Machine Learning Foundations

Evolution of Machine Learning and Artificial Intelligence

February 2019

Notices

Contents

Abstract

Artificial Intelligence (AI) and Machine Learning (ML) are terms of interest to business people, technicians, and researchers around the world. Most descriptions of the terms oversimplify their true relationship. This paper provides a foundation for understanding artificial intelligence, describes how AI is now based on a foundation of machine learning, and provides an overview of AWS machine learning services.

Introduction

Most articles that discuss the relationship between artificial intelligence (AI) and machine learning (ML) focus on the fact that ML is a domain or area of study within AI. Although that is true historically, an even stronger relationship exists—that successful artificial intelligence applications are almost all implemented using a foundation of ML techniques. Instead of a component, machine learning has become the basis of modern AI.

To support this theory, we review how AI systems and applications worked in the first three decades versus how they work today. We begin with an overview of AI's original structure and approach, describe the rise of machine learning as its own discipline, show how ML provides the foundation for modern AI, review how AWS supports customers using machine learning. We conclude with observations about why AI and ML are not as easily distinguished as they might first appear.

Evolution of Artificial Intelligence

Symbolic Artificial Intelligence

Artificial Intelligence as a branch of computer science began in the 1950s. Its two main goals were to 1) study human intelligence by modeling and simulating it on a computer, and 2) make computers more useful by solving complex problems like humans do.

From its inception through the 1980s, most AI systems were programmed by hand, usually in functional, declarative, or other high-level languages such as LISP or Prolog. Several custom languages were created for specific areas (e.g., STRIPS for planning). Symbols within the languages represented concepts in the real world or abstract ideas, and formed the basis of most knowledge representations.

Although AI practitioners used standard computer science techniques, such as search algorithms, graph data structures and grammars, a significant amount of AI programming was heuristic—using rules of thumb—rather than algorithmic due to the complexity of the problems. Part of the difficulty of producing AI solutions then was that to make a system successful, all of the conditionals, rules, scenarios and exceptions needed to be added programmatically to the code.

Artificial Intelligence Domains

Researchers were interested in general AI, or creating machines that could function as a system in a way indistinguishable from humans, but due to the complexity of it, most focused on solving problems in one specific domain, such as perception, reasoning, memory, speech, motion, and so on. Major AI domains at this time are listed in the following table.

Table 1: Domains in Symbolic AI (1950s to 1980s)

Domain	Description
Problem Solving	Broad, general domain for solving problems, making decisions, satisfying constraints and other types of reasoning. Subdomains included expert or knowledge-based systems, planning, automatic programming, game playing and automated deduction. Problem solving was arguably the most successful domain of symbolic AI.
Machine Learning	Automatically generating new facts, concepts or truths by rote, from experience, or by taking advice.
Natural Language	Understanding and generating written human languages (e.g., English or Japanese) by parsing sentences and converting them into a knowledge representation such as a semantic network, and then returning results as properly constructed sentences easily understood by people.
Speech Recognition	Converting sound waves into phonemes, words and ultimately sentences to pass off to Natural Language Understanding systems, and also speech synthesis to convert text responses into natural sounding speech for the user.
Vision	Converting pixels in an image into edges, regions, textures and geometrical objects in order to make sense of a scene, and ultimately recognize what exists in the field of vision.
Robotics	Planning and controlling actuators to move or manipulate objects in the physical world.

Artificial Intelligence Illustrated

In the following diagram, lower levels depict layers that provide the tools and foundation used to build solutions in each domain. For example, below the *Primary Domains* are a sampling of the many *Inferencing Mechanisms* and *Knowledge Representations* that were commonly used at the time.

