Understanding the major types of machine learning
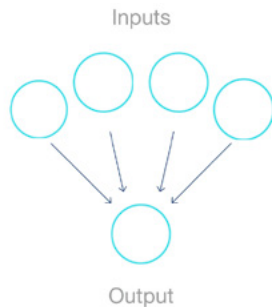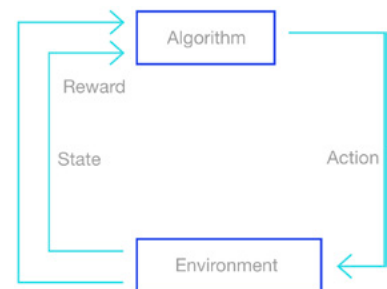
	Supervised learning	Unsupervised learning	Reinforcement learning
What it is	An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output (eg, how the inputs "time of year" and "interest rates" predict housing prices)	An algorithm explores input data without being given an explicit output variable (eg, explores customer demographic data to identify patterns)	An algorithm learns to perform a task simply by trying to maximize rewards it receives for its actions (eg, maximizes points it receives for increasing returns of an investment portfolio)
When to use it	You know how to classify the input data and the type of behavior you want to predict, but you need the algorithm to calculate it for you on new data	You do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you	You don't have a lot of training data; you cannot clearly define the ideal end state; or the only way to learn about the environment is to interact with it
How it works	1. A human labels every element of the input data (eg, in the case of predicting housing prices, labels the input data as "time of year," "interest rates," etc) and defines the output variable (eg, housing prices) 2. The algorithm is trained on the data to find the connection between the input variables and the output 3. Once training is complete—typically when the algorithm is sufficiently accurate—the algorithm is applied to new data	1. The algorithm receives unlabeled data (eg, a set of data describing customer journeys on a website) 2. It infers a structure from the data 3. The algorithm identifies groups of data that exhibit similar behavior (eg, forms clusters of customers that exhibit similar buying behaviors)	1. The algorithm takes an action on the environment (eg, makes a trade in a financial portfolio) 2. It receives a reward if the action brings the machine a step closer to maximizing the total rewards available (eg, the highest total return on the portfolio) 3. The algorithm optimizes for the best series of actions by correcting itself over time

Supervised learning: Algorithms and sample business use cases[1]

Algorithms	Sample business use cases
Linear regression Highly interpretable, standard method for model-ing the past relationship between independent input variables and dependent output variables (which can have an infinite number of values) to help predict future values of the output variables	■ Understand product-sales drivers such as competition prices, distribution, advertisement, etc ■ Optimize price points and estimate product-price elasticities
Logistic regression Extension of linear regression that's used for classifation tasks, meaning the output variable is binary (eg, only black or white) rather than continuous (eg, an infinite list of potential colors)	■ Classify customers based on how likely they are to repay a loan ■ Predict if a skin lesion is benign or malignant based on its characteristics (size, shape, color, etc)
Linear/quadratic discriminant analysis Upgrades a logistic regression to deal with nonlinear problems—those in which changes to the value of input variables do not result in proportional changes to the output variables.	■ Predict client churn ■ Predict a sales lead's likelihood of closing
Decision tree Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (eg, if a feature is a color, each possible color becomes a new branch) until a final decision output is made	■ Provide a decision framework for hiring new employees ■ Understand product attributes that make a product most likely to be purchased
Naive Bayes Classification technique that applies Bayes theorem, which allows the probability of an event to be calculated based on knowledge of factors that might affect that event (eg, if an email contains the word "money," then the probability of it being spam is high)	■ Analyze sentiment to assess product perception in the market ■ Create classifiers to filter spam emails

---

[1] We've listed some of the most commonly used algorithms today—this list is not intended to be exhaustive. Additionally, a number of different models can often solve the same business problem. Conversely, the nature of an available data set often precludes using a model typically employed to solve a particular problem. For these reasons, the sample business use cases are meant only to be illustrative of the types of problems these models can solve.

