich insbesondere angesichts der weiterhin beeindruckenden Fortschritte im Bereich des maschinellen Lernens hoffe, dass dies mehr Menschen davon überzeugen wird, dass diese Art von Paradigma auch für die Robotik wirklich vielversprechend ist, insbesondere im Hinblick auf den Umgang mit der enormen Vielfalt an Objekten und Objekten Umgebungen auf der Welt. Im Hinblick auf die Zukunft, im Hinblick darauf, zu verstehen, was in den nächsten 10 oder 20 Jahren in der Robotik passieren könnte, freue ich mich wirklich sehr über das, was wir ein wenig angesprochen haben: den Versuch, Roboter anhand umfassenderer Datensätze zu trainieren. Ein Großteil des Roboterlernens besteht immer noch darin, einen Datensatz für ein Projekt in einem Labor zu sammeln und dann anhand dieses kleinen Datensatzes zu trainieren. Es ist klein, weil es sein muss, weil Sie es für dieses Projekt gesammelt haben. Und ich denke, wenn wir uns vorstellen, dass Computer-Vision-Forscher wahrscheinlich keine großen Fortschritte machen würden, wenn sie für jedes einzelne Projekt, das sie durchführen, Bildnetze sammeln müssten. Daher prognostiziere ich zumindest für die kommenden Jahre, dass wir uns einem Paradigma zuwenden, bei dem wir Datensätze häufig wiederverwenden, viel größere Datensätze verwenden und im Idealfall Datensätze zwischen Institutionen und Roboterplattformen teilen. Und die Daten, auf denen diese Systeme trainiert werden, so zu vergrößern, dass sie breiter verallgemeinert werden können. >> Das ist eine spannende Vision. Und wie kommen wir dorthin? Für Bilder verwendet jeder PNG-Dateien, GIF-Dateien und JPEG-Dateien, sodass die Daten hochgradig portierbar und in hohem Maße gemeinsam nutzbar sind. Wenn verschiedene Forschungslabore nur über unterschiedliche Hardware verfügen, wie lässt sich das dann für die Robotik umsetzen? >> Ja, es ist definitiv eine Herausforderung. Der Punkt, an dem wir anfangen, und auf den ich mich sehr freue, besteht darin, zunächst einmal zu versuchen, tatsächlich einen kleinen Konsens zu erzielen und die Community zusammenzubringen, um zu sagen: Okay, fangen wir mit dieser Roboterplattform mit ungefähr diesem Setup an, damit wir kann zumindest einen Teil davon standardisieren und dann versuchen, viele, viele verschiedene Daten auf dieser ziemlich standardisierten Plattform zu sammeln. Beginnen Sie dort, und wenn wir dann in der Lage sind, durch die Sammlung eines breiten Datensatzes mit diesen standardisierten Entscheidungen ein paar Lebenszeichen und positive Erfolge zu verzeichnen, dann beginnen wir, immer weniger zu standardisieren und machen weiter. Vielleicht derselbe Roboter, aber mit einem anderen Steuerungsstapel, und dann fängt man vielleicht an, sich anderen Versionen dieses Roboters zuzuwenden, anderen Greifern und letztendlich vielen verschiedenen Robotern und vielen Umgebungen. >> Cool, ja, großartig, das ist eine aufregende Vision. Das ist also eines der Dinge, die Sie in der Robotik tun. Wenn Sie es einmal allgemeiner betrachten: Bei all Ihrer Arbeit im Bereich Reinforcement Learning, von der einige nicht einmal auf die Robotik anwendbar sind, andere Anwendungen jedoch, wohin hoffen Sie, dass sich das Enforcement Learning und die Robotik in den nächsten Jahren entwickeln werden? >> Ja, ich denke also, ich hoffe, dass es zu wiederverwendbaren Datensätzen und sehr breiten Datensätzen kommt. Ich glaube auch, dass es über die Robotik hinaus noch viele andere Anwendungen des Reinforcement Learning gibt. So haben wir beispielsweise in einigen unserer jüngsten Arbeiten Reinforcement Learning und eigentlich Meta Reinforcement Learning, eine Kombination aus Meta Learning und Reinforcement Learning, auf die Bildung angewendet, um Feedback zu Schülerarbeiten zu geben, wobei die Schüler ein Spiel wie Breakout und programmierten Dann würde das System das Spiel tatsächlich spielen und die von den Schülern verursachten Fehler finden und ihnen Feedback geben, wie sie ihr Programm verbessern und weiterhin lernen können, wie man programmiert. >> Das ist ja sehr cool. Tatsächlich sind Roboter so visuell, dass man manchmal leicht vergisst, dass Reinforcement Learning eine allgemeine Technik zur Entscheidungsfindung mit verzögerten Belohnungen ist. Das ist für viele andere Dinge als Roboter nützlich. Ehrlich gesagt, wenn ich mehr Roboter sehe, habe ich immer versucht, alle meine Schüler

