

Machine learning

Machine learning (ML) is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks.^[1] It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so.^[2] Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.^[3]

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers, but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning.^{[5][6]} Some implementations of machine learning use data and neural networks in a way that mimics the working of a biological brain.^{[7][8]} In its application across business problems, machine learning is also referred to as predictive analytics.

Contents

Overview

History and relationships to other fields

Artificial intelligence

Data mining

Optimization

Generalization

Statistics

Theory

Approaches

Supervised learning

Unsupervised learning

Semi-supervised learning

Reinforcement learning

Dimensionality reduction

Other types

Models

Training models

Applications

Limitations

Bias

Overfitting

Other limitations and vulnerabilities

Model assessments

Ethics

Hardware

Neuromorphic/Physical Neural Networks

Embedded Machine Learning

Software

Free and open-source software

Proprietary software with free and open-source editions

Proprietary software

Journals

Conferences

See also

References

Sources

Further reading

External links

Overview

Learning algorithms work on the basis that strategies, algorithms, and inferences that worked well in the past are likely to continue working well in the future. These inferences can be obvious, such as "since the sun rose every morning for the last 10,000 days, it will probably rise tomorrow morning as well". They can be nuanced, such as "X% of families have geographically separate species with color variants, so there is a Y% chance that undiscovered black swans exist".^[9]

Machine learning programs can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.^[10]

The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST dataset of handwritten digits has often been used.^[10]

History and relationships to other fields

The term *machine learning* was coined in 1959 by Arthur Samuel, an IBM employee and pioneer in the field of computer gaming and artificial intelligence.^{[11][12]} Also the synonym *self-teaching computers* were used in this time period.^{[13][14]}

By the early 1960s an experimental "learning machine" with punched tape memory, called Cybertron, had been developed by Raytheon Company to analyze sonar signals, electrocardiograms and speech patterns using rudimentary reinforcement learning. It was repetitively "trained" by a human operator/teacher to recognize patterns and equipped with a "goof" button to cause it to re-evaluate incorrect decisions.^[15] A representative book on research into machine learning during the 1960s was Nilsson's book on Learning Machines, dealing mostly with machine learning for pattern classification.^[16] Interest related to pattern recognition continued into the 1970s, as described by Duda and Hart in 1973.^[17] In 1981 a report was given on using teaching strategies so that a neural network learns to recognize 40 characters (26 letters, 10 digits, and 4 special symbols) from a computer terminal.^[18]

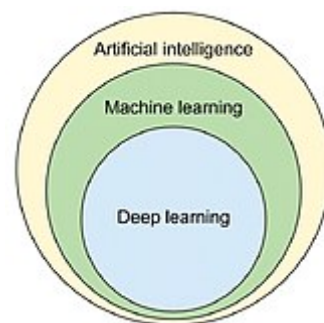
Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T , as measured by P , improves with experience E ."^[19] This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?"^[20]

Modern day machine learning has two objectives, one is to classify data based on models which have been developed, the other purpose is to make predictions for future outcomes based on these models. A hypothetical algorithm specific to classifying data may use computer vision of moles coupled with supervised learning in order to train it to classify the cancerous moles. A machine learning algorithm for stock trading may inform the trader of future potential predictions.^[21]

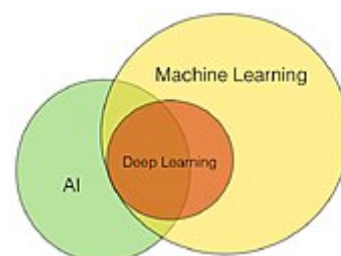
Artificial intelligence

As a scientific endeavor, machine learning grew out of the quest for artificial intelligence. In the early days of AI as an academic discipline, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what was then termed "neural networks"; these were mostly perceptrons and other models that were later found to be reinventions of the generalized linear models of statistics.^[24] Probabilistic reasoning was also employed, especially in automated medical diagnosis.^{[25]:488}

However, an increasing emphasis on the logical, knowledge-based approach caused a rift between AI and machine learning. Probabilistic systems were plagued by theoretical and practical problems of data acquisition and representation.^{[25]:488} By 1980, expert systems had come to dominate AI, and statistics was out of favor.^[26] Work on symbolic/knowledge-based learning did continue within AI, leading to inductive logic programming, but the more statistical line of research was now outside the field of AI proper, in pattern recognition and information retrieval.^{[25]:708–710,755} Neural networks research had been abandoned by AI and computer science around the same time. This line, too, was continued outside the AI/CS field, as "connectionism", by researchers from other disciplines including Hopfield, Rumelhart and Hinton. Their main success came in the mid-1980s with the reinvention of backpropagation.^{[25]:25}



Machine learning as subfield of AI^[22]



Part of machine learning as subfield of AI or part of AI as subfield of machine learning^[23]

Machine learning (ML), reorganized as a separate field, started to flourish in the 1990s. The field changed its goal from achieving artificial intelligence to tackling solvable problems of a practical nature. It shifted focus away from the symbolic approaches it had inherited from AI, and toward methods and models borrowed from statistics, fuzzy logic, and probability theory.^[26]

The difference between ML and AI is frequently misunderstood. ML learns and predicts based on passive observations, whereas AI implies an agent interacting with the environment to learn and take actions that maximize its chance of successfully achieving its goals.^[27]

As of 2020, many sources continue to assert that ML remains a subfield of AI.^{[28][29][26]} Others have the view that not all ML is part of AI, but only an 'intelligent subset' of ML should be considered AI.^{[4][30][31]}

Data mining

Machine learning and data mining often employ the same methods and overlap significantly, but while machine learning focuses on prediction, based on *known* properties learned from the training data, data mining focuses on the discovery of (previously) *unknown* properties in the data (this is the analysis step of knowledge discovery in databases). Data mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as "unsupervised learning" or as a preprocessing step to improve learner accuracy. Much of the confusion between these two research communities (which do often have separate conferences and separate journals, ECML PKDD being a major exception) comes from the basic assumptions they work with: in machine learning, performance is usually evaluated with respect to the ability to *reproduce known* knowledge, while in knowledge discovery and data mining (KDD) the key task is the discovery of previously *unknown* knowledge. Evaluated with respect to known knowledge, an uninformed (unsupervised) method will easily be outperformed by other supervised methods, while in a typical KDD task, supervised methods cannot be used due to the unavailability of training data.

Optimization

Machine learning also has intimate ties to optimization: many learning problems are formulated as minimization of some loss function on a training set of examples. Loss functions express the discrepancy between the predictions of the model being trained and the actual problem instances (for example, in classification, one wants to assign a label to instances, and models are trained to correctly predict the pre-assigned labels of a set of examples).^[32]

Generalization

The difference between optimization and machine learning arises from the goal of generalization: while optimization algorithms can minimize the loss on a training set, machine learning is concerned with minimizing the loss on unseen samples. Characterizing the generalization of various learning algorithms is an active topic of current research, especially for deep learning algorithms.

Statistics

Machine learning and statistics are closely related fields in terms of methods, but distinct in their principal goal: statistics draws population inferences from a sample, while machine learning finds generalizable predictive patterns.^[33] According to Michael I. Jordan, the ideas of machine learning, from methodological

principles to theoretical tools, have had a long pre-history in statistics.^[34] He also suggested the term data science as a placeholder to call the overall field.^[34]

Leo Breiman distinguished two statistical modeling paradigms: data model and algorithmic model,^[28] wherein "algorithmic model" means more or less the machine learning algorithms like Random forest.

Some statisticians have adopted methods from machine learning, leading to a combined field that they call *statistical learning*.^[29]

Theory

A core objective of a learner is to generalize from its experience.^{[4][30]} Generalization in this context is the ability of a learning machine to perform accurately on new, unseen examples/tasks after having experienced a learning data set. The training examples come from some generally unknown probability distribution (considered representative of the space of occurrences) and the learner has to build a general model about this space that enables it to produce sufficiently accurate predictions in new cases.

The computational analysis of machine learning algorithms and their performance is a branch of theoretical computer science known as computational learning theory via the Probably Approximately Correct Learning (PAC) model. Because training sets are finite and the future is uncertain, learning theory usually does not yield guarantees of the performance of algorithms. Instead, probabilistic bounds on the performance are quite common. The bias-variance decomposition is one way to quantify generalization error.

For the best performance in the context of generalization, the complexity of the hypothesis should match the complexity of the function underlying the data. If the hypothesis is less complex than the function, then the model has under fitted the data. If the complexity of the model is increased in response, then the training error decreases. But if the hypothesis is too complex, then the model is subject to overfitting and generalization will be poorer.^[31]

In addition to performance bounds, learning theorists study the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time. There are two kinds of time complexity results: Positive results show that a certain class of functions can be learned in polynomial time. Negative results show that certain classes cannot be learned in polynomial time.

Approaches

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

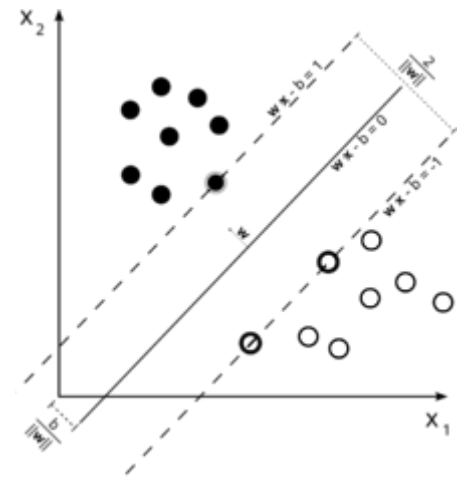
- Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.
- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).
- Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.^[4]

Supervised learning

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs.^[35] The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs.^[36] An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.^[19]

Types of supervised-learning algorithms include active learning, classification and regression.^[27] Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. As an example, for a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email.

Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.



A support-vector machine is a supervised learning model that divides the data into regions separated by a linear boundary. Here, the linear boundary divides the black circles from the white.

Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function.^[37] Though unsupervised learning encompasses other domains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called *clusters*) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some *similarity metric* and evaluated, for example, by *internal compactness*, or the similarity between members of the same cluster, and *separation*, the difference between clusters. Other methods are based on *estimated density* and *graph connectivity*.