Algorithms	Sample business use cases
Support vector machine A technique that's typically used for classification but can be transformed to perform regression. It draws an optimal division between classes (as wide as possible). It also can be quickly generalized to solve nonlinear problems	▪ Predict how many patients a hospital will need to serve in a time period ▪ Predict how likely someone is to click on an online ad
Random forest Classification or regression model that improves the accuracy of a simple decision tree by generating multiple decision trees and taking a majority vote of them to predict the output, which is a continuous variable (eg, age) for a regression problem and a discrete variable (eg, either black, white, or red) for classification	▪ Predict call volume in call centers for staffing decisions ▪ Predict power usage in an electrical-distribution grid
AdaBoost Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	▪ Detect fraudulent activity in credit-card transactions. Achieves lower accuracy than deep learning ▪ Simple, low-cost way to classify images (eg, recognize land usage from satellite images for climate-change models). Achieves lower accuracy than deep learning
Gradient-boosting trees Classification or regression technique that generates decision trees sequentially, where each tree focuses on correcting the errors coming from the previous tree model. The final output is a combination of the results from all trees	▪ Forecast product demand and inventory levels ▪ Predict the price of cars based on their characteristics (eg, age and mileage)
Simple neural network Model in which artificial neurons (software-based calculators) make up three layers (an input layer, a hidden layer where calculations take place, and an output layer) that can be used to classify data or find the relationship between variables in regression problems	▪ Predict the probability that a patient joins a healthcare program ▪ Predict whether registered users will be willing or not to pay a particular price for a product

An executive's guide to AI

Unsupervised learning: Algorithms and sample business use cases[2]

Algorithms	Sample business use cases
K-means clustering Puts data into a number of groups (k) that each contain data with similar characteristics (as determined by the model, not in advance by humans)	■ Segment customers into groups by distinct charateristics (eg, age group)—for instance, to better assign marketing campaigns or prevent churn
Gaussian mixture model A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters)	■ Segment customers to better assign marketing campaigns using less-distinct customer characteristics (eg, product preferences) ■ Segment employees based on likelihood of attrition
Hierarchical clustering Splits or aggregates clusters along a hierarchical tree to form a classification system	■ Cluster loyalty-card customers into progressively more microsegmented groups ■ Inform product usage/development by grouping customers mentioning keywords in social-media data
Recommender system Often uses cluster behavior prediction to identify the important data necessary for making a recommendation	■ Recommend what movies consumers should view based on preferences of other customers with similar attributes ■ Recommend news articles a reader might want to read based on the article she or he is reading

---

[2] We've listed some of the most commonly used algorithms today—this list is not intended to be exhaustive. Additionally, a number of different models can often solve the same business problem. Conversely, the nature of an available data set often precludes using a model typically employed to solve a particular problem. For these reasons, the sample business use cases are meant only to be illustrative of the types of problems these models can solve.

An executive's guide to AI

## Reinforcement learning: Sample business use cases[3]

- Optimize the trading strategy for an options-trading portfolio

- Balance the load of electricity grids in varying demand cycles

- Stock and pick inventory using robots

- Optimize the driving behavior of self-driving cars

- Optimize pricing in real time for an online auction of a product with limited supply

Deep learning: A definition

Deep learning is a type of machine learning that can process a wider range of data resources, requires less data preprocessing by humans, and can often produce more accurate results than traditional machine-learning approaches. In deep learning, interconnected layers of software-based calculators known as "neurons" form a neural network. The network can ingest vast amounts of input data and process them through multiple layers that learn increasingly complex features of the data at each layer. The network can then make a determination about the data, learn if its determination is correct, and use what it has learned to make determinations about new data. For example, once it learns what an object looks like, it can recognize the object in a new image.

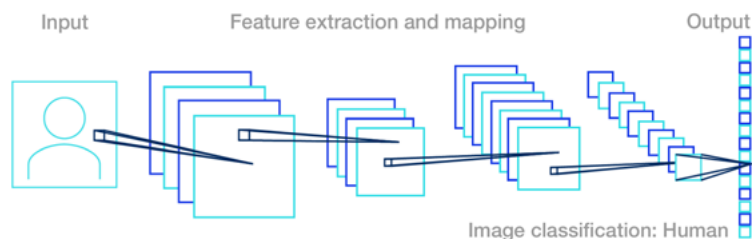Deep learning can often outperform traditional methods

% reduction in error rate achieved by deep learning vs traditional methods
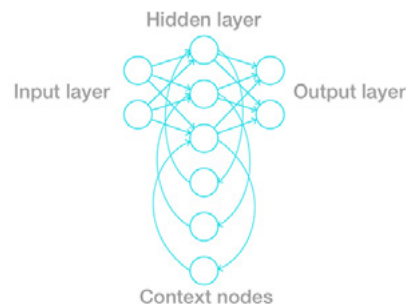
Image classification	41%
Facial recognition	27%
Voice recognition	25%

---
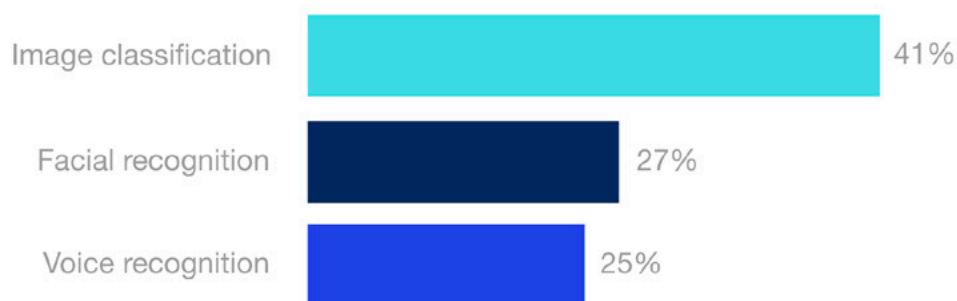
[3] The sample business use cases are meant only to be illustrative of the types of problems these models can solve.

Understanding the major deep learning models and their business use cases[4]

Convolutional neural network

Input

Feature extraction and mapping

Output

Image classification: Human

Recurrent neural network

Hidden layer

Input layer

Output layer

Context nodes

What it is

A multilayered neural network with a special architecture designed to extract increasingly complex features of the data at each layer to determine the output

A multilayered neural network that can store information in context nodes, allowing it to learn data sequences and output a number or another sequence

---

When to use it

When you have an unstructured data set (eg, images) and you need to infer information from it

When you are working with time-series data or sequences (eg, audio recordings or text)

---

[4] The sample business use cases are meant only to be illustrative of the types of problems these models can solve.

Convolutional neural network

Recurrent neural network

How it works

Processing an image

1. The convolutional neural network (CNN) receives an image—for example, of the letter "A"—that it processes as a collection of pixels

2. In the hidden, inner layers of the model, it identifies unique features, for example, the individual lines that make up "A"

3. The CNN can now classify a different image as the letter "A" if it finds in it the unique features previously identified as making up the letter

Predicting the next word in the sentence "Are you free _____ ?"

1. A recurrent neural network (RNN) neuron receives a command that indicates the start of a sentence

2. The neuron receives the word "Are" and then outputs a vector of numbers that feeds back into the neuron to help it "remember" that it received "Are" (and that it received it first). The same process occurs when it receives "you" and "free," with the state of the neuron updating upon receiving each word

3. After receiving "free," the neuron assigns a probability to every word in the English vocabulary that could complete the sentence. If trained well, the RNN will assign the word "tomorrow" one of the highest probabilities and will choose it to complete the sentence

Business use cases

- Diagnose health diseases from medical scans

- Detect a company logo in social media to better understand joint marketing opportunities (eg, pairing of brands in one product)

- Understand customer brand perception and usage through images

- Detect defective products on a production line through images

- Generate analyst reports for securities traders

- Provide language translation

- Track visual changes to an area after a disaster to assess potential damage claims (in conjunction with CNNs)

- Assess the likelihood that a credit-card transaction is fraudulent

- Generate captions for images

- Power chatbots that can address more nuanced customer needs and inquiries

Timeline: Why AI now?

A convergence of algorithmic advances, data proliferation, and tremendous increases in computing power and storage has propelled AI from hype to reality.

Algorithmic advancements

1805 – Legendre lays the groundwork for machine learning
French mathematician Adrien-Marie Legendre publishes the least square method for regression, which he used to determine, from astronomical observations, the orbits of bodies around the sun. Although this method was developed as a statistical framework, it would provide the basis for many of today's machine-learning models.

1958 – Rosenblatt develops the first self-learning algorithm
American psychologist and computer scientist Frank Rosenblatt creates the perceptron algorithm, an early type of artificial neural network (ANN), which stands as the first algorithmic model that could learn on its own. American computer scientist Arthur Samuel would coin the term "machine learning" the following year for these types of self-learning models (as well as develop a groundbreaking checkers program seen as an early success in AI).

1965 – Birth of deep learning
Ukrainian mathematician Alexey Grigorevich Ivakhnenko develops the first general working learning algorithms for supervised multilayer artificial neural networks (ANNs), in which several ANNs are stacked on top of one another and the output of one ANN layer feeds into the next. The architecture is very similar to today's deep-learning architectures.

1986 – Backpropagation takes hold
American psychologist David Rumelhart, British cognitive psychologist and computer scientist Geoffrey Hinton, and American computer scientist Ronald Williams publish on backpropagation, popularizing this key technique for training artificial neural networks (ANNs) that was originally proposed by American scientist Paul Werbos in 1982. Backpropagation allows the ANN to optimize itself without human intervention (in this case, it found features in family-tree data that weren't obvious or provided to the algorithm in advance). Still, lack of computational power and the massive amounts of data needed to train these multilayered networks prevent ANNs leveraging backpropagation from being used widely

Explosion of data

1991 – Opening of the World Wide Web
The European Organization for Nuclear Research (CERN) begins opening up the World Wide Web to the public.

Early 2000s – Broadband adoption begins among home Internet users
Broadband allows users access to increasingly speedy Internet connections, up from the paltry 56 kbps available for downloading through dial-up in the late 1990s. Today, available broadband speeds can surpass 100 mbps (1 mbps = 1,000 kbps). Bandwidth-hungry applications like YouTube could not have become commercially viable without the advent of broadband.

Exponential increases in computing power and storage

1965 – Moore recognizes exponential growth in chip power
Intel cofounder Gordon Moore notices that the number of transistors per square inch on integrated circuits has doubled every year since their invention. His observation becomes Moore's law, which predicts the trend will continue into the foreseeable future (although it later proves to do so roughly every 18 months). At the time, state-of-the-art computational speed is in the order of three million floating-point operations per second (FLOPS).

1997 – Increase in computing power drives IBM's Deep Blue victory over Garry Kasparov
Deep Blue's success against the world chess champion largely stems from masterful engineering and the tremendous power computers possess at that time. Deep Blue's computer achieves around 11 gigaFLOPS (11 billion FLOPS).

1999 – More computing power for AI algorithms arrives ... but no one realizes it yet
Nvidia releases the GeForce 256 graphics card, marketed as the world's first true graphics processing unit (GPU). The technology will later prove fundamental to deep learning by performing computations much faster than computer processing units (CPUs).

1989 – Birth of CNNs for image recognition
French computer scientist Yann LeCun, now director of AI research for Facebook, and others publish a paper describing how a type of artificial neural network called a convolutional neural network (CNN) is well suited for shape-recognition tasks. LeCun and team apply CNNs to the task of recognizing handwritten characters, with the initial goal of building automatic mail-sorting machines. Today, CNNs are the state-of-the-art model for image recognition and classification.

1992 – Upgraded SVMs provide early natural-language-processing solution
Computer engineers Bernhard E. Boser (Swiss), Isabelle M. Guyon (French), and Russian mathematician Vladimir N. Vapnik discover that algorithmic models called support vector machines (SVMs) can be easily upgraded to deal with nonlinear problems by using a technique called kernel trick, leading to widespread usage of SVMs in many natural-language-processing problems, such as classifying sentiment and understanding human speech.

1997 – RNNs get a "memory," positioning them to advance speech to text
In 1991, German computer scientist Sepp Hochreiter showed that a special type of artificial neural network (ANN) called a recurrent neural network (RNN) can be useful in sequencing tasks (speech to text, for example) if it could remember the behavior of part sequences better. In 1997, Hochreiter and fellow computer scientist Jürgen Schmidhuber solve the problem by developing long short-term memory (LSTM). Today, RNNs with LSTM are used in many major speech-recognition applications.

1998 – Brin and Page publish PageRank algorithm
The algorithm, which ranks web pages higher the more other web pages link to them, forms the initial prototype of Google's search engine. This brainchild of Google founders Sergey Brin and Larry Page revolutionizes Internet searches, opening the door to the creation and consumption of more content and data on the World Wide Web. The algorithm would also go on to become one of the most important for businesses as they vie for attention on an increasingly sprawling Internet.

2004 – Facebook debuts
Harvard student Mark Zuckerberg and team launch "Thefacebook," as it was originally dubbed. By the end of 2005, the number of data-generating Facebook users approaches six million.

2004 – Web 2.0 hits its stride, launching the era of user-generated data
Web 2.0 refers to the shifting of the Internet paradigm from passive content viewing to interactive and collaborative content creation, social media, blogs, video, and other channels. Publishers Tim O'Reilly and Dale Dougherty popularize the term, though it was coined by designer Darcy DiNucci in 1999.

2005 – YouTube debuts
Within about 18 months, the site would serve up almost 100 million views per day.

2005 – Number of Internet users worldwide passes one-billion mark

2002 — Amazon brings cloud storage and computing to the masses
Amazon launches its Amazon Web Services, offering cloud-based storage and computing power to users. Cloud computing would come to revolutionize and democratize data storage and computation, giving millions of users access to powerful IT systems—previously only available to big tech companies—at a relatively low cost.

2004 – Dean and Ghemawat introduce the MapReduce algorithm to cope with data explosion
With the World Wide Web taking off, Google seeks out novel ideas to deal with the resulting proliferation of data. Computer scientist Jeff Dean (current head of Google Brain) and Google software engineer Sanjay Ghemawat develop MapReduce to deal with immense amounts of data by parallelizing processes across large data sets using a substantial number of computers.

2005 – Cost of one gigabyte of disk storage drops to $0.79, from $277 ten years earlier
And the price of DRAM, a type of random-access memory (RAM) commonly used in PCs, drops to $158 per gigabyte, from $31,633 in 1995.

2006 – Hinton reenergizes the use of deep-learning models
To speed the training of deep-learning models, Geoffrey Hinton develops a way to pretrain them with a deep-belief network (a class of neural network) before employing backpropagation. While his method would become obsolete when computational power increased to a level that allowed for efficient deep-learning-model training, Hinton's work popularized the use of deep learning worldwide—and many credit him with coining the phrase "deep learning."

2007 – Introduction of the iPhone propels smartphone revolution—and amps up data generation
Apple cofounder and CEO Steve Jobs introduces the iPhone in January 2007. The total number of smartphones sold in 2007 reaches about 122 million. The era of around-the-clock consumption and creation of data and content by smartphone users begins.

2006 – Cutting and Cafarella introduce Hadoop to store and process massive amounts of data
Inspired by Google's MapReduce, computer scientists Doug Cutting and Mike Cafarella develop the Hadoop software to store and process enormous data sets. Yahoo uses it first, to deal with the explosion of data coming from indexing web pages and online data.

2009 – UC Berkeley introduces Spark to handle big data models more efficiently
Developed by Romanian-Canadian computer scientist Matei Zaharia at UC Berkeley's AMPLab, Spark streams huge amounts of data leveraging RAM, making it much faster at processing data than software that must read/write on hard drives. It revolutionizes the ability to update big data and perform analytics in real time.

2009 – Ng uses GPUs to train deep-learning models more effectively
American computer scientist Andrew Ng and his team at Stanford University show that training deep-belief networks with 100 million parameters on GPUs is more than 70 times faster than doing so on CPUs, a finding that would reduce training that once took weeks to only one day.

2010 – Number of smartphones sold in the year nears 300 million
This represents a nearly 2.5 times increase over the number sold in 2007.

2010 – Microsoft and Google introduce their clouds
Cloud computing and storage take another step toward ubiquity when Microsoft makes Azure available and Google launches its Google Cloud Storage (the Google Cloud Platform would come online about a year later).

2010 – Worldwide IP traffic exceeds 20 exabytes (20 billion gigabytes) per month
Internet protocol (IP) traffic is aided by growing adoption of broadband, particularly in the United States, where adoption reaches 65 percent, according to Cisco, which reports this monthly figure and the annual figure of 242 exabytes.

2011 – IBM Watson beats *Jeopardy!*
IBM's question answering system, Watson, defeats the two greatest *Jeopardy!* champions, Brad Rutter and Ken Jennings, by a significant margin. IBM Watson uses ten racks of IBM Power 750 servers capable of 80 teraFLOPS (that's 80 trillion FLOPS—the state of the art in the mid-1960s was around three million FLOPS).

2012 – Number of Facebook users hits one billion
The amount of data processed by the company's systems soars past 500 terabytes.

2012 – Deep-learning system wins renowned image-classification contest for the first time
Geoffrey Hinton's team wins ImageNet's image-classification competition by a large margin, with an error rate of 15.3 percent versus the second-best error rate of 26.2 percent, using a convolutional neural network (CNN). Hinton's team trained its CNN on 1.2 million images using

2012 – Google demonstrates the effectiveness of deep learning for image recognition
Google uses 16,000 processors to train a deep artificial neural network with one billion connections on ten million randomly selected YouTube video thumbnails over the course of three days. Without receiving any information about the images, the network starts recognizing pictures of cats, marking the beginning of significant advances in image recognition.

2013 – DeepMind teaches an algorithm to play Atari using reinforcement learning and deep learning
While reinforcement learning dates to the late 1950s, it gains in popularity this year when Canadian research scientist Vlad Mnih from DeepMind (not yet a Google company) applies it in conjunction with a convolutional neural network to play Atari video games at superhuman levels.

2014 – Number of mobile devices exceeds number of humans
As of October 2014, GSMA reports the number of mobile devices at around 7.22 billion, while the US Census Bureau reports the number of people globally at around 7.20 billion.

2017 – Electronic-device users generate 2.5 quintillion bytes of data per day
According to this estimate, about 90 percent of the world's data were produced in the past two years. And, every minute, YouTube users watch more than four million videos and mobile users send more than 15 million texts.

2017 – Google introduces upgraded TPU that speeds machine-learning processes
Google first introduced its tensor processing unit (TPU) in 2016, which it used to run its own machine-learning models at a reported 15 to 30 times faster than GPUs and CPUs. In 2017, Google announced an upgraded version of the TPU that was faster (180 million teraFLOPS—more when multiple TPUs are combined), could be used to train models in addition to running them, and would be offered to the paying public via the cloud. TPU availability could spawn even more (and more powerful and efficient) machine-learning-based business applications.

2017 – AlphaZero beats AlphaGo Zero after learning to play three different games in less than 24 hours
While creating AI software with full general intelligence remains decades off (if possible at all), Google's DeepMind takes another step closer to it with AlphaZero, which learns three computer games: Go, chess, and shogi. Unlike AlphaGo Zero, which received some instruction from human experts, AlphaZero learns strictly by playing itself, and then goes on to defeat its predecessor AlphaGo Zero at Go (after eight hours of self-play) as well as some of the world's best chess- and shogi-playing computer programs (after four and two hours of self-play, respectively).