



# Glossary of artificial intelligence

---

This **glossary of artificial intelligence** is a list of definitions of terms and concepts relevant to the study of artificial intelligence (AI), its subdisciplines, and related fields. Related glossaries include Glossary of computer science, Glossary of robotics, and Glossary of machine vision.

## A

---

### A\* search

A graph traversal and pathfinding algorithm which is used in many fields of computer science due to its completeness, optimality, and optimal efficiency.

### abductive logic programming (ALP)

A high-level knowledge-representation framework that can be used to solve problems declaratively based on abductive reasoning. It extends normal logic programming by allowing some predicates to be incompletely defined, declared as abducible predicates.

### abductive reasoning

A form of logical inference which starts with an observation or set of observations then seeks to find the simplest and most likely explanation. This process, unlike deductive reasoning, yields a plausible conclusion but does not positively verify it.<sup>[1]</sup> abductive inference,<sup>[1]</sup> or retroduction<sup>[2]</sup>

### ablation

The removal of a component of an AI system. An *ablation study* aims to determine the contribution of a component to an AI system by removing the component, and then analyzing the resultant performance of the system.<sup>[3]</sup>

### abstract data type

A mathematical model for data types, where a data type is defined by its behavior (semantics) from the point of view of a *user* of the data, specifically in terms of possible values, possible operations on data of this type, and the behavior of these operations.

### abstraction

The process of removing physical, spatial, or temporal details<sup>[4]</sup> or attributes in the study of objects or systems in order to more closely attend to other details of interest<sup>[5]</sup>

### accelerating change

A perceived increase in the rate of technological change throughout history, which may suggest faster and more profound change in the future and may or may not be accompanied by equally profound social and cultural change.

### action language

A language for specifying state transition systems, and is commonly used to create formal models of the effects of actions on the world.<sup>[6]</sup> Action languages are commonly used in the artificial intelligence and robotics domains, where they describe how actions affect the states of systems over time, and may be used for automated planning.

### action model learning

An area of machine learning concerned with creation and modification of software agent's knowledge about effects and preconditions of the actions that can be executed within its environment. This knowledge is usually represented in logic-based action description language and used as the input for automated planners.

### **action selection**

A way of characterizing the most basic problem of intelligent systems: what to do next. In artificial intelligence and computational cognitive science, "the action selection problem" is typically associated with intelligent agents and animats—artificial systems that exhibit complex behaviour in an agent environment.

### **activation function**

In artificial neural networks, the activation function of a node defines the output of that node given an input or set of inputs.

### **adaptive algorithm**

An algorithm that changes its behavior at the time it is run, based on *a priori* defined reward mechanism or criterion.

### **adaptive neuro fuzzy inference system (ANFIS)**

A kind of artificial neural network that is based on Takagi–Sugeno fuzzy inference system. The technique was developed in the early 1990s.<sup>[7][8]</sup> Since it integrates both neural networks and fuzzy logic principles, it has potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF–THEN rules that have learning capability to approximate nonlinear functions.<sup>[9]</sup> Hence, ANFIS is considered to be a universal estimator.<sup>[10]</sup> For using the ANFIS in a more efficient and optimal way, one can use the best parameters obtained by genetic algorithm.<sup>[11][12]</sup>

### **admissible heuristic**

In computer science, specifically in algorithms related to pathfinding, a heuristic function is said to be admissible if it never overestimates the cost of reaching the goal, i.e. the cost it estimates to reach the goal is not higher than the lowest possible cost from the current point in the path.<sup>[13]</sup>

### **affective computing**

The study and development of systems and devices that can recognize, interpret, process, and simulate human affects. Affective computing is an interdisciplinary field spanning computer science, psychology, and cognitive science.<sup>[14][15]</sup>

### **agent architecture**

A blueprint for software agents and intelligent control systems, depicting the arrangement of components. The architectures implemented by intelligent agents are referred to as cognitive architectures.<sup>[16]</sup>

### **AI accelerator**

A class of microprocessor<sup>[17]</sup> or computer system<sup>[18]</sup> designed as hardware acceleration for artificial intelligence applications, especially artificial neural networks, machine vision, and machine learning.