dazu zu bringen, das Löten zu lernen. Ich weiß nicht, ob das ein- >> Ich habe das Löten im Grundstudium gelernt, obwohl ich nicht glaube, dass ich das jemals in meiner Robotikarbeit verwendet habe. Wow, ich verstehe. Vielleicht haben sich die Dinge weiterentwickelt, aber ich musste, ja. >> Ich denke, wir würden einfach Roboter kaufen, die nicht gelötet werden müssen. >> Ihre Reise hat Sie also dazu geführt, Spitzenarbeit in den Bereichen Reinforcement Learning und Robotik zu leisten. Ich bin gespannt, welchen Rat Sie jemandem geben würden, der in das Reinforcement Learning oder in die Robotik einsteigen möchte. Nicht jeder besitzt einen coolen Roboter, das ist ein einfaches Programm, aber wie soll jemand anfangen? >> Ja, mein wichtigster Rat wäre einfach, zu versuchen, die Füße nass zu machen und einfach zu versuchen, etwas aufzubauen, zu versuchen, etwas zu programmieren und zu programmieren. Es gibt viele verschiedene Dinge, die Sie tun können. Und ich denke, dass Lernen, indem man versucht, etwas aufzubauen, indem man versucht, etwas zu tun, eine der coolsten Arten zu lernen ist, weil man ein bestimmtes Ziel vor Augen hat. >> Aber würden Sie etwas mit einem simulierten Roboter bauen oder denken Sie, Sie sollten einfach einen kleinen Roboter bauen? >> Ich denke, dass es wahrscheinlich am einfachsten ist, mit einem simulierten Roboter zu beginnen, und es gibt weniger Lernaufwand und weniger Dinge, die man tun muss, man muss nicht rausgehen und Hardware kaufen. Trotzdem denke ich, dass es großartig ist, gleich zu Beginn ein wenig mit Hardware zu arbeiten, und man lernt dadurch auf jeden Fall, wie schwierig Robotik ist. Robotik ist wirklich schwierig. Aber auch dadurch, dass man darum kämpft, den Roboter dazu zu bringen, etwas zu tun, ist es umso lohnender, wenn er tatsächlich etwas Cooles tut, wenn er etwas tut, was man von ihm wollte. Und ich weiß nicht, einige der Roboter, die wir verwenden, sind ziemlich teuer. Aber heutzutage gibt es tatsächlich viele Roboter, die man kaufen kann, die relativ günstig sind, oder man kann sozusagen seinen eigenen Roboter bauen, indem man Räder und eine Kamera oder vielleicht sogar ein Smartphone miteinander verbindet, um etwas zu bauen. Ich habe tatsächlich in der Mittelschule mit der LEGO-Robotik angefangen, und man konnte auch Roboter aus LEGO bauen und so anfangen. >> Ja, tatsächlich, ich erinnere mich, dass ich an LEGO-Robotikwettbewerben teilgenommen habe, als ich in der Mittelschule war oder so. Das waren interessante Erfahrungen. Was würden Sie also jemandem empfehlen, der sich auf maschinelles Lernen spezialisiert und in den Bereichen Durchsetzungslernen und Robotik arbeiten möchte? Was würden Sie ihm sonst noch empfehlen? Welche anderen Ressourcen wären Ihrer Meinung nach hilfreich? >> Auf jeden Fall, es gibt also Ressourcen, die sich nur auf das Lernen über Reinforcement Learning beziehen, und es gibt eine Art Online-Kurse, es gibt auch Kursvideos, es gibt Lehrbücher wie Sutton Barto. Das sind also viele Ressourcen. Ich denke auch, dass es online, ich weiß nicht, viele Videos, Tutorials und Codebasen gibt. Man kann sich tatsächlich eine Physik-Engine wie die Majko Physics Engine ansehen, die frei verfügbar ist, und versuchen, damit etwas zu erkunden. Also, ja, das sind ein paar Ressourcen. Es gibt auch eine Sache, die wir häufig zur Kommunikation mit Robotern nutzen: das sogenannte Robot Operating System (ROS). Daher kann es auch nützlich sein, sich darüber zu informieren. >> Das kam aus meinem Forschungslabor vor langer Zeit, in dem ich mit einer Garage zusammengearbeitet habe. Ja, gute Zeiten. Entschuldigung, ich wollte nicht unterbrechen. >> Ja, ich denke, das sind einige der Ressourcen, und ja, nicht viel mehr. >> Cool, ja, sehr cool. Da die Welt immer weiter voranschreitet und sich weiterentwickelt, hat man das Gefühl, dass die Zahl der Wege für jemanden, der sich in diesem Bereich engagieren möchte, sogar noch zunimmt und es für jemanden wahrscheinlich von Jahr zu Jahr ein bisschen einfacher wird, in dieses Feld einzusteigen. Es ist also eine aufregende Zeit. >> Ja, auf jeden Fall. >> Vielen Dank, dass Sie hier an der Stanford University dabei sind. Ich gehöre zu Professor Chelsea Finn, die sowohl in der Informatik- als auch in der Abteilung für Elektromotoren tätig ist und eine Forschungsgruppe leitet, die innovative Arbeiten zur Anwendung von Reinforcement Learning und maschinellem Lernen in der Robotik und anderen Anwendungen durchführt.