Semi-supervised learning

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

In weakly supervised learning, the training labels are noisy, limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger effective training sets.^[38]

Reinforcement learning

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov decision process (MDP). Many reinforcement learning algorithms use dynamic programming techniques.^[39] Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

Dimensionality reduction

Dimensionality reduction is a process of reducing the number of random variables under consideration by obtaining a set of principal variables.^[40] In other words, it is a process of reducing the dimension of the feature set, also called "number of features". Most of the dimensionality reduction techniques can be considered as either feature elimination or extraction. One of the popular methods of dimensionality reduction is principal component analysis (PCA). PCA involves changing higher-dimensional data (e.g., 3D) to a smaller space (e.g., 2D). This results in a smaller dimension of data (2D instead of 3D), while keeping all original variables in the model without changing the data.^[41] The manifold hypothesis proposes that high-dimensional data sets lie along low-dimensional manifolds, and many dimensionality reduction techniques make this assumption, leading to the area of manifold learning and manifold regularization.

Other types

Other approaches have been developed which don't fit neatly into this three-fold categorisation, and sometimes more than one is used by the same machine learning system. For example topic modeling, meta learning.^[42]

As of 2020, deep learning has become the dominant approach for much ongoing work in the field of machine learning.^[10]

Self learning

Self-learning as a machine learning paradigm was introduced in 1982 along with a neural network capable of self-learning named *crossbar adaptive array* (CAA).^[43] It is a learning with no external rewards and no external teacher advice. The CAA self-learning algorithm computes, in a crossbar fashion, both decisions

about actions and emotions (feelings) about consequence situations. The system is driven by the interaction between cognition and emotion.^[44] The self-learning algorithm updates a memory matrix $W = \|w(a,s)\|$ such that in each iteration executes the following machine learning routine:

```
In situation s perform an action a;  
Receive consequence situation s';  
Compute emotion of being in consequence situation v(s');  
Update crossbar memory  $w'(a,s) = w(a,s) + v(s')$ .
```

It is a system with only one input, situation s, and only one output, action (or behavior) a. There is neither a separate reinforcement input nor an advice input from the environment. The backpropagated value (secondary reinforcement) is the emotion toward the consequence situation. The CAA exists in two environments, one is the behavioral environment where it behaves, and the other is the genetic environment, wherefrom it initially and only once receives initial emotions about situations to be encountered in the behavioral environment. After receiving the genome (species) vector from the genetic environment, the CAA learns a goal-seeking behavior, in an environment that contains both desirable and undesirable situations.^[45]

Feature learning

Several learning algorithms aim at discovering better representations of the inputs provided during training.^[46] Classic examples include principal components analysis and cluster analysis. Feature learning algorithms, also called representation learning algorithms, often attempt to preserve the information in their input but also transform it in a way that makes it useful, often as a pre-processing step before performing classification or predictions. This technique allows reconstruction of the inputs coming from the unknown data-generating distribution, while not being necessarily faithful to configurations that are implausible under that distribution. This replaces manual feature engineering, and allows a machine to both learn the features and use them to perform a specific task.

Feature learning can be either supervised or unsupervised. In supervised feature learning, features are learned using labeled input data. Examples include artificial neural networks, multilayer perceptrons, and supervised dictionary learning. In unsupervised feature learning, features are learned with unlabeled input data. Examples include dictionary learning, independent component analysis, autoencoders, matrix factorization^[47] and various forms of clustering.^{[48][49][50]}

Manifold learning algorithms attempt to do so under the constraint that the learned representation is low-dimensional. Sparse coding algorithms attempt to do so under the constraint that the learned representation is sparse, meaning that the mathematical model has many zeros. Multilinear subspace learning algorithms aim to learn low-dimensional representations directly from tensor representations for multidimensional data, without reshaping them into higher-dimensional vectors.^[51] Deep learning algorithms discover multiple levels of representation, or a hierarchy of features, with higher-level, more abstract features defined in terms of (or generating) lower-level features. It has been argued that an intelligent machine is one that learns a representation that disentangles the underlying factors of variation that explain the observed data.^[52]

Feature learning is motivated by the fact that machine learning tasks such as classification often require input that is mathematically and computationally convenient to process. However, real-world data such as images, video, and sensory data has not yielded attempts to algorithmically define specific features. An alternative is to discover such features or representations through examination, without relying on explicit algorithms.

Sparse dictionary learning

Sparse dictionary learning is a feature learning method where a training example is represented as a linear combination of basis functions, and is assumed to be a sparse matrix. The method is strongly NP-hard and difficult to solve approximately.^[53] A popular heuristic method for sparse dictionary learning is the K-SVD algorithm. Sparse dictionary learning has been applied in several contexts. In classification, the problem is to determine the class to which a previously unseen training example belongs. For a dictionary where each class has already been built, a new training example is associated with the class that is best sparsely represented by the corresponding dictionary. Sparse dictionary learning has also been applied in image denoising. The key idea is that a clean image patch can be sparsely represented by an image dictionary, but the noise cannot.^[54]

Anomaly detection

In data mining, anomaly detection, also known as outlier detection, is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data.^[55] Typically, the anomalous items represent an issue such as bank fraud, a structural defect, medical problems or errors in a text. Anomalies are referred to as outliers, novelties, noise, deviations and exceptions.^[56]

In particular, in the context of abuse and network intrusion detection, the interesting objects are often not rare objects, but unexpected bursts of inactivity. This pattern does not adhere to the common statistical definition of an outlier as a rare object, and many outlier detection methods (in particular, unsupervised algorithms) will fail on such data unless it has been aggregated appropriately. Instead, a cluster analysis algorithm may be able to detect the micro-clusters formed by these patterns.^[57]

Three broad categories of anomaly detection techniques exist.^[58] Unsupervised anomaly detection techniques detect anomalies in an unlabeled test data set under the assumption that the majority of the instances in the data set are normal, by looking for instances that seem to fit the least to the remainder of the data set. Supervised anomaly detection techniques require a data set that has been labeled as "normal" and "abnormal" and involves training a classifier (the key difference to many other statistical classification problems is the inherently unbalanced nature of outlier detection). Semi-supervised anomaly detection techniques construct a model representing normal behavior from a given normal training data set and then test the likelihood of a test instance to be generated by the model.

Robot learning

Robot learning is inspired by a multitude of machine learning methods, starting from supervised learning, reinforcement learning,^{[59][60]} and finally meta-learning (e.g. MAML).

Association rules

Association rule learning is a rule-based machine learning method for discovering relationships between variables in large databases. It is intended to identify strong rules discovered in databases using some measure of "interestingness".^[61]

Rule-based machine learning is a general term for any machine learning method that identifies, learns, or evolves "rules" to store, manipulate or apply knowledge. The defining characteristic of a rule-based machine learning algorithm is the identification and utilization of a set of relational rules that collectively represent the knowledge captured by the system. This is in contrast to other machine learning algorithms that commonly identify a singular model that can be universally applied to any instance in order to make a prediction.^[62] Rule-based machine learning approaches include learning classifier systems, association rule learning, and artificial immune systems.

Based on the concept of strong rules, Rakesh Agrawal, Tomasz Imieliński and Arun Swami introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets.^[63] For example, the rule $\{\text{onions, potatoes}\} \Rightarrow \{\text{burger}\}$ found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, they are likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as promotional pricing or product placements. In addition to market basket analysis, association rules are employed today in application areas including Web usage mining, intrusion detection, continuous production, and bioinformatics. In contrast with sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.

Learning classifier systems (LCS) are a family of rule-based machine learning algorithms that combine a discovery component, typically a genetic algorithm, with a learning component, performing either supervised learning, reinforcement learning, or unsupervised learning. They seek to identify a set of context-dependent rules that collectively store and apply knowledge in a piecewise manner in order to make predictions.^[64]

Inductive logic programming (ILP) is an approach to rule-learning using logic programming as a uniform representation for input examples, background knowledge, and hypotheses. Given an encoding of the known background knowledge and a set of examples represented as a logical database of facts, an ILP system will derive a hypothesized logic program that entails all positive and no negative examples. Inductive programming is a related field that considers any kind of programming language for representing hypotheses (and not only logic programming), such as functional programs.

Inductive logic programming is particularly useful in bioinformatics and natural language processing. Gordon Plotkin and Ehud Shapiro laid the initial theoretical foundation for inductive machine learning in a logical setting.^{[65][66][67]} Shapiro built their first implementation (Model Inference System) in 1981: a Prolog program that inductively inferred logic programs from positive and negative examples.^[68] The term *inductive* here refers to philosophical induction, suggesting a theory to explain observed facts, rather than mathematical induction, proving a property for all members of a well-ordered set.

Models

Performing machine learning involves creating a model, which is trained on some training data and then can process additional data to make predictions. Various types of models have been used and researched for machine learning systems.

Artificial neural networks

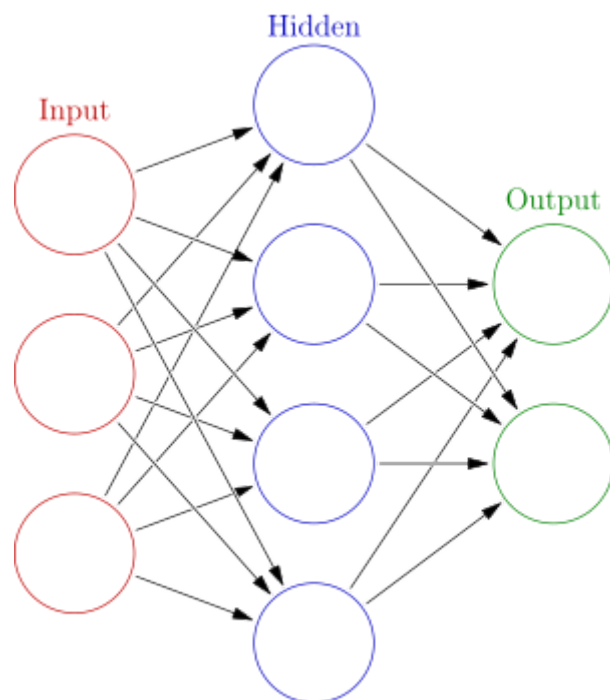
Artificial neural networks (ANNs), or connectionist systems, are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules.

An ANN is a model based on a collection of connected units or nodes called "artificial neurons", which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit information, a "signal", from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called "edges". Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection.

Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly after traversing the layers multiple times.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.

Deep learning consists of multiple hidden layers in an artificial neural network. This approach tries to model the way the human brain processes light and sound into vision and hearing. Some successful applications of deep learning are computer vision and speech recognition.^[69]



An artificial neural network is an interconnected group of nodes, akin to the vast network of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.

Decision trees

Decision tree learning uses a decision tree as a predictive model to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining, and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data, but the resulting classification tree can be an input for decision making.

Support-vector machines

Support-vector machines (SVMs), also known as support-vector networks, are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other.^[70] An SVM training algorithm is a non-probabilistic, binary, linear classifier, although methods such as Platt scaling exist to use SVM in a probabilistic classification setting. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Regression analysis

Regression analysis encompasses a large variety of statistical methods to estimate the relationship between input variables and their associated features. Its most common form is linear regression, where a single line is drawn to best fit the given data according to a mathematical criterion such as ordinary least squares. The latter is often extended by regularization methods to mitigate overfitting and bias, as in ridge regression. When dealing with non-linear problems, go-to models include polynomial regression (for example, used for trendline fitting in Microsoft Excel^[71]), logistic regression (often used in statistical classification) or even kernel regression, which introduces non-linearity by taking advantage of the kernel trick to implicitly map input variables to higher-dimensional space.

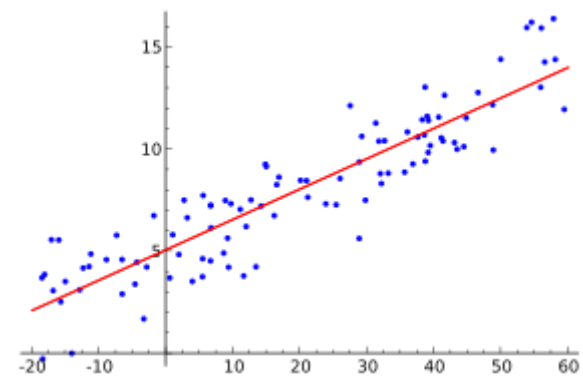
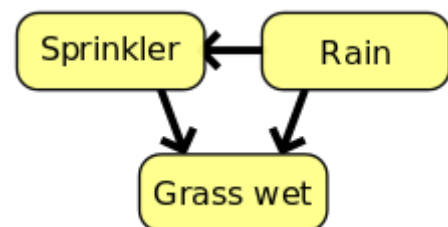


Illustration of linear regression on a data set

Bayesian networks

A Bayesian network, belief network, or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independence with a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Efficient algorithms exist that perform inference and learning. Bayesian networks that model sequences of variables, like speech signals or protein sequences, are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.



A simple Bayesian network. Rain influences whether the sprinkler is activated, and both rain and the sprinkler influence whether the grass is wet.

Genetic algorithms

A genetic algorithm (GA) is a search algorithm and heuristic technique that mimics the process of natural selection, using methods such as mutation and crossover to generate new genotypes in the hope of finding good solutions to a given problem. In machine learning, genetic algorithms were used in the 1980s and 1990s.^{[72][73]} Conversely, machine learning techniques have been used to improve the performance of genetic and evolutionary algorithms.^[74]

Training models

Typically, machine learning models require a high quantity of reliable data in order for the models to perform accurate predictions. When training a machine learning model, machine learning engineers need to target and collect a large and representative sample of data. Data from the training set can be as varied as a corpus of text, a collection of images, sensor data, and data collected from individual users of a service. Overfitting is something to watch out for when training a machine learning model. Trained models derived from biased or non-evaluated data can result in skewed or undesired predictions. Bias models may result in detrimental outcomes thereby furthering the negative impacts on society or objectives. Algorithmic bias is a potential result of data not being fully prepared for training. Machine learning ethics is becoming a field of study and notably be integrated within machine learning engineering teams.

Federated learning

Federated learning is an adapted form of distributed artificial intelligence to training machine learning models that decentralizes the training process, allowing for users' privacy to be maintained by not needing to send their data to a centralized server. This also increases efficiency by decentralizing the training process to many devices. For example, Gboard uses federated machine learning to train search query prediction models on users' mobile phones without having to send individual searches back to Google.^[75]

Applications

There are many applications for machine learning, including:

- Agriculture
- Anatomy
- Adaptive website
- Affective computing
- Astronomy
- Automated decision-making
- Banking
- Bioinformatics
- Brain–machine interfaces
- Cheminformatics
- Citizen science
- Climate science
- Computer networks
- Computer vision
- Credit-card fraud detection
- Data quality
- DNA sequence classification
- Economics
- Financial market analysis^[76]
- General game playing
- Handwriting recognition
- Information retrieval
- Insurance
- Internet fraud detection
- Knowledge graph embedding
- Linguistics
- Machine learning control
- Machine perception
- Machine translation
- Marketing
- Medical diagnosis
- Natural language processing
- Natural language understanding
- Online advertising

- Optimization
- Recommender systems
- Robot locomotion
- Search engines
- Sentiment analysis
- Sequence mining
- Software engineering
- Speech recognition
- Structural health monitoring
- Syntactic pattern recognition
- Telecommunication
- Theorem proving
- Time-series forecasting
- User behavior analytics
- Behaviorism

In 2006, the media-services provider Netflix held the first "Netflix Prize" competition to find a program to better predict user preferences and improve the accuracy of its existing Cinematch movie recommendation algorithm by at least 10%. A joint team made up of researchers from AT&T Labs-Research in collaboration with the teams Big Chaos and Pragmatic Theory built an ensemble model to win the Grand Prize in 2009 for \$1 million.^[77] Shortly after the prize was awarded, Netflix realized that viewers' ratings were not the best indicators of their viewing patterns ("everything is a recommendation") and they changed their recommendation engine accordingly.^[78] In 2010 The Wall Street Journal wrote about the firm Rebellion Research and their use of machine learning to predict the financial crisis.^[79] In 2012, co-founder of Sun Microsystems, Vinod Khosla, predicted that 80% of medical doctors' jobs would be lost in the next two decades to automated machine learning medical diagnostic software.^[80] In 2014, it was reported that a machine learning algorithm had been applied in the field of art history to study fine art paintings and that it may have revealed previously unrecognized influences among artists.^[81] In 2019 Springer Nature published the first research book created using machine learning.^[82] In 2020, machine learning technology was used to help make diagnoses and aid researchers in developing a cure for COVID-19.^[83] Machine learning is recently applied to predict the green behavior of human-being.^[84] Recently, machine learning technology is also applied to optimise smartphone's performance and thermal behaviour based on the user's interaction with the phone.^{[85][86][87]}

Limitations

Although machine learning has been transformative in some fields, machine-learning programs often fail to deliver expected results.^{[88][89][90]} Reasons for this are numerous: lack of (suitable) data, lack of access to the data, data bias, privacy problems, badly chosen tasks and algorithms, wrong tools and people, lack of resources, and evaluation problems.^[91]

In 2018, a self-driving car from Uber failed to detect a pedestrian, who was killed after a collision.^[92] Attempts to use machine learning in healthcare with the IBM Watson system failed to deliver even after years of time and billions of dollars invested.^{[93][94]}

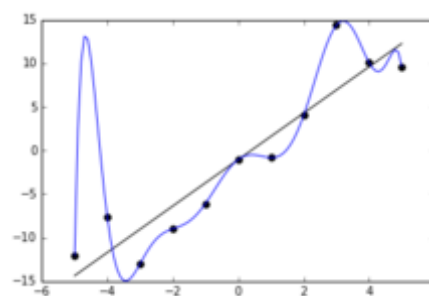
Machine learning has been used as a strategy to update the evidence related to a systematic review and increased reviewer burden related to the growth of biomedical literature. While it has improved with training sets, it has not yet developed sufficiently to reduce the workload burden without limiting the necessary sensitivity for the findings research themselves.^[95]

Bias

Machine learning approaches in particular can suffer from different data biases. A machine learning system trained specifically on current customers may not be able to predict the needs of new customer groups that are not represented in the training data. When trained on man-made data, machine learning is likely to pick up the constitutional and unconscious biases already present in society.^[96] Language models learned from data have been shown to contain human-like biases.^{[97][98]} Machine learning systems used for criminal risk assessment have been found to be biased against black people.^{[99][100]} In 2015, Google photos would often tag black people as gorillas,^[101] and in 2018 this still was not well resolved, but Google reportedly was still using the workaround to remove all gorillas from the training data, and thus was not able to recognize real gorillas at all.^[102] Similar issues with recognizing non-white people have been found in many other systems.^[103] In 2016, Microsoft tested a chatbot that learned from Twitter, and it quickly picked up racist and sexist language.^[104] Because of such challenges, the effective use of machine learning may take longer to be adopted in other domains.^[105] Concern for fairness in machine learning, that is, reducing bias in machine learning and propelling its use for human good is increasingly expressed by artificial intelligence scientists, including Fei-Fei Li, who reminds engineers that "There's nothing artificial about AI...It's inspired by people, it's created by people, and—most importantly—it impacts people. It is a powerful tool we are only just beginning to understand, and that is a profound responsibility."^[106]

Overfitting

Settling on a bad, overly complex theory gerrymandered to fit all the past training data is known as overfitting. Many systems attempt to reduce overfitting by rewarding a theory in accordance with how well it fits the data, but penalizing the theory in accordance with how complex the theory is.^[9]



The blue line could be an example of overfitting a linear function due to random noise.

Other limitations and vulnerabilities

Learners can also disappoint by "learning the wrong lesson". A toy example is that an image classifier trained only on pictures of brown horses and black cats might conclude that all brown patches are likely to be horses.^[107] A real-world example is that, unlike humans, current image classifiers often don't primarily make judgments from the spatial relationship between components of the picture, and they learn relationships between pixels that humans are oblivious to, but that still correlate with images of certain types of real objects. Modifying these patterns on a legitimate image can result in "adversarial" images that the system misclassifies.^{[108][109]}

Adversarial vulnerabilities can also result in nonlinear systems, or from non-pattern perturbations. Some systems are so brittle that changing a single adversarial pixel predictably induces misclassification. Machine learning models are often vulnerable to manipulation and/or evasion via adversarial machine learning.^[110]

Researchers have demonstrated how backdoors can be placed undetectably into classifying (e.g. for categories "spam" and well-visible "not spam" of posts) machine learning models which are often developed and/or trained by third parties. Parties can change the classification of any input, including in cases for which a type of data/software transparency is provided, possibly including white-box access.^{[111][112][113]}

Model assessments

Classification of machine learning models can be validated by accuracy estimation techniques like the holdout method, which splits the data in a training and test set (conventionally 2/3 training set and 1/3 test set designation) and evaluates the performance of the training model on the test set. In comparison, the K-fold-cross-validation method randomly partitions the data into K subsets and then K experiments are performed each respectively considering 1 subset for evaluation and the remaining K-1 subsets for training the model. In addition to the holdout and cross-validation methods, bootstrap, which samples n instances with replacement from the dataset, can be used to assess model accuracy.^[114]

In addition to overall accuracy, investigators frequently report sensitivity and specificity meaning True Positive Rate (TPR) and True Negative Rate (TNR) respectively. Similarly, investigators sometimes report the false positive rate (FPR) as well as the false negative rate (FNR). However, these rates are ratios that fail to reveal their numerators and denominators. The total operating characteristic (TOC) is an effective method to express a model's diagnostic ability. TOC shows the numerators and denominators of the previously mentioned rates, thus TOC provides more information than the commonly used receiver operating characteristic (ROC) and ROC's associated area under the curve (AUC).^[115]

Ethics

Machine learning poses a host of ethical questions. Systems which are trained on datasets collected with biases may exhibit these biases upon use (algorithmic bias), thus digitizing cultural prejudices.^[116] For example, in 1988, the UK's Commission for Racial Equality found that St. George's Medical School had been using a computer program trained from data of previous admissions staff and this program had denied nearly 60 candidates who were found to be either women or had non-European sounding names.^[96] Using job hiring data from a firm with racist hiring policies may lead to a machine learning system duplicating the bias by scoring job applicants by similarity to previous successful applicants.^{[117][118]} Responsible collection of data and documentation of algorithmic rules used by a system thus is a critical part of machine learning.

AI can be well-equipped to make decisions in technical fields, which rely heavily on data and historical information. These decisions rely on the objectivity and logical reasoning.^[119] Because human languages contain biases, machines trained on language corpora will necessarily also learn these biases.^{[120][121]}

Other forms of ethical challenges, not related to personal biases, are seen in health care. There are concerns among health care professionals that these systems might not be designed in the public's interest but as income-generating machines.^[122] This is especially true in the United States where there is a long-standing ethical dilemma of improving health care, but also increase profits. For example, the algorithms could be designed to provide patients with unnecessary tests or medication in which the algorithm's proprietary owners hold stakes. There is potential for machine learning in health care to provide professionals an additional tool to diagnose, medicate, and plan recovery paths for patients, but this requires these biases to be mitigated.^[123]

Hardware

Since the 2010s, advances in both machine learning algorithms and computer hardware have led to more efficient methods for training deep neural networks (a particular narrow subdomain of machine learning) that contain many layers of non-linear hidden units.^[124] By 2019, graphic processing units (GPUs), often with AI-specific enhancements, had displaced CPUs as the dominant method of training large-scale commercial cloud AI.^[125] OpenAI estimated the hardware computing used in the largest deep learning projects from AlexNet (2012) to AlphaZero (2017), and found a 300,000-fold increase in the amount of compute required, with a doubling-time trendline of 3.4 months.^{[126][127]}

Neuromorphic/Physical Neural Networks

A physical neural network or Neuromorphic computer is a type of artificial neural network in which an electrically adjustable material is used to emulate the function of a neural synapse. "Physical" neural network is used to emphasize the reliance on physical hardware used to emulate neurons as opposed to software-based approaches. More generally the term is applicable to other artificial neural networks in which a memristor or other electrically adjustable resistance material is used to emulate a neural synapse.^{[128][129]}

Embedded Machine Learning

Embedded Machine Learning is a sub-field of machine learning, where the machine learning model is run on embedded systems with limited computing resources such as wearable computers, edge devices and microcontrollers.^{[130][131][132]} Running machine learning model in embedded devices removes the need for transferring and storing data on cloud servers for further processing, henceforth, reducing data breaches and privacy leaks happening because of transferring data, and also minimizes theft of intellectual properties, personal data and business secrets. Embedded Machine Learning could be applied through several techniques including hardware acceleration,^{[133][134]} using approximate computing,^[135] optimization of machine learning models and many more.^{[136][137]}

Software

Software suites containing a variety of machine learning algorithms include the following:

Free and open-source software

- Caffe
- Microsoft Cognitive Toolkit
- Deeplearning4j
- DeepSpeed
- ELKI
- Infer.NET
- Keras
- LightGBM
- Mahout
- Mallet
- ML.NET
- mlpack
- MXNet
- Neural Lab
- OpenNN
- Orange
- pandas (software)
- ROOT (TMVA with ROOT)
- scikit-learn
- Shogun
- Spark MLlib
- SystemML
- TensorFlow
- Torch / PyTorch
- Weka / MOA
- XGBoost
- Yooreeka

Proprietary software with free and open-source editions

- KNIME
- RapidMiner

Proprietary software

- [Amazon Machine Learning](#)
- [Angoss KnowledgeSTUDIO](#)
- [Azure Machine Learning](#)
- [Ayasdi](#)
- [IBM Watson Studio](#)
- [Google Prediction API](#)
- [IBM SPSS Modeler](#)
- [KXEN Modeler](#)
- [LIONsolver](#)
- [Mathematica](#)
- [MATLAB](#)
- [Neural Designer](#)
- [NeuroSolutions](#)
- [Oracle Data Mining](#)
- [Oracle AI Platform Cloud Service](#)
- [PolyAnalyst](#)
- [RCASE](#)
- [SAS Enterprise Miner](#)
- [SequenceL](#)
- [Splunk](#)
- [STATISTICA Data Miner](#)

Journals

- [Journal of Machine Learning Research](#)
- [Machine Learning](#)
- [Nature Machine Intelligence](#)
- [Neural Computation](#)

Conferences

- [AAAI Conference on Artificial Intelligence](#)
- [Association for Computational Linguistics \(ACL\)](#)
- [European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases \(ECML PKDD\)](#)
- [International Conference on Computational Intelligence Methods for Bioinformatics and Biostatistics \(CIBB\)](#)
- [International Conference on Machine Learning \(ICML\)](#)
- [International Conference on Learning Representations \(ICLR\)](#)
- [International Conference on Intelligent Robots and Systems \(IROS\)](#)
- [Conference on Knowledge Discovery and Data Mining \(KDD\)](#)
- [Conference on Neural Information Processing Systems \(NeurIPS\)](#)

See also

- [Automated machine learning](#) – Process of automating the application of machine learning
- [Big data](#) – Information assets characterized by high volume, velocity, and variety
- [Differentiable programming](#) – Programming paradigm
- [List of important publications in machine learning](#)
- [List of datasets for machine-learning research](#)

References

1. Mitchell, Tom (1997). *Machine Learning* (<http://www.cs.cmu.edu/~tom/mlbook.html>). New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892 (<https://www.worldcat.org/oclc/36417892>)

417892).

2. The definition "without being explicitly programmed" is often attributed to Arthur Samuel, who coined the term "machine learning" in 1959, but the phrase is not found verbatim in this publication, and may be a paraphrase that appeared later. Confer "Paraphrasing Arthur Samuel (1959), the question is: How can computers learn to solve problems without being explicitly programmed?" in Koza, John R.; Bennett, Forrest H.; Andre, David; Keane, Martin A. (1996). "Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming". *Artificial Intelligence in Design '96*. Artificial Intelligence in Design '96. Springer, Dordrecht. pp. 151–170. doi:10.1007/978-94-009-0279-4_9 (https://doi.org/10.1007%2F978-94-009-0279-4_9). ISBN 978-94-010-6610-5.
3. Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F., "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning (<https://ieeexplore.ieee.org/document/9244647>)" IEEE Transactions on Vehicular Technology, 2020.
4. Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, Springer, ISBN 978-0-387-31073-2
5. Machine learning and pattern recognition "can be viewed as two facets of the same field." [4]: vii
6. Friedman, Jerome H. (1998). "Data Mining and Statistics: What's the connection?". *Computing Science and Statistics*. **29** (1): 3–9.
7. "What is Machine Learning?" (<https://www.ibm.com/cloud/learn/machine-learning>). *www.ibm.com*. Retrieved 2021-08-15.
8. Zhou, Victor (2019-12-20). "Machine Learning for Beginners: An Introduction to Neural Networks" (<https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9>). *Medium*. Retrieved 2021-08-15.
9. Domingos 2015, Chapter 6, Chapter 7.
10. Ethem Alpaydin (2020). *Introduction to Machine Learning* (Fourth ed.). MIT. pp. xix, 1–3, 13–18. ISBN 978-0262043793.
11. Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". *IBM Journal of Research and Development*. **3** (3): 210–229. CiteSeerX 10.1.1.368.2254 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.368.2254>). doi:10.1147/rd.33.0210 (<https://doi.org/10.1147%2Frd.33.0210>).
12. R. Kohavi and F. Provost, "Glossary of terms," Machine Learning, vol. 30, no. 2–3, pp. 271–274, 1998.
13. Gerovitch, Slava (9 April 2015). "How the Computer Got Its Revenge on the Soviet Union" (<https://nautilus.us/issue/23/dominoes/how-the-computer-got-its-revenge-on-the-soviet-union>). *Nautilus*. Retrieved 19 September 2021.
14. Lindsay, Richard P. (1 September 1964). "The Impact of Automation On Public Administration" (<https://journals.sagepub.com/doi/10.1177/106591296401700364>). *Western Political Quarterly*. **17** (3): 78–81. doi:10.1177/106591296401700364 (<https://doi.org/10.1177%2F106591296401700364>). ISSN 0043-4078 (<https://www.worldcat.org/issn/0043-4078>). S2CID 154021253 (<https://api.semanticscholar.org/CorpusID:154021253>). Retrieved 6 October 2021.
15. "Science: The Goof Button," *Time (magazine)*, 18 August 1961.
16. Nilsson N. Learning Machines, McGraw Hill, 1965.
17. Duda, R., Hart P. Pattern Recognition and Scene Analysis, Wiley Interscience, 1973
18. S. Bozinovski "Teaching space: A representation concept for adaptive pattern classification" COINS Technical Report No. 81-28, Computer and Information Science Department, University of Massachusetts at Amherst, MA, 1981. <https://web.cs.umass.edu/publication/docs/1981/UM-CS-1981-028.pdf>

19. Mitchell, T. (1997). *Machine Learning*. McGraw Hill. p. 2. ISBN 978-0-07-042807-2.
20. Harnad, Stevan (2008), "The Annotation Game: On Turing (1950) on Computing, Machinery, and Intelligence" (<https://web.archive.org/web/20120309113922/http://eprints.ecs.soton.ac.uk/12954/>), in Epstein, Robert; Peters, Grace (eds.), *The Turing Test Sourcebook: Philosophical and Methodological Issues in the Quest for the Thinking Computer*, Kluwer, pp. 23–66, ISBN 9781402067082, archived from the original (<http://eprints.ecs.soton.ac.uk/12954/>) on 2012-03-09, retrieved 2012-12-11
21. "Introduction to AI Part 1" (<https://edzion.com/2020/12/09/introduction-to-ai-part-1/>). Edzion. 2020-12-08. Retrieved 2020-12-09.
22. "AN EMPIRICAL SCIENCE RESEARCH ON BIOINFORMATICS IN MACHINE LEARNING – Journal" (http://www.journalimcms.org/special_issue/an-empirical-science-research-on-bioinformatics-in-machine-learning/). Retrieved 28 October 2020.
23. "rasbt/stat453-deep-learning-ss20" (https://github.com/rasbt/stat453-deep-learning-ss20/blob/master/L01-intro/L01-intro_slides.pdf) (PDF). *GitHub*. 9 November 2021.
24. Sarle, Warren (1994). "Neural Networks and statistical models". *CiteSeerX* 10.1.1.27.699 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.699>).
25. Russell, Stuart; Norvig, Peter (2003) [1995]. *Artificial Intelligence: A Modern Approach* (2nd ed.). Prentice Hall. ISBN 978-0137903955.
26. Langley, Pat (2011). "The changing science of machine learning" (<https://doi.org/10.1007%2Fs10994-011-5242-y>). *Machine Learning*. **82** (3): 275–279. doi:10.1007/s10994-011-5242-y (<https://doi.org/10.1007%2Fs10994-011-5242-y>).
27. Alpaydin, Ethem (2010). *Introduction to Machine Learning* (<https://books.google.com/books?id=7f5bBAAQBAJ>). MIT Press. p. 9. ISBN 978-0-262-01243-0.
28. Cornell University Library (August 2001). "Breiman: Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)" (http://projecteuclid.org/download/pdf_1/euclid.ss/1009213726). *Statistical Science*. **16** (3). doi:10.1214/ss/1009213726 (<https://doi.org/10.1214%2Fss%2F1009213726>). S2CID 62729017 (<https://api.semanticscholar.org/CorpusID:62729017>). Retrieved 8 August 2015.
29. Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani (2013). *An Introduction to Statistical Learning* (<http://www-bcf.usc.edu/~gareth/ISL/>). Springer. p. vii.
30. Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). *Foundations of Machine Learning*. USA, Massachusetts: MIT Press. ISBN 9780262018258.
31. Alpaydin, Ethem (2010). *Introduction to Machine Learning* (https://archive.org/details/introductiontoma00alpa_0). London: The MIT Press. ISBN 978-0-262-01243-0. Retrieved 4 February 2017.
32. Le Roux, Nicolas; Bengio, Yoshua; Fitzgibbon, Andrew (2012). "Improving+First+and+Second-Order+Methods+by+Modeling+Uncertainty&pg=PA403" "Improving First and Second-Order Methods by Modeling Uncertainty" (<https://books.google.com/books?id=JPQx7s2L1A8C&q=>). In Sra, Suvrit; Nowozin, Sebastian; Wright, Stephen J. (eds.). *Optimization for Machine Learning*. MIT Press. p. 404. ISBN 9780262016469.
33. Bzdok, Danilo; Altman, Naomi; Krzywinski, Martin (2018). "Statistics versus Machine Learning" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6082636>). *Nature Methods*. **15** (4): 233–234. doi:10.1038/nmeth.4642 (<https://doi.org/10.1038%2Fnmeth.4642>). PMC 6082636 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6082636>). PMID 30100822 (<https://pubmed.ncbi.nlm.nih.gov/30100822>).
34. Michael I. Jordan (2014-09-10). "statistics and machine learning" (https://www.reddit.com/r/MachineLearning/comments/2fxi6v/ama_michael_i_jordan/ckelmtt?context=3). reddit. Retrieved 2014-10-01.
35. Russell, Stuart J.; Norvig, Peter (2010). *Artificial Intelligence: A Modern Approach* (Third ed.). Prentice Hall. ISBN 9780136042594.

36. Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). *Foundations of Machine Learning*. The MIT Press. ISBN 9780262018258.
37. Jordan, Michael I.; Bishop, Christopher M. (2004). "Neural Networks". In Allen B. Tucker (ed.). *Computer Science Handbook, Second Edition (Section VII: Intelligent Systems)*. Boca Raton, Florida: Chapman & Hall/CRC Press LLC. ISBN 978-1-58488-360-9.
38. Alex Ratner; Stephen Bach; Paroma Varma; Chris. "Weak Supervision: The New Programming Paradigm for Machine Learning" (https://web.archive.org/web/20190606043931/https://hazyresearch.github.io/snorkel/blog/ws_blog_post.html). *hazyresearch.github.io*. referencing work by many other members of Hazy Research. Archived from the original (http://hazyresearch.github.io/snorkel/blog/ws_blog_post.html) on 2019-06-06. Retrieved 2019-06-06.
39. van Otterlo, M.; Wiering, M. (2012). *Reinforcement learning and markov decision processes. Reinforcement Learning. Adaptation, Learning, and Optimization*. Vol. 12. pp. 3–42. doi:10.1007/978-3-642-27645-3_1 (https://doi.org/10.1007%2F978-3-642-27645-3_1). ISBN 978-3-642-27644-6.
40. science.sciencemag.org/content/290/5500/2323
41. towardsdatascience.com/all-machine-learning-models-explained-in-6-minutes-9fe30ff6776a
42. Pavel Brazdil, Christophe Giraud Carrier, Carlos Soares, Ricardo Vilalta (2009). *Metalearning: Applications to Data Mining* (Fourth ed.). Springer Science+Business Media. pp. 10–14, *passim*. ISBN 978-3540732624.
43. Bozinovski, S. (1982). "A self-learning system using secondary reinforcement". In Trappl, Robert (ed.). *Cybernetics and Systems Research: Proceedings of the Sixth European Meeting on Cybernetics and Systems Research*. North Holland. pp. 397–402. ISBN 978-0-444-86488-8.
44. Bozinovski, Stevo (2014) "Modeling mechanisms of cognition-emotion interaction in artificial neural networks, since 1981." *Procedia Computer Science* p. 255-263
45. Bozinovski, S. (2001) "Self-learning agents: A connectionist theory of emotion based on crossbar value judgment." *Cybernetics and Systems* 32(6) 637-667.
46. Y. Bengio; A. Courville; P. Vincent (2013). "Representation Learning: A Review and New Perspectives". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **35** (8): 1798–1828. arXiv:1206.5538 (<https://arxiv.org/abs/1206.5538>). doi:10.1109/tpami.2013.50 (<https://doi.org/10.1109%2Ftpami.2013.50>). PMID 23787338 (<https://pubmed.ncbi.nlm.nih.gov/23787338>). S2CID 393948 (<https://api.semanticscholar.org/CorpusID:393948>).
47. Nathan Srebro; Jason D. M. Rennie; Tommi S. Jaakkola (2004). *Maximum-Margin Matrix Factorization*. NIPS.
48. Coates, Adam; Lee, Honglak; Ng, Andrew Y. (2011). *An analysis of single-layer networks in unsupervised feature learning* (https://web.archive.org/web/20170813153615/http://machinelearning.wustl.edu/mlpapers/paper_files/AISTATS2011_CoatesNL11.pdf) (PDF). Int'l Conf. on AI and Statistics (AISTATS). Archived from the original (http://machinelearning.wustl.edu/mlpapers/paper_files/AISTATS2011_CoatesNL11.pdf) (PDF) on 2017-08-13. Retrieved 2018-11-25.
49. Csurka, Gabriella; Dance, Christopher C.; Fan, Lixin; Willamowski, Jutta; Bray, Cédric (2004). *Visual categorization with bags of keypoints* (<https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/csurka-eccv-04.pdf>) (PDF). ECCV Workshop on Statistical Learning in Computer Vision.
50. Daniel Jurafsky; James H. Martin (2009). *Speech and Language Processing*. Pearson Education International. pp. 145–146.

51. Lu, Haiping; Plataniotis, K.N.; Venetsanopoulos, A.N. (2011). "A Survey of Multilinear Subspace Learning for Tensor Data" (http://www.dsp.utoronto.ca/~haiping/Publication/SurveyMSL_PR2011.pdf) (PDF). *Pattern Recognition*. **44** (7): 1540–1551. Bibcode:2011PatRe..44.1540L (<https://ui.adsabs.harvard.edu/abs/2011PatRe..44.1540L>). doi:10.1016/j.patcog.2011.01.004 (<https://doi.org/10.1016%2Fj.patcog.2011.01.004>).
52. Yoshua Bengio (2009). *Learning Deep Architectures for AI* (<https://books.google.com/books?id=cq5ewg7FniMC&pg=PA3>). Now Publishers Inc. pp. 1–3. ISBN 978-1-60198-294-0.
53. Tillmann, A. M. (2015). "On the Computational Intractability of Exact and Approximate Dictionary Learning". *IEEE Signal Processing Letters*. **22** (1): 45–49. arXiv:1405.6664 (<http://arxiv.org/abs/1405.6664>). Bibcode:2015ISPL...22...45T (<https://ui.adsabs.harvard.edu/abs/2015ISPL...22...45T>). doi:10.1109/LSP.2014.2345761 (<https://doi.org/10.1109%2FLSP.2014.2345761>). S2CID 13342762 (<https://api.semanticscholar.org/CorpusID:13342762>).
54. Aharon, M, M Elad, and A Bruckstein. 2006. "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation (<http://sites.fas.harvard.edu/~cs278/papers/ksvd.pdf>)." *Signal Processing, IEEE Transactions on* 54 (11): 4311–4322
55. Zimek, Arthur; Schubert, Erich (2017), "Outlier Detection", *Encyclopedia of Database Systems*, Springer New York, pp. 1–5, doi:10.1007/978-1-4899-7993-3_80719-1 (https://doi.org/10.1007%2F978-1-4899-7993-3_80719-1), ISBN 9781489979933
56. Hodge, V. J.; Austin, J. (2004). "A Survey of Outlier Detection Methodologies" (<http://eprints.whiterose.ac.uk/767/1/hodgevj4.pdf>) (PDF). *Artificial Intelligence Review*. **22** (2): 85–126. CiteSeerX 10.1.1.318.4023 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.318.4023>). doi:10.1007/s10462-004-4304-y (<https://doi.org/10.1007%2Fs10462-004-4304-y>). S2CID 59941878 (<https://api.semanticscholar.org/CorpusID:59941878>).
57. Dokas, Paul; Ertöz, Levent; Kumar, Vipin; Lazarevic, Aleksandar; Srivastava, Jaideep; Tan, Pang-Ning (2002). "Data mining for network intrusion detection" (<http://www.csee.umbc.edu/~kolari1/Mining/ngdm/dokas.pdf>) (PDF). *Proceedings NSF Workshop on Next Generation Data Mining*.
58. Chandola, V.; Banerjee, A.; Kumar, V. (2009). "Anomaly detection: A survey". *ACM Computing Surveys*. **41** (3): 1–58. doi:10.1145/1541880.1541882 (<https://doi.org/10.1145%2F1541880.1541882>). S2CID 207172599 (<https://api.semanticscholar.org/CorpusID:207172599>).
59. Fleer, S.; Moringen, A.; Klatzky, R. L.; Ritter, H. (2020). "Learning efficient haptic shape exploration with a rigid tactile sensor array, S. Fleer, A. Moringen, R. Klatzky, H. Ritter" (<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC6940144>). *PLOS ONE*. **15** (1): e0226880. arXiv:1902.07501 (<https://arxiv.org/abs/1902.07501>). doi:10.1371/journal.pone.0226880 (<https://doi.org/10.1371%2Fjournal.pone.0226880>). PMC 6940144 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6940144>). PMID 31896135 (<https://pubmed.ncbi.nlm.nih.gov/31896135>).
60. Moringen, Alexandra; Fleer, Sascha; Walck, Guillaume; Ritter, Helge (2020), Nisky, Ilana; Hartcher-O'Brien, Jess; Wiertlewski, Michaël; Smeets, Jeroen (eds.), "Attention-Based Robot Learning of Haptic Interaction" (http://link.springer.com/10.1007/978-3-030-58147-3_51), *Haptics: Science, Technology, Applications*, Cham: Springer International Publishing, vol. 12272, pp. 462–470, doi:10.1007/978-3-030-58147-3_51 (https://doi.org/10.1007%2F978-3-030-58147-3_51), ISBN 978-3-030-58146-6, S2CID 220069113 (<https://api.semanticscholar.org/CorpusID:220069113>), retrieved 2022-01-19
61. Piatetsky-Shapiro, Gregory (1991), *Discovery, analysis, and presentation of strong rules*, in Piatetsky-Shapiro, Gregory; and Frawley, William J.; eds., *Knowledge Discovery in Databases*, AAAI/MIT Press, Cambridge, MA.

62. Bassel, George W.; Glaab, Enrico; Marquez, Julietta; Holdsworth, Michael J.; Bacardit, Jaume (2011-09-01). "Functional Network Construction in Arabidopsis Using Rule-Based Machine Learning on Large-Scale Data Sets" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3203449>). *The Plant Cell*. **23** (9): 3101–3116. doi:10.1105/tpc.111.088153 (<https://doi.org/10.1105/tpc.111.088153>). ISSN 1532-298X (<https://www.worldcat.org/issn/1532-298X>). PMC 3203449 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3203449>). PMID 21896882 (<https://pubmed.ncbi.nlm.nih.gov/21896882>).
63. Agrawal, R.; Imieliński, T.; Swami, A. (1993). "Mining association rules between sets of items in large databases". *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93*. p. 207. CiteSeerX 10.1.1.40.6984 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.6984>). doi:10.1145/170035.170072 (<https://doi.org/10.1145/170035.170072>). ISBN 978-0897915922. S2CID 490415 (<https://api.semanticscholar.org/CorpusID:490415>).
64. Urbanowicz, Ryan J.; Moore, Jason H. (2009-09-22). "Learning Classifier Systems: A Complete Introduction, Review, and Roadmap" (<https://doi.org/10.1155/2009/736398>). *Journal of Artificial Evolution and Applications*. **2009**: 1–25. doi:10.1155/2009/736398 (<https://doi.org/10.1155/2009/736398>). ISSN 1687-6229 (<https://www.worldcat.org/issn/1687-6229>).
65. Plotkin G.D. Automatic Methods of Inductive Inference (<https://www.era.lib.ed.ac.uk/bitstream/handle/1842/6656/Plotkin1972.pdf;sequence=1>), PhD thesis, University of Edinburgh, 1970.
66. Shapiro, Ehud Y. Inductive inference of theories from facts (<http://ftp.cs.yale.edu/publications/techreports/tr192.pdf>), Research Report 192, Yale University, Department of Computer Science, 1981. Reprinted in J.-L. Lassez, G. Plotkin (Eds.), Computational Logic, The MIT Press, Cambridge, MA, 1991, pp. 199–254.
67. Shapiro, Ehud Y. (1983). *Algorithmic program debugging*. Cambridge, Mass: MIT Press. ISBN 0-262-19218-7
68. Shapiro, Ehud Y. "The model inference system (<http://dl.acm.org/citation.cfm?id=1623364>).". *Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2*. Morgan Kaufmann Publishers Inc., 1981.
69. Honglak Lee, Roger Grosse, Rajesh Ranganath, Andrew Y. Ng. "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.802&rep=rep1&type=pdf>)" *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.
70. Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks" (<https://doi.org/10.1007/BF00994018>). *Machine Learning*. **20** (3): 273–297. doi:10.1007/BF00994018 (<https://doi.org/10.1007/BF00994018>).
71. Stevenson, Christopher. "Tutorial: Polynomial Regression in Excel" (<https://facultystaff.richmond.edu/~cstevens/301/Excel4.html>). *facultystaff.richmond.edu*. Retrieved 22 January 2017.
72. Goldberg, David E.; Holland, John H. (1988). "Genetic algorithms and machine learning" (https://deepblue.lib.umich.edu/bitstream/2027.42/46947/1/10994_2005_Article_422926.pdf) (PDF). *Machine Learning*. **3** (2): 95–99. doi:10.1007/bf00113892 (<https://doi.org/10.1007/BF00113892>). S2CID 35506513 (<https://api.semanticscholar.org/CorpusID:35506513>).
73. Michie, D.; Spiegelhalter, D. J.; Taylor, C. C. (1994). "Machine Learning, Neural and Statistical Classification". *Ellis Horwood Series in Artificial Intelligence*. Bibcode:1994mlns.book.....M (<https://ui.adsabs.harvard.edu/abs/1994mlns.book.....M>).

74. Zhang, Jun; Zhan, Zhi-hui; Lin, Ying; Chen, Ni; Gong, Yue-jiao; Zhong, Jing-hui; Chung, Henry S.H.; Li, Yun; Shi, Yu-hui (2011). "Evolutionary Computation Meets Machine Learning: A Survey". *Computational Intelligence Magazine*. **6** (4): 68–75. doi:10.1109/mci.2011.942584 (<https://doi.org/10.1109%2Fmci.2011.942584>). S2CID 6760276 (<https://api.semanticscholar.org/CorpusID:6760276>).
75. "Federated Learning: Collaborative Machine Learning without Centralized Training Data" (<http://ai.googleblog.com/2017/04/federated-learning-collaborative.html>). *Google AI Blog*. Retrieved 2019-06-08.
76. Machine learning is included in the CFA Curriculum (discussion is top down); see: Kathleen DeRose and Christophe Le Lanno (2020). "Machine Learning" (<https://www.cfainstitute.org/-/media/documents/study-session/2020-l2-ss3.ashx>).
77. "BelKor Home Page" (<https://web.archive.org/web/20151110062742/http://www2.research.att.com/~volinsky/netflix/>) research.att.com
78. "The Netflix Tech Blog: Netflix Recommendations: Beyond the 5 stars (Part 1)" (<https://web.archive.org/web/20160531002916/http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html>). 2012-04-06. Archived from the original (<http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html>) on 31 May 2016. Retrieved 8 August 2015.
79. Scott Patterson (13 July 2010). "Letting the Machines Decide" (<https://www.wsj.com/articles/SB10001424052748703834604575365310813948080>). *The Wall Street Journal*. Retrieved 24 June 2018.
80. Vinod Khosla (January 10, 2012). "Do We Need Doctors or Algorithms?" (<https://techcrunch.com/2012/01/10/doctors-or-algorithms/>). Tech Crunch.
81. When A Machine Learning Algorithm Studied Fine Art Paintings, It Saw Things Art Historians Had Never Noticed (<https://medium.com/the-physics-arxiv-blog/when-a-machine-learning-algorithm-studied-fine-art-paintings-it-saw-things-art-historians-had-never-b8e4e7bf7d3e>), *The Physics at ArXiv blog*
82. Vincent, James (2019-04-10). "The first AI-generated textbook shows what robot writers are actually good at" (<https://www.theverge.com/2019/4/10/18304558/ai-writing-academic-research-book-springer-nature-artificial-intelligence>). *The Verge*. Retrieved 2019-05-05.
83. Vaishya, Raju; Javaid, Mohd; Khan, Ibrahim Haleem; Haleem, Abid (July 1, 2020). "Artificial Intelligence (AI) applications for COVID-19 pandemic" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7195043>). *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*. **14** (4): 337–339. doi:10.1016/j.dsx.2020.04.012 (<https://doi.org/10.1016%2Fj.dsx.2020.04.012>). PMC 7195043 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7195043>). PMID 32305024 (<https://pubmed.ncbi.nlm.nih.gov/32305024>).
84. Rezapouraghdam, Hamed; Akhshik, Arash; Ramkissoon, Haywantee (March 10, 2021). "Application of machine learning to predict visitors' green behavior in marine protected areas: evidence from Cyprus" (<https://doi.org/10.1080%2F09669582.2021.1887878>). *Journal of Sustainable Tourism*: 1–25. doi:10.1080/09669582.2021.1887878 (<https://doi.org/10.1080%2F09669582.2021.1887878>).
85. Dey, Somdip; Singh, Amit Kumar; Wang, Xiaohang; McDonald-Maier, Klaus (2020-06-15). "User Interaction Aware Reinforcement Learning for Power and Thermal Efficiency of CPU-GPU Mobile MPSoCs" (<https://ieeexplore.ieee.org/document/9116294>). *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*: 1728–1733. doi:10.23919/DATE48585.2020.9116294 (<https://doi.org/10.23919%2FDATE48585.2020.9116294>). ISBN 978-3-9819263-4-7. S2CID 219858480 (<https://api.semanticscholar.org/CorpusID:219858480>).

86. Quested, Tony. "Smartphones get smarter with Essex innovation | Business Weekly | Technology News | Business news | Cambridge and the East of England" (<https://www.businessweekly.co.uk/news/academia-research/smartphones-get-smarter-essex-innovation>). *www.businessweekly.co.uk*. Retrieved 2021-06-17.
87. Williams, Rhiannon (2020-07-21). "Future smartphones 'will prolong their own battery life by monitoring owners' behaviour'" (<https://inews.co.uk/news/technology/future-smartphones-pr-long-battery-life-monitoring-behaviour-558689>). *i.* Retrieved 2021-06-17.
88. "Why Machine Learning Models Often Fail to Learn: QuickTake Q&A" (<https://web.archive.org/web/20170320225010/https://www.bloomberg.com/news/articles/2016-11-10/why-machine-learning-models-often-fail-to-learn-quicktake-q-a>). *Bloomberg.com*. 2016-11-10. Archived from the original (<https://www.bloomberg.com/news/articles/2016-11-10/why-machine-learning-models-often-fail-to-learn-quicktake-q-a>) on 2017-03-20. Retrieved 2017-04-10.
89. "The First Wave of Corporate AI Is Doomed to Fail" (<https://hbr.org/2017/04/the-first-wave-of-corporate-ai-is-doomed-to-fail>). *Harvard Business Review*. 2017-04-18. Retrieved 2018-08-20.
90. "Why the A.I. euphoria is doomed to fail" (<https://venturebeat.com/2016/09/17/why-the-a-i-euphoria-is-doomed-to-fail/>). *VentureBeat*. 2016-09-18. Retrieved 2018-08-20.
91. "9 Reasons why your machine learning project will fail" (<https://www.kdnuggets.com/2018/07/why-machine-learning-project-fail.html>). *www.kdnuggets.com*. Retrieved 2018-08-20.
92. "Why Uber's self-driving car killed a pedestrian" (<https://www.economist.com/the-economist-explains/2018/05/29/why-ubers-self-driving-car-killed-a-pedestrian>). *The Economist*. Retrieved 2018-08-20.
93. "IBM's Watson recommended 'unsafe and incorrect' cancer treatments - STAT" (<https://www.statnews.com/2018/07/25/ibm-watson-recommended-unsafe-incorrect-treatments/>). *STAT*. 2018-07-25. Retrieved 2018-08-21.
94. Hernandez, Daniela; Greenwald, Ted (2018-08-11). "IBM Has a Watson Dilemma" (<https://www.wsj.com/articles/ibm-bet-billions-that-watson-could-improve-cancer-treatment-it-hasnt-worked-1533961147>). *Wall Street Journal*. ISSN 0099-9660 (<https://www.worldcat.org/issn/0099-9660>). Retrieved 2018-08-21.
95. Reddy, Shivani M.; Patel, Sheila; Weyrich, Meghan; Fenton, Joshua; Viswanathan, Meera (2020). "Comparison of a traditional systematic review approach with review-of-reviews and semi-automation as strategies to update the evidence" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7574591>). *Systematic Reviews*. **9** (1): 243. doi:10.1186/s13643-020-01450-2 (<https://doi.org/10.1186/s13643-020-01450-2>). ISSN 2046-4053 (<https://www.worldcat.org/issn/2046-4053>). PMC 7574591 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7574591>). PMID 33076975 (<https://pubmed.ncbi.nlm.nih.gov/33076975>).
96. Garcia, Megan (2016). "Racist in the Machine". *World Policy Journal*. **33** (4): 111–117. doi:10.1215/07402775-3813015 (<https://doi.org/10.1215/07402775-3813015>). ISSN 0740-2775 (<https://www.worldcat.org/issn/0740-2775>). S2CID 151595343 (<https://api.semanticscholar.org/CorpusID:151595343>).
97. Caliskan, Aylin; Bryson, Joanna J.; Narayanan, Arvind (2017-04-14). "Semantics derived automatically from language corpora contain human-like biases". *Science*. **356** (6334): 183–186. arXiv:1608.07187 (<https://arxiv.org/abs/1608.07187>). Bibcode:2017Sci...356..183C (<https://ui.adsabs.harvard.edu/abs/2017Sci...356..183C>). doi:10.1126/science.aal4230 (<https://doi.org/10.1126/science.aal4230>). ISSN 0036-8075 (<https://www.worldcat.org/issn/0036-8075>). PMID 28408601 (<https://pubmed.ncbi.nlm.nih.gov/28408601>). S2CID 23163324 (<https://api.semanticscholar.org/CorpusID:23163324>).

98. Wang, Xinan; Dasgupta, Sanjoy (2016), Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I. (eds.), "An algorithm for L1 nearest neighbor search via monotonic embedding" (<http://papers.nips.cc/paper/6227-an-algorithm-for-l1-nearest-neighbor-search-via-monotonic-embedding.pdf>) (PDF), *Advances in Neural Information Processing Systems 29*, Curran Associates, Inc., pp. 983–991, retrieved 2018-08-20
99. Julia Angwin; Jeff Larson; Lauren Kirchner; Surya Mattu (2016-05-23). "Machine Bias" (<http://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>). *ProPublica*. Retrieved 2018-08-20.
100. "Opinion | When an Algorithm Helps Send You to Prison" (<https://www.nytimes.com/2017/10/26/opinion/algorithm-compass-sentencing-bias.html>). *New York Times*. Retrieved 2018-08-20.
101. "Google apologises for racist blunder" (<https://www.bbc.co.uk/news/technology-33347866>). *BBC News*. 2015-07-01. Retrieved 2018-08-20.
102. "Google 'fixed' its racist algorithm by removing gorillas from its image-labeling tech" (<https://www.theverge.com/2018/1/12/16882408/google-racist-gorillas-photo-recognition-algorithm-ai>). *The Verge*. Retrieved 2018-08-20.
103. "Opinion | Artificial Intelligence's White Guy Problem" (<https://www.nytimes.com/2016/06/26/opinion/sunday/artificial-intelligences-white-guy-problem.html>). *New York Times*. Retrieved 2018-08-20.
104. Metz, Rachel. "Why Microsoft's teen chatbot, Tay, said lots of awful things online" (<https://www.technologyreview.com/s/601111/why-microsoft-accidentally-unleashed-a-neo-nazi-sexbot/>). *MIT Technology Review*. Retrieved 2018-08-20.
105. Simonite, Tom. "Microsoft says its racist chatbot illustrates how AI isn't adaptable enough to help most businesses" (<https://www.technologyreview.com/s/603944/microsoft-ai-isnt-yet-adaptable-enough-to-help-businesses/>). *MIT Technology Review*. Retrieved 2018-08-20.
106. Hempel, Jessi (2018-11-13). "Fei-Fei Li's Quest to Make Machines Better for Humanity" (<https://www.wired.com/story/fei-fei-li-artificial-intelligence-humanity/>). *Wired*. ISSN 1059-1028 (<https://www.worldcat.org/issn/1059-1028>). Retrieved 2019-02-17.
107. Domingos 2015, p. 286.
108. "Single pixel change fools AI programs" (<https://www.bbc.com/news/technology-41845878>). *BBC News*. 3 November 2017. Archived (<https://web.archive.org/web/20180322011306/http://www.bbc.com/news/technology-41845878>) from the original on 22 March 2018. Retrieved 12 March 2018.
109. "AI Has a Hallucination Problem That's Proving Tough to Fix" (<https://www.wired.com/story/ai-has-a-hallucination-problem-thats-proving-tough-to-fix/>). *WIRED*. 2018. Archived (<https://web.archive.org/web/20180312024533/https://www.wired.com/story/ai-has-a-hallucination-problem-thats-proving-tough-to-fix/>) from the original on 12 March 2018. Retrieved 12 March 2018.
110. "Adversarial Machine Learning - CLTC UC Berkeley Center for Long-Term Cybersecurity" (<https://cltc.berkeley.edu/aml/>). *CLTC*.
111. "Machine-learning models vulnerable to undetectable backdoors" (https://www.theregister.com/2022/04/21/machine_learning_models_backdoors/). *The Register*. Retrieved 13 May 2022.
112. "Undetectable Backdoors Plantable In Any Machine-Learning Algorithm" (<https://spectrum.ieee.org/machine-learningbackdoor>). *IEEE Spectrum*. 10 May 2022. Retrieved 13 May 2022.
113. Goldwasser, Shafi; Kim, Michael P.; Vaikuntanathan, Vinod; Zamir, Or (14 April 2022). "Planting Undetectable Backdoors in Machine Learning Models". *arXiv:2204.06974* (<https://arxiv.org/abs/2204.06974>) [cs.LG (<https://arxiv.org/archive/cs.LG>)].

114. Kohavi, Ron (1995). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection" (<http://web.cs.iastate.edu/~jtian/cs573/Papers/Kohavi-IJCAI-95.pdf>) (PDF). *International Joint Conference on Artificial Intelligence*.
115. Pontius, Robert Gilmore; Si, Kangping (2014). "The total operating characteristic to measure diagnostic ability for multiple thresholds". *International Journal of Geographical Information Science*. **28** (3): 570–583. doi:10.1080/13658816.2013.862623 (<https://doi.org/10.1080%2F13658816.2013.862623>). S2CID 29204880 (<https://api.semanticscholar.org/CorpusID:29204880>).
116. Bostrom, Nick (2011). "The Ethics of Artificial Intelligence" (<https://web.archive.org/web/20160304015020/http://www.nickbostrom.com/ethics/artificial-intelligence.pdf>) (PDF). Archived from the original (<http://www.nickbostrom.com/ethics/artificial-intelligence.pdf>) (PDF) on 4 March 2016. Retrieved 11 April 2016.
117. Edionwe, Tolulope. "The fight against racist algorithms" (<https://theoutline.com/post/1571/the-fight-against-racist-algorithms>). *The Outline*. Retrieved 17 November 2017.
118. Jeffries, Adrianne. "Machine learning is racist because the internet is racist" (<https://theoutline.com/post/1439/machine-learning-is-racist-because-the-internet-is-racist>). *The Outline*. Retrieved 17 November 2017.
119. Bostrom, Nick; Yudkowsky, Eliezer (2011). "THE ETHICS OF ARTIFICIAL INTELLIGENCE" (<https://www.nickbostrom.com/ethics/artificial-intelligence.pdf>) (PDF). *Nick Bostrom*.
120. M.O.R. Prates, P.H.C. Avelar, L.C. Lamb (11 Mar 2019). "Assessing Gender Bias in Machine Translation -- A Case Study with Google Translate". arXiv:1809.02208 (<https://arxiv.org/abs/1809.02208>) [cs.CY (<https://arxiv.org/archive/cs.CY>)].
121. Narayanan, Arvind (August 24, 2016). "Language necessarily contains human biases, and so will machines trained on language corpora" (<https://freedom-to-tinker.com/2016/08/24/language-necessarily-contains-human-biases-and-so-will-machines-trained-on-language-corpora/>). *Freedom to Tinker*.
122. Char, Danton S.; Shah, Nigam H.; Magnus, David (2018-03-15). "Implementing Machine Learning in Health Care — Addressing Ethical Challenges" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5962261>). *New England Journal of Medicine*. **378** (11): 981–983. doi:10.1056/NEJMp1714229 (<https://doi.org/10.1056%2FNEJMp1714229>). ISSN 0028-4793 (<https://www.worldcat.org/issn/0028-4793>). PMC 5962261 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5962261>). PMID 29539284 (<https://pubmed.ncbi.nlm.nih.gov/29539284>).
123. Char, D. S.; Shah, N. H.; Magnus, D. (2018). "Implementing Machine Learning in Health Care—Addressing Ethical Challenges" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5962261>). *New England Journal of Medicine*. **378** (11): 981–983. doi:10.1056/nejmp1714229 (<https://doi.org/10.1056%2Fnejmp1714229>). PMC 5962261 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5962261>). PMID 29539284 (<https://pubmed.ncbi.nlm.nih.gov/29539284>).
124. Research, AI (23 October 2015). "Deep Neural Networks for Acoustic Modeling in Speech Recognition" (<http://airesearch.com/ai-research-papers/deep-neural-networks-for-acoustic-modeling-in-speech-recognition/>). *airesearch.com*. Retrieved 23 October 2015.
125. "GPUs Continue to Dominate the AI Accelerator Market for Now" (<https://www.informationweek.com/big-data/ai-machine-learning/gpus-continue-to-dominate-the-ai-accelerator-market-for-now/a/d-id/1336475>). *InformationWeek*. December 2019. Retrieved 11 June 2020.
126. Ray, Tiernan (2019). "AI is changing the entire nature of compute" (<https://www.zdnet.com/article/ai-is-changing-the-entire-nature-of-compute/>). *ZDNet*. Retrieved 11 June 2020.
127. "AI and Compute" (<https://openai.com/blog/ai-and-compute/>). *OpenAI*. 16 May 2018. Retrieved 11 June 2020.

128. "Cornell & NTT's Physical Neural Networks: A "Radical Alternative for Implementing Deep Neural Networks" That Enables Arbitrary Physical Systems Training | Synced" (<https://syncedreview.com/2021/05/27/deepmind-podracers-tpu-based-rl-frameworks-deliver-exceptional-performance-at-low-cost-28/>). 27 May 2021.
129. "Nano-spaghetti to solve neural network power consumption" (https://www.theregister.com/2021/10/05/analogue_neural_network_research/).
130. Fafoutis, Xenofon; Marchegiani, Letizia; Elsts, Atis; Pope, James; Piechocki, Robert; Craddock, Ian (2018-05-07). "Extending the battery lifetime of wearable sensors with embedded machine learning" (https://ieeexplore.ieee.org/abstract/document/8355116?casa_token=LCpUeGLS1e8AAAAA:2OjuJfNwZBnV2pgDxfnEAC-jbrETv_BpTcX35_aFqN6IULF_xu1xbYbVSRpD-zMd4GCUMELyG). *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*: 269–274. doi:10.1109/WF-IoT.2018.8355116 (<https://doi.org/10.1109%2FWF-IoT.2018.8355116>). hdl:1983/b8fdb58b-7114-45c6-82e4-4ab239c1327f (<https://hdl.handle.net/1983%2Fb8fdb58b-7114-45c6-82e4-4ab239c1327f>). ISBN 978-1-4673-9944-9. S2CID 19192912 (<https://api.semanticscholar.org/CorpusID:19192912>).
131. "A Beginner's Guide To Machine learning For Embedded Systems" (<https://analyticsindiamag.com/a-beginners-guide-to-machine-learning-for-embedded-systems/>). *Analytics India Magazine*. 2021-06-02. Retrieved 2022-01-17.
132. Synced (2022-01-12). "Google, Purdue & Harvard U's Open-Source Framework for TinyML Achieves up to 75x Speedups on FPGAs | Synced" (<https://syncedreview.com/2022/01/12/deepmind-podracers-tpu-based-rl-frameworks-deliver-exceptional-performance-at-low-cost-183/>). *syncedreview.com*. Retrieved 2022-01-17.
133. Giri, Davide; Chiu, Kuan-Lin; Di Guglielmo, Giuseppe; Mantovani, Paolo; Carloni, Luca P. (2020-06-15). "ESP4ML: Platform-Based Design of Systems-on-Chip for Embedded Machine Learning" (https://ieeexplore.ieee.org/abstract/document/9116317?casa_token=5lTmgrrbu4AAAAA:v7pDHPEWIRuo2Vk3pU06194PO0-W21UOdyZqADrZxrRdPBZDMLwQrjJSAHUhHtzJmLu_VdgW). *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*: 1049–1054. arXiv:2004.03640 (<https://arxiv.org/abs/2004.03640>). doi:10.23919/DATE48585.2020.9116317 (<https://doi.org/10.23919%2FDATE48585.2020.9116317>). ISBN 978-3-9819263-4-7. S2CID 210928161 (<https://api.semanticscholar.org/CorpusID:210928161>).
134. Louis, Marcia Sahaya; Azad, Zahra; Delshadtehrani, Leila; Gupta, Suyog; Warden, Pete; Reddi, Vijay Janapa; Joshi, Ajay (2019). "Towards Deep Learning using TensorFlow Lite on RISC-V" (<https://edge.seas.harvard.edu/publications/towards-deep-learning-using-tensorflow-lite-risc-v>). *Harvard University*. Retrieved 2022-01-17.
135. Ibrahim, Ali; Osta, Mario; Alameh, Mohamad; Saleh, Moustafa; Chible, Hussein; Valle, Maurizio (2019-01-21). "Approximate Computing Methods for Embedded Machine Learning" (https://ieeexplore.ieee.org/abstract/document/8617877?casa_token=arUW5Oy-tzwAAAAA:l9x6edlfskM6kGNFUNGzAFrjEBv_8kYTz7ERTxtXu9jAqdrYCCdbbwjBdgwXvb6QAH_-0VJJ). *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*: 845–848. doi:10.1109/ICECS.2018.8617877 (<https://doi.org/10.1109%2FICECS.2018.8617877>). ISBN 978-1-5386-9562-3. S2CID 58670712 (<https://api.semanticscholar.org/CorpusID:58670712>).
136. "dblp: TensorFlow Eager: A Multi-Stage, Python-Embedded DSL for Machine Learning" (<https://dblp.org/rec/journals/corr/abs-1903-01855.html>). *dblp.org*. Retrieved 2022-01-17.
137. Branco, Sérgio; Ferreira, André G.; Cabral, Jorge (2019-11-05). "Machine Learning in Resource-Scarce Embedded Systems, FPGAs, and End-Devices: A Survey" (<https://doi.org/10.3390/electronics8111289>). *Electronics*. 8 (11): 1289. doi:10.3390/electronics8111289 (<https://doi.org/10.3390%2Felectronics8111289>). ISSN 2079-9292 (<https://www.worldcat.org/issn/2079-9292>).

Sources

- Domingos, Pedro (September 22, 2015). *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books. ISBN 978-0465065707.
- Nilsson, Nils (1998). *Artificial Intelligence: A New Synthesis* (<https://archive.org/details/artificialintell0000nils>). Morgan Kaufmann. ISBN 978-1-55860-467-4. Archived (<https://web.archive.org/web/20200726131654/https://archive.org/details/artificialintell0000nils>) from the original on 26 July 2020. Retrieved 18 November 2019.
- Russell, Stuart J.; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach* (<http://aima.cs.berkeley.edu/>) (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.
- Poole, David; Mackworth, Alan; Goebel, Randy (1998). *Computational Intelligence: A Logical Approach* (<https://archive.org/details/computationalint00pool>). New York: Oxford University Press. ISBN 978-0-19-510270-3. Archived (<https://web.archive.org/web/20200726131436/https://archive.org/details/computationalint00pool>) from the original on 26 July 2020. Retrieved 22 August 2020.

Further reading

- Nils J. Nilsson, *Introduction to Machine Learning* (<https://ai.stanford.edu/people/nilsson/mlbook.html>).
- Trevor Hastie, Robert Tibshirani and Jerome H. Friedman (2001). *The Elements of Statistical Learning* (<https://web.stanford.edu/~hastie/ElemStatLearn/>), Springer. ISBN 0-387-95284-5.
- Pedro Domingos (September 2015), *The Master Algorithm*, Basic Books, ISBN 978-0-465-06570-7
- Ian H. Witten and Eibe Frank (2011). *Data Mining: Practical machine learning tools and techniques* Morgan Kaufmann, 664pp., ISBN 978-0-12-374856-0.
- Ethem Alpaydin (2004). *Introduction to Machine Learning*, MIT Press, ISBN 978-0-262-01243-0.
- David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms* (<http://www.inference.phy.cam.ac.uk/mackay/itila/book.html>) Cambridge: Cambridge University Press, 2003. ISBN 0-521-64298-1
- Richard O. Duda, Peter E. Hart, David G. Stork (2001) *Pattern classification* (2nd edition), Wiley, New York, ISBN 0-471-05669-3.
- Christopher Bishop (1995). *Neural Networks for Pattern Recognition*, Oxford University Press. ISBN 0-19-853864-2.
- Stuart Russell & Peter Norvig, (2009). *Artificial Intelligence – A Modern Approach* (<http://aima.cs.berkeley.edu/>). Pearson, ISBN 9789332543515.
- Ray Solomonoff, *An Inductive Inference Machine*, IRE Convention Record, Section on Information Theory, Part 2, pp., 56–62, 1957.
- Ray Solomonoff, *An Inductive Inference Machine* (<http://world.std.com/~rjs/indinf56.pdf>) A privately circulated report from the 1956 Dartmouth Summer Research Conference on AI.
- Kevin P. Murphy (2021). *Probabilistic Machine Learning: An Introduction* (<https://probml.github.io/pml-book/book1.html>), MIT Press.

External links

-  Quotations related to Machine learning at Wikiquote

- [International Machine Learning Society \(https://web.archive.org/web/20171230081341/http://machinelearning.org/\)](https://web.archive.org/web/20171230081341/http://machinelearning.org/)
 - [mloss \(https://mloss.org/\)](https://mloss.org/) is an academic database of open-source machine learning software.
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Machine_learning&oldid=1100788337"

This page was last edited on 27 July 2022, at 18:05 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License 3.0; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.