### **AI-complete**

In the field of artificial intelligence, the most difficult problems are informally known as AI-complete or AI-hard, implying that the difficulty of these computational problems is equivalent to that of solving the central artificial intelligence problem—making computers as intelligent as people, or strong AI.<sup>[19]</sup> To call a problem AI-complete reflects an attitude that it would not be solved by a simple specific algorithm.

### **algorithm**

An unambiguous specification of how to solve a class of problems. Algorithms can perform calculation, data processing, and automated reasoning tasks.

### **algorithmic efficiency**

A property of an algorithm which relates to the number of computational resources used by the algorithm. An algorithm must be analyzed to determine its resource usage, and the efficiency of an algorithm can be measured based on usage of different resources. Algorithmic efficiency can be thought of as analogous to engineering productivity for a repeating or continuous process.

### **algorithmic probability**

In algorithmic information theory, algorithmic probability, also known as Solomonoff probability, is a mathematical method of assigning a prior probability to a given observation. It was invented by Ray Solomonoff in the 1960s.<sup>[20]</sup>

### **AlphaGo**

A computer program that plays the board game Go.<sup>[21]</sup> It was developed by Alphabet Inc.'s Google DeepMind in London. AlphaGo has several versions including AlphaGo Zero, AlphaGo Master, AlphaGo Lee, etc.<sup>[22]</sup> In October 2015, AlphaGo became the first computer Go program to beat a human professional Go player without handicaps on a full-sized 19×19 board.<sup>[23][24]</sup>

### **ambient intelligence (AmI)**

Electronic environments that are sensitive and responsive to the presence of people.

### **analysis of algorithms**

The determination of the computational complexity of algorithms, that is the amount of time, storage and/or other resources necessary to execute them. Usually, this involves determining a function that relates the length of an algorithm's input to the number of steps it takes (its time complexity) or the number of storage locations it uses (its space complexity).

### **analytics**

The discovery, interpretation, and communication of meaningful patterns in data.

### **answer set programming (ASP)**

A form of declarative programming oriented towards difficult (primarily NP-hard) search problems. It is based on the stable model (answer set) semantics of logic programming. In ASP, search problems are reduced to computing stable models, and answer set solvers—programs for generating stable models—are used to perform search.

### **ant colony optimization (ACO)**

A probabilistic technique for solving computational problems that can be reduced to finding good paths through graphs.

### **anytime algorithm**

An algorithm that can return a valid solution to a problem even if it is interrupted before it ends.

### **application programming interface (API)**

A set of subroutine definitions, communication protocols, and tools for building software. In general terms, it is a set of clearly defined methods of communication among various components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. An API may be for a web-based system, operating system, database system, computer hardware, or software library.

### **approximate string matching**

The technique of finding strings that match a pattern approximately (rather than exactly). The problem of approximate string matching is typically divided into two sub-problems:

finding approximate substring matches inside a given string and finding dictionary strings that match the pattern approximately.

### **approximation error**

The discrepancy between an exact value and some approximation to it.

### **argumentation framework**

A way to deal with contentious information and draw conclusions from it. In an abstract argumentation framework,<sup>[25]</sup> entry-level information is a set of abstract arguments that, for instance, represent data or a proposition. Conflicts between arguments are represented by a binary relation on the set of arguments. In concrete terms, you represent an argumentation framework with a directed graph such that the nodes are the arguments, and the arrows represent the attack relation. There exist some extensions of the Dung's framework, like the logic-based argumentation frameworks<sup>[26]</sup> or the value-based argumentation frameworks.<sup>[27]</sup>

### **artificial general intelligence (AGI)**

A type of AI that matches or surpasses human cognitive capabilities across a wide range of cognitive tasks.

### **artificial immune system (AIS)**

A class of computationally intelligent, rule-based machine learning systems inspired by the principles and processes of the vertebrate immune system. The algorithms are typically modeled after the immune system's characteristics of learning and memory for use in problem-solving.

### **artificial intelligence (AI)**

Any intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and other animals. In computer science, AI research is defined as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals.<sup>[28]</sup> Colloquially, the term "artificial intelligence" is applied when a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".<sup>[29]</sup>

### **Artificial Intelligence Markup Language**

An XML dialect for creating natural language software agents.

### **Association for the Advancement of Artificial Intelligence (AAAI)**

An international, nonprofit, scientific society devoted to promote research in, and responsible use of, artificial intelligence. AAAI also aims to increase public understanding of artificial intelligence (AI), improve the teaching and training of AI practitioners, and provide guidance for research planners and funders concerning the importance and potential of current AI developments and future directions.<sup>[30]</sup>

### **asymptotic computational complexity**

In computational complexity theory, asymptotic computational complexity is the usage of asymptotic analysis for the estimation of computational complexity of algorithms and computational problems, commonly associated with the usage of the big O notation.

### **attention mechanism**

Machine learning-based **attention** is a mechanism mimicking cognitive attention. It calculates "soft" weights for each word, more precisely for its embedding, in the context window. It can do it either in parallel (such as in transformers) or sequentially (such as in recursive neural networks). "Soft" weights can change during each runtime, in contrast to "hard" weights, which are (pre-)trained and fine-tuned and remain frozen afterwards. Multiple attention heads are used in transformer-based large language models.

### **attributinal calculus**

A logic and representation system defined by Ryszard S. Michalski. It combines elements of predicate logic, propositional calculus, and multi-valued logic. Attributional calculus provides a formal language for *natural induction*, an inductive learning process whose results are in forms natural to people.

### **augmented reality (AR)**

An interactive experience of a real-world environment where the objects that reside in the real-world are "augmented" by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory, and olfactory.<sup>[31]</sup>

### **autoencoder**

A type of artificial neural network used to learn efficient codings of unlabeled data (unsupervised learning). A common implementation is the variational autoencoder (VAE).

### **automata theory**

The study of abstract machines and automata, as well as the computational problems that can be solved using them. It is a theory in theoretical computer science and discrete mathematics (a subject of study in both mathematics and computer science).

### **automated machine learning (AutoML)**

A field of machine learning (ML) which aims to automatically configure an ML system to maximize its performance (e.g, classification accuracy).

### **automated planning and scheduling**

A branch of artificial intelligence that concerns the realization of strategies or action sequences, typically for execution by intelligent agents, autonomous robots and unmanned vehicles. Unlike classical control and classification problems, the solutions are complex and must be discovered and optimized in multidimensional space. Planning is also related to decision theory.<sup>[32]</sup>

### **automated reasoning**

An area of computer science and mathematical logic dedicated to understanding different aspects of reasoning. The study of automated reasoning helps produce computer programs that allow computers to reason completely, or nearly completely, automatically. Although automated reasoning is considered a sub-field of artificial intelligence, it also has connections with theoretical computer science, and even philosophy.

### **autonomic computing (AC)**

The self-managing characteristics of distributed computing resources, adapting to unpredictable changes while hiding intrinsic complexity to operators and users. Initiated by IBM in 2001, this initiative ultimately aimed to develop computer systems capable of self-management, to overcome the rapidly growing complexity of computing systems management, and to reduce the barrier that complexity poses to further growth.<sup>[33]</sup>

### **autonomous car**

A vehicle that is capable of sensing its environment and moving with little or no human input.<sup>[34][35][36]</sup>

### **autonomous robot**

A robot that performs behaviors or tasks with a high degree of autonomy. Autonomous robotics is usually considered to be a subfield of artificial intelligence, robotics, and information engineering.<sup>[37]</sup>

## **backpropagation**

A method used in artificial neural networks to calculate a gradient that is needed in the calculation of the weights to be used in the network.<sup>[38]</sup> Backpropagation is shorthand for "the backward propagation of errors", since an error is computed at the output and distributed backwards throughout the network's layers. It is commonly used to train deep neural networks,<sup>[39]</sup> a term referring to neural networks with more than one hidden layer.<sup>[40]</sup>

## **backpropagation through structure (BPTS)**

A gradient-based technique for training recurrent neural networks, proposed in a 1996 paper written by Christoph Goller and Andreas Küchler.<sup>[41]</sup>

## **backpropagation through time (BPTT)**

A gradient-based technique for training certain types of recurrent neural networks, such as Elman networks. The algorithm was independently derived by numerous researchers.<sup>[42][43][44]</sup>

## **backward chaining**

An inference method described colloquially as working backward from the goal. It is used in automated theorem provers, inference engines, proof assistants, and other artificial intelligence applications.<sup>[45]</sup>

## **bag-of-words model**

A simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity. The bag-of-words model has also been used for computer vision.<sup>[46]</sup> The bag-of-words model is commonly used in methods of document classification where the (frequency of) occurrence of each word is used as a feature for training a classifier.<sup>[47]</sup>

## **bag-of-words model in computer vision**

In computer vision, the bag-of-words model (BoW model) can be applied to image classification, by treating image features as words. In document classification, a bag of words is a sparse vector of occurrence counts of words; that is, a sparse histogram over the vocabulary. In computer vision, a *bag of visual words* is a vector of occurrence counts of a vocabulary of local image features.

## **batch normalization**

A technique for improving the performance and stability of artificial neural networks. It is a technique to provide any layer in a neural network with inputs that are zero mean/unit variance.<sup>[48]</sup> Batch normalization was introduced in a 2015 paper.<sup>[49][50]</sup> It is used to normalize the input layer by adjusting and scaling the activations.

## **Bayesian programming**

A formalism and a methodology for having a technique to specify probabilistic models and solve problems when less than the necessary information is available.

## **bees algorithm**

A population-based search algorithm which was developed by Pham, Ghanbarzadeh and et al. in 2005.<sup>[51]</sup> It mimics the food foraging behaviour of honey bee colonies. In its basic version the algorithm performs a kind of neighborhood search combined with global search, and can be used for both combinatorial optimization and continuous optimization. The only condition for the application of the bees algorithm is that some measure of distance between the solutions is defined. The effectiveness and specific abilities of the bees algorithm have been proven in a number of studies.<sup>[52][53][54][55]</sup>

## **behavior informatics (BI)**



The informatics of behaviors so as to obtain behavior intelligence and behavior insights.<sup>[56]</sup>

### **behavior tree (BT)**

A mathematical model of plan execution used in computer science, robotics, control systems and video games. They describe switchings between a finite set of tasks in a modular fashion. Their strength comes from their ability to create very complex tasks composed of simple tasks, without worrying how the simple tasks are implemented. BTs present some similarities to hierarchical state machines with the key difference that the main building block of a behavior is a task rather than a state. Its ease of human understanding make BTs less error-prone and very popular in the game developer community. BTs have shown to generalize several other control architectures.<sup>[57][58]</sup>

### **belief–desire–intention software model (BDI)**

A software model developed for programming intelligent agents. Superficially characterized by the implementation of an agent's *beliefs*, *desires* and *intentions*, it actually uses these concepts to solve a particular problem in agent programming. In essence, it provides a mechanism for separating the activity of selecting a plan (from a plan library or an external planner application) from the execution of currently active plans. Consequently, BDI agents are able to balance the time spent on deliberating about plans (choosing what to do) and executing those plans (doing it). A third activity, creating the plans in the first place (planning), is not within the scope of the model, and is left to the system designer and programmer.

### **bias–variance tradeoff**

In statistics and machine learning, the bias–variance tradeoff is the property of a set of predictive models whereby models with a lower bias in parameter estimation have a higher variance of the parameter estimates across samples, and vice versa.

### **big data**

A term used to refer to data sets that are too large or complex for traditional data-processing application software to adequately deal with. Data with many cases (rows) offer greater statistical power, while data with higher complexity (more attributes or columns) may lead to a higher false discovery rate.<sup>[59]</sup>

### **Big O notation**

A mathematical notation that describes the limiting behavior of a function when the argument tends towards a particular value or infinity. It is a member of a family of notations invented by Paul Bachmann,<sup>[60]</sup> Edmund Landau,<sup>[61]</sup> and others, collectively called Bachmann–Landau notation or asymptotic notation.