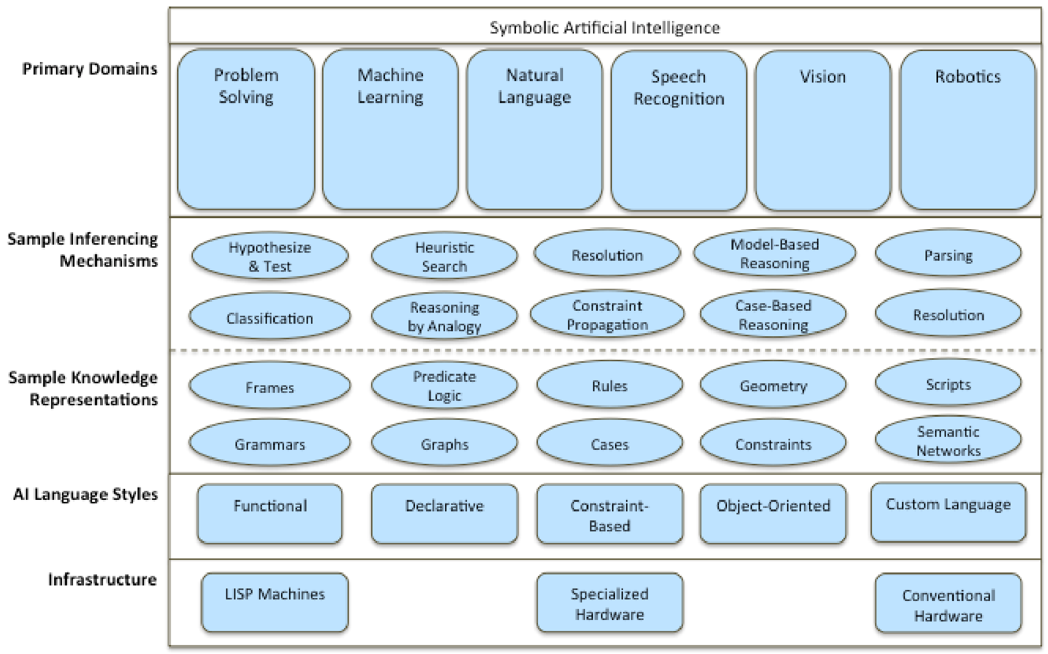

Figure 1: Overview of Symbolic Artificial Intelligence

The *Sample Knowledge Representations* stored knowledge and information to be reasoned on by the system. Common categories of knowledge representations included structured (e.g., frames which can be compared to objects and semantic networks which are like knowledge graphs) and logic-based (e.g., propositional and predicate logic, modal logic, and grammars). The advantage of these symbolic knowledge representations over other types of models is that they are transparent, explainable, composable, and modifiable. They support many types of inferencing or reasoning mechanisms, which manipulate the knowledge representations to solve problems, understand sentences and provide solutions in each domain.

The *AI Language Styles* and *Infrastructure* layers show some types of languages and infrastructure used to develop AI systems at this time. Both tended to be specialized, and not easily integrated with external data or enterprise systems.

A Question of Chess and Telephones

A question asked at the time was “which is a harder problem to solve: answering the telephone or playing chess at a master level?” The answer is counter-intuitive to most people. Although even children can answer a telephone properly, very few people play chess at a master level. However, for traditional AI, chess is the perfect problem. It is bounded, has limited well-understood moves and can be solved using heuristic search of the game’s state space. Answering a telephone on the other hand is quite difficult. Doing it properly requires multiple complex skills that are difficult for symbolic AI, including speech recognition and synthesis, natural language processing, problem solving, intelligent information retrieval, planning and potentially taking complex actions.

Successes of Symbolic AI

Generally considered to have disappointing results, at least in light of the high expectations that were set, symbolic AI did have several successes as well. Most of the software deemed useful was turned into algorithms and data structures used in software development today. Business rule engines that are in common use were derived from AI’s expert system inference engines and shells. Other common computing concepts credited to or developed in AI labs include timesharing, rapid iterative development, the mouse and Graphical User Interfaces (GUIs). The list below describes some of the strengths and limitations of this approach to artificial intelligence.

Table 2: Strengths and Limitations of Symbolic AI

Strength	Limitation
Simulates high-level human reasoning for many problems	Systems tended not to learn or acquire new knowledge or capabilities autonomously, depending instead on regular developer maintenance
Problem Solving domain had several successes in areas such as expert systems, planning and constrain propagation.	Most domains, including machine learning, natural language, speech and vision, did not produce significant, general results.
Can capture and work from heuristic knowledge rather than step-by-step instructions	Problem Solving domain, specifically expert or knowledge-based systems, require articulated human expertise extracted and refined using knowledge engineering techniques
Encodes specific, known logic easily, e.g., enforces compliance rules	Systems tended to be brittle and unpredictable at the boundaries of their scope - they didn’t know what they didn’t know
Straightforward to review internal data structures, heuristics and algorithms	Built on isolated infrastructure with little integration to external data or systems
Provides explanations for answers when requested	Requires more context and common-sense information to resolve many real-world situations

Strength	Limitation
Does not require significant amounts of data to create	Many approaches were not distributed or easily scalable, though there were hardware, networking and software constraints to distribution as well
Requires less compute resources to develop	Difficult to create and maintain systems
Many tools and algorithms were incorporated into mainstream system development	

As research money associated with symbolic AI disappeared, many researchers and practitioners turned their attention to different and pragmatic forms of information search and retrieval, data mining, and diverse forms of machine learning.

Rise of Machine Learning

From the late 1980s to the 2000s, several diverse approaches to machine learning were studied, including neural networks, biological and evolutionary techniques, and mathematical modeling. The most successful results early in that period were achieved by the statistical approach to machine learning. Algorithms, such as linear and logistic regression, classification, decision trees, and kernel-based methods (i.e., Support Vector Machines) gained popularity.

Later, deep learning proved to be a powerful way to structure and train neural networks to solve complex problems. The basic approach to training them remained similar, but there were several improvements driving deep learning's success, including:

- Much larger networks with many more layers
- Huge data sets with thousands to millions of training examples
- Algorithmic improvements to neural network performance, generalization capability, and ability to distribute training across servers
- Faster hardware (such as GPUs and Tensor Cores) to handle orders of magnitude more computations, which are required to train the complex network structures using large data sets.

Deep learning is key to solving the complex problems that symbolic AI could not. One factor in the success of deep learning is its ability to formulate, identify and use features discovered on its own. Instead of people trying to determine what it should look for, the deep learning algorithms identified the most salient features automatically.

Problems that were intractable for symbolic AI—such as vision, natural language understanding, speech recognition and complex motion and manipulation—are now being solved, often with accuracy rates nearing or surpassing human capability. Today, the answer to the question of which is harder for machines—answering the telephone or playing chess at a master level—is becoming harder to answer. Although there is important work yet to be done, machine learning has made significant progress in enabling machines to function more like people in many areas, including directed conversations with humans.

Machine learning has become a branch of computer science in its own right. It is key to solving specific, practical artificial intelligence problems.

AI has a New Foundation

Artificial intelligence today no longer relies primarily on symbolic knowledge representations and programmed inferencing mechanisms. Instead, modern AI is built on a new foundation, machine learning. Whether it is the models or decision trees of conventional mathematics-based machine learning, or the neural network architectures of deep learning, most artificial intelligence applications today across the AI domains are based on machine learning technology.

This new structure for artificial intelligence is depicted in the following diagram. The structure of this diagram parallels the diagram of symbolic AI in order to show how the foundation and the nature of artificial intelligence systems have changed. Although some of the domains in the top layer of the diagram remain the same—Natural Language, Speech Recognition and Vision—the others have changed. Instead of the broad Problem Solving category seen in Figure 1 for symbolic AI, there are two more focused categories for predictions and recommendation systems, which are the dominant forms of problem solving systems developed today. And, in addition to more traditional robotics, the domain now includes autonomous vehicles to highlight recent projects in self-driving cars and drones. Finally, since it is now the foundation of the AI domains, machine learning is no longer included in the top-level domains.

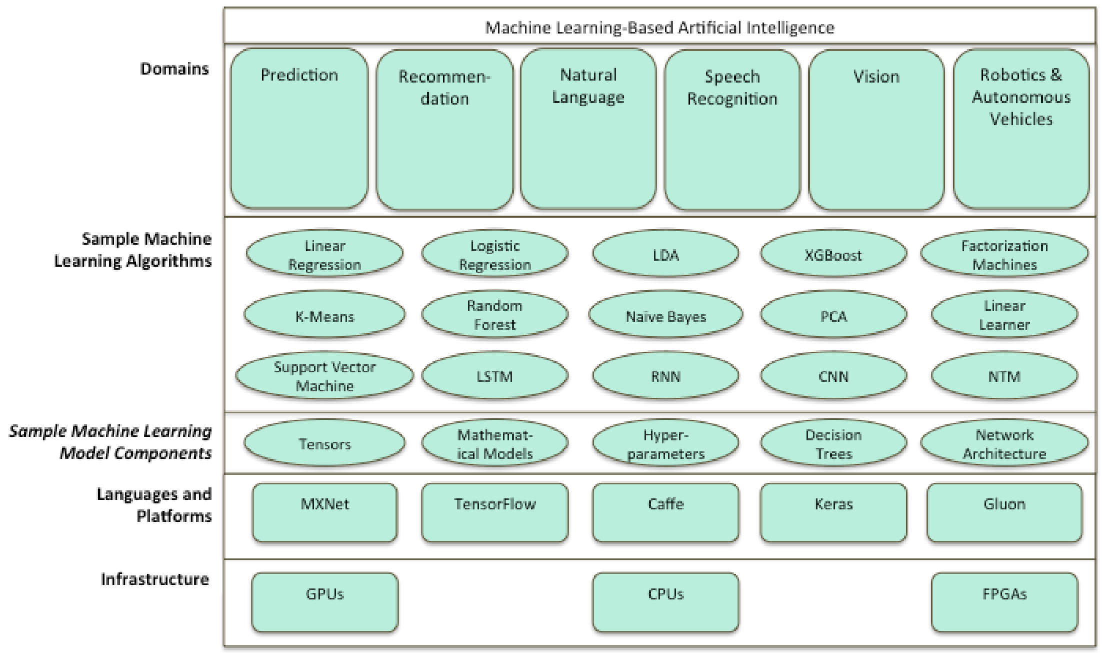

Figure 2: Machine Learning as a foundation for Artificial Intelligence

There are still many questions and challenges for machine learning. The following list provides some of the strengths and limitations of artificial intelligence based on a machine learning foundation.

Table 3: Strengths and Limitations of ML Based AI

Strength	Limitation
Easy to train new solutions given data and tools	Experiencing hype, and researchers and practitioners need to properly set expectations
Large number of diverse algorithms to solve many types of problems	Requires large amounts of clean, potentially labeled data
Solves problems in all AI domains, often approaching or exceeding human level of capability	Problems in data, such as staleness, incompleteness, or adversarial injection of bad data, can skew results
No human expertise or complex knowledge engineering required, solutions are derived from examples	Some, especially statistically-based ML algorithms, rely on manual feature engineering

Strength	Limitation
Deep learning extracts features automatically, which enables complex perception and understanding solutions	System logic is not programmed and must be learned. This can lead to more subjective results, such as competing levels of activation, where precise answers are needed (e.g., specific true or false answers for compliance or verification problems).
Trained ML models can be replicated and reused in ensembles or components of other solutions	Selecting the best algorithm, network architecture, and hyperparameters is more art than science and requires iteration - though tools for hyperparameter optimization are now available
Making predictions or producing results is often faster than traditional inferencing or algorithmic approaches	Training on complex problems with large data sets requires significant time and compute resources
Algorithms for training ML models can be engineered to be distributed and one-pass, improving scalability and reducing training time	It is often difficult to explain how the model derived the results by looking at its structure and results of its training.
Can be trained and deployed on scalable, high-performance infrastructure	Most algorithms solve problems in one step, so no chains of reasoning or partial results are available, though outputs can reflect numeric "confidence"
Deployed using common mechanisms like microservices / APIs for ease of integrations with other systems	

An important take-away from Table 2 and Table 3 is that they are somewhat complementary. ML-based AI can benefit from the strengths of symbolic AI. Some ML approaches, including automatically learning decision trees, already merge the two approaches effectively. Active research continues into other means of combining the strengths of both approaches, as well as many open questions.

Given that today's AI is built on the new foundation of machine learning that has long been the realm of researchers and data scientists, how can we best enable people from different backgrounds in diverse organizations to leverage it?

AWS and Machine Learning

AWS is committed to democratizing machine learning. Our goal is to make machine learning widely accessible to customers with different levels of training and experience, and to organizations across the board. AWS innovates rapidly, creating services and features for customers prioritized by their needs. Machine Learning services are no exception. In the diagram below, you can see how the current AWS Machine Learning services map to the other AI diagrams.

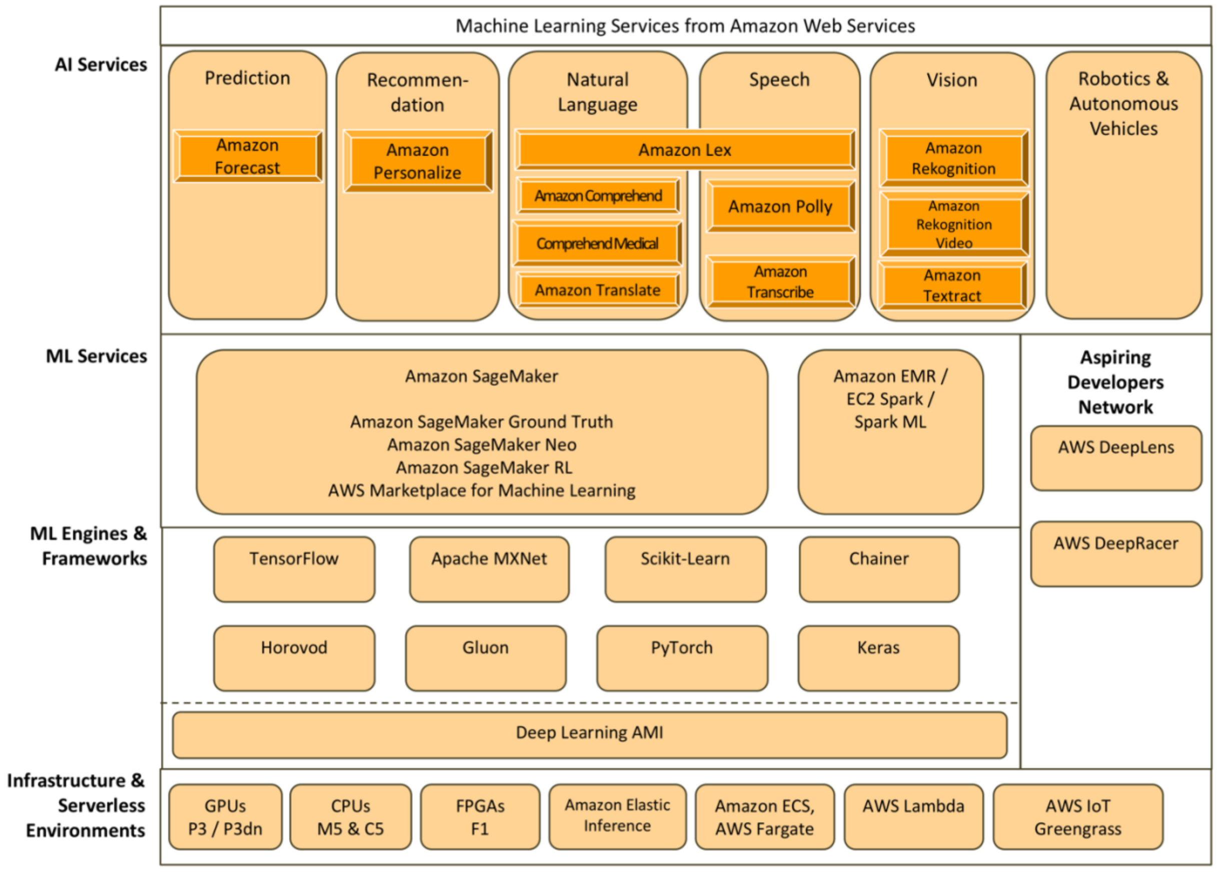

Figure 3: AWS Machine Learning Services

AWS Machine Learning Services for Builders

The first layer shows *AI Services*, which are intended for builders creating specific solutions that require prediction, recommendation, natural language, speech, vision, or other capabilities. These intelligent services are created using machine learning, and especially deep learning models, but do not require the developer to have any knowledge of machine learning to use them. Instead, these capabilities come pre-trained, are accessible via API call and provide customers the ability to add intelligence to their applications.

Amazon Forecast

Amazon Forecast is a fully managed service that delivers highly accurate forecasts, and is based on the same technology used at Amazon.com. You provide historical data plus any additional data that you believe impacts your forecasts. Amazon Forecast examines the data, identifies what is meaningful and produces a forecasting model.

Amazon Personalize

Amazon Personalize makes it easy for developers to create individualized product and content recommendations for customers using their applications. You provide an activity stream from your application, inventory of items you want to recommend and potential demographic information from your users. Amazon Personalize processes and examines the data, identifies what is meaningful, selects the right algorithms, and trains and optimizes a personalization model.

Amazon Lex

Amazon Lex is a service for building conversational interfaces into any application using voice and text. Amazon Lex provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text, to enable you to build applications with highly engaging user experiences and lifelike conversational interactions. With Amazon Lex, the same deep learning technologies that power Amazon Alexa are now available to any developer, enabling you to quickly and easily build sophisticated, natural language, conversational bots ("chatbots").

Amazon Comprehend

Amazon Comprehend is a natural language processing (NLP) service that uses machine learning to find insights and relationships in text. Amazon Comprehend identifies the language of the text; extracts key phrases, places, people, brands, or events; understands how positive or negative the text is and automatically organizes a collection of text files by topic.

Amazon Comprehend Medical

Amazon Comprehend Medical is a natural language processing service that extracts relevant medical information from unstructured text using advanced machine learning models. You can use the extracted medical information and their relationships to build or enhance applications.

Amazon Translate

Amazon Translate is a neural machine translation service that delivers fast, high-quality, and affordable language translation. Neural machine translation is a form of language translation automation that uses deep learning models to deliver more accurate and more natural sounding translation than traditional statistical and rule-based translation algorithms. Amazon Translate allows you to localize content - such as websites and applications - for international users, and to easily translate large volumes of text efficiently.

Amazon Polly

Amazon Polly is a service that turns text into lifelike speech, allowing you to create applications that talk, and build entirely new categories of speech-enabled products. Amazon Polly is a Text-to-Speech service that uses advanced deep learning technologies to synthesize speech that sounds like a human voice.

Amazon Transcribe

Amazon Transcribe is an automatic speech recognition (ASR) service that makes it easy for developers to add speech-to-text capability to their applications. Using the Amazon Transcribe API, you can analyze audio files stored in Amazon S3 and have the service return a text file of the transcribed speech.

Amazon Rekognition

Amazon Rekognition makes it easy to add image and video analysis to your applications. You just provide an image or video to the Rekognition API, and the service can identify the objects, people, text, scenes, and activities, as well as detect any inappropriate content. Amazon Rekognition also provides highly accurate facial analysis and facial recognition. You can detect, analyze, and compare faces for a wide variety of user verification, cataloging, people counting, and public safety use cases.

Amazon Textract

Amazon Textract automatically extracts text and data from scanned documents and forms, going beyond simple optical character recognition to identify contents of fields in forms and information stored in tables.

AWS Machine Learning Services for Custom ML Models

The *ML Services* layer in Figure 3 provides more access to managed services and resources used by developers, data scientists, researchers and other customers to create their own custom ML models. Custom ML models address tasks such as inferencing and prediction, recommender systems and guiding autonomous vehicles.

Amazon SageMaker

Amazon SageMaker is a fully-managed machine learning (ML) service that enables developers and data scientists to quickly and easily build, train, and deploy machine learning models at any scale. Amazon SageMaker Ground Truth helps build training data sets quickly and accurately using an active learning model to label data, combining machine learning and human interaction to make the model progressively better.

SageMaker provides fully-managed and pre-built Jupyter notebooks to address common use cases. The services come with multiple built-in, high-performance algorithms, and there is the AWS Marketplace for Machine Learning containing more than 100 additional pre-trained ML models and algorithms. You can also bring your own algorithms and frameworks that are built into a Docker container.

Amazon Sagemaker includes built-in, fully managed Reinforcement Learning (RL) algorithms. RL is ideal for situations where there is not pre-labeled historical data, but there is an optimal outcome. RL trains using rewards and penalties, which direct the model toward the desired behavior. SageMaker supports RL in multiple frameworks, including TensorFlow and MXNet, as well as custom developed frameworks.

SageMaker sets up and manages environments for training, and provides hyperparameter optimization with Automatic Model Tuning to make the model as accurate as possible. Sagemaker Neo allows you to deploy the same trained model to multiple platforms. Using machine learning, Neo optimizes the performance and size of the model and deploys to edge devices containing the Neo runtime. AWS has released the code as the open source Neo-AI project on GitHub under the Apache Software License. SageMaker deployments run models spread across availability zones to deliver high performance and high availability.

Amazon EMR/EC2 with Spark/Spark ML

Amazon EMR provides a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable Amazon EC2 instances. You can also run other popular distributed frameworks such as Apache Spark - including the Spark ML machine learning library, HBase, Presto, and Flink in Amazon EMR, and interact with data in other AWS data stores such as Amazon S3 and Amazon DynamoDB. Spark and Spark ML can also be run on Amazon EC2 instances to pre-process data, engineer features or run machine learning models.

Aspiring Developers Framework

In parallel with *ML Services*, is the *Aspiring Developers Framework* layer. With a focus on teaching ML technology and techniques to users, this layer is not intended for production use at scale. Currently, the aspiring developers framework consists of two service offerings.

AWS DeepLens

AWS DeepLens helps put deep learning in the hands of developers, with a fully programmable video camera, tutorials, code, and pre-trained models designed to expand deep learning skills. DeepLens offers developers the opportunity to use neural networks to learn and make predictions through computer vision projects, tutorials, and real world, hands-on exploration with a physical device.

AWS DeepRacer

AWS DeepRacer is a 1/18th scale race car that provides a way to get started with reinforcement learning (RL). AWS DeepRacer provides a means to experiment with and learn about RL by building models in Amazon SageMaker, testing in the simulator and deploying an RL model into the car.

ML Engines and Frameworks

Below the ML Platform layer is the *ML Engines and Frameworks* layer. This layer provides direct, hands-on access to the most popular machine learning tools. In this layer are the AWS Deep Learning AMIs that equip you with the infrastructure and tools to accelerate deep learning in the cloud. The AMIs package together several important tools and frameworks, and are pre-installed with Apache MXNet, TensorFlow, PyTorch, the Microsoft Cognitive Toolkit (CNTK), Caffe, Caffe2, Theano, Torch, Gluon, Chainer and Keras to train sophisticated, custom AI models. The Deep Learning AMIs let you create managed, auto-scaling clusters of GPUs for large scale training, or run inference on trained models with compute-optimized or general-purpose CPU instances.

ML Model Training and Deployment Support

The *Infrastructure & Serverless Environments* layer provides the tools that support the training and deployment of machine learning models. Machine learning requires a broad set of powerful compute options, ranging from GPUs for compute-intensive deep learning, to FPGAs for specialized hardware acceleration, to high-memory instances for running inference.

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2 provides a wide selection of instance types optimized to fit machine learning use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources, whether you are training models or running inference on trained models.

Amazon Elastic Inference

Amazon Elastic Inference allows you to attach low-cost GPU-powered acceleration to Amazon EC2 and Amazon SageMaker instances for making predictions with your model. Rather than attaching a full GPU, which is more than required for most models, Elastic Inference can provide savings of up to 75% by allowing separate configuration of the right amount of acceleration for the specific model.

Amazon Elastic Container Service (Amazon ECS)

Amazon ECS supports running and scaling containerized applications, including trained machine learning models from Amazon SageMaker and containerized Spark ML.

Serverless Options

Serverless options remove the burden of managing specific infrastructure, and allow customers to focus on deploying the ML models and other logic necessary to run their systems. Some of the serverless ML deployment options provided by AWS include Amazon SageMaker model deployment, AWS Fargate for containers, and AWS Lambda for serverless code deployment.

ML at the Edge

AWS also provides an option for pushing ML models to the edge to run locally on connected devices using Amazon SageMaker Neo and AWS IoT Greengrass ML Inference. This allows customers to use ML models that are built and trained in the cloud and deploy and run ML inference locally on connected devices.

Conclusions

Many people use the terms AI and ML interchangeably. On the surface, this seems incorrect, because historically machine learning is just a domain inside of AI and AI covers a much broader set of systems.

Today, the algorithms and models of machine learning replace traditional symbolic inferencing, knowledge representations and languages. Training on large data sets has replaced hand-coded algorithms and heuristic approaches. Problems that seemed intractable using symbolic AI methods are modeled consistently with remarkable results using this approach. Machine learning has in fact become the foundation of most modern AI systems. Therefore, it actually makes more sense today than ever for the terms AI and ML to be used interchangeably

AWS provides several machine learning offerings, ranging from pre-trained, ready-to-use services, to the most popular tools and frameworks for creating custom ML models. Customers across industries and with varying levels of experience can add ML capabilities to improve existing systems, as well as create leading-edge applications in areas that were not previously accessible.

Contributors

Contributors to this document include:

- David Bailey, Cloud Infrastructure Architect, Amazon Web Services
- Mark Roy, Solutions Architect, Amazon Web Services
- Denis Batalov, Tech Leader, ML & AI, Amazon Web Services

Further Reading

For additional information, see:

- AWS Whitepapers page
- AWS Machine Learning page
- AWS Machine Learning Training
- AWS Documentation

Document Revisions

Date	Description
February 2019	First publication