### **binary tree**

A tree data structure in which each node has at most two children, which are referred to as the *left child* and the *right child*. A recursive definition using just set theory notions is that a (non-empty) binary tree is a tuple  $(L, S, R)$ , where  $L$  and  $R$  are binary trees or the empty set and  $S$  is a singleton set.<sup>[62]</sup> Some authors allow the binary tree to be the empty set as well.<sup>[63]</sup>

### **blackboard system**

An artificial intelligence approach based on the blackboard architectural model,<sup>[64][65][66][67]</sup> where a common knowledge base, the "blackboard", is iteratively updated by a diverse group of specialist knowledge sources, starting with a problem specification and ending with a solution. Each knowledge source updates the blackboard with a partial solution when its internal constraints match the blackboard state. In this way, the specialists work together to solve the problem.

### **Boltzmann machine**

A type of stochastic recurrent neural network and Markov random field.<sup>[68]</sup> Boltzmann machines can be seen as the stochastic, generative counterpart of Hopfield networks.

### **Boolean satisfiability problem**

The problem of determining if there exists an interpretation that satisfies a given Boolean formula. In other words, it asks whether the variables of a given Boolean formula can be consistently replaced by the values TRUE or FALSE in such a way that the formula evaluates to TRUE. If this is the case, the formula is called *satisfiable*. On the other hand, if no such assignment exists, the function expressed by the formula is FALSE for all possible variable assignments and the formula is *unsatisfiable*. For example, the formula "a AND NOT b" is satisfiable because one can find the values  $a = \text{TRUE}$  and  $b = \text{FALSE}$ , which make  $(a \text{ AND NOT } b) = \text{TRUE}$ . In contrast, "a AND NOT a" is unsatisfiable.

### **boosting**

A machine learning ensemble metaheuristic for primarily reducing bias (as opposed to variance), by training models sequentially, each one correcting the errors of its predecessor.

### **bootstrap aggregating**

A machine learning ensemble metaheuristic for primarily reducing variance (as opposed to bias), by training multiple models independently and averaging their predictions.

### **brain technology**

A technology that employs the latest findings in neuroscience. The term was first introduced by the Artificial Intelligence Laboratory in Zurich, Switzerland, in the context of the ROBOY project.<sup>[69]</sup> Brain Technology can be employed in robots,<sup>[70]</sup> know-how management systems<sup>[71]</sup> and any other application with self-learning capabilities. In particular, Brain Technology applications allow the visualization of the underlying learning architecture often coined as "know-how maps".

### **branching factor**

In computing, tree data structures, and game theory, the number of children at each node, the outdegree. If this value is not uniform, an *average branching factor* can be calculated.

### **brute-force search**

A very general problem-solving technique and algorithmic paradigm that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement.

## **C**

---

### **capsule neural network (CapsNet)**

A machine learning system that is a type of artificial neural network (ANN) that can be used to better model hierarchical relationships. The approach is an attempt to more closely mimic biological neural organization.<sup>[72]</sup>

### **case-based reasoning (CBR)**

Broadly construed, the process of solving new problems based on the solutions of similar past problems.

### **chatbot**



A computer program or an artificial intelligence which conducts a conversation via auditory or textual methods.<sup>[73]</sup>

### **cloud robotics**

A field of robotics that attempts to invoke cloud technologies such as cloud computing, cloud storage, and other Internet technologies centred on the benefits of converged infrastructure and shared services for robotics. When connected to the cloud, robots can benefit from the powerful computation, storage, and communication resources of modern data center in the cloud, which can process and share information from various robots or agent (other machines, smart objects, humans, etc.). Humans can also delegate tasks to robots remotely through networks. Cloud computing technologies enable robot systems to be endowed with powerful capability whilst reducing costs through cloud technologies. Thus, it is possible to build lightweight, low cost, smarter robots have intelligent "brain" in the cloud. The "brain" consists of data center, knowledge base, task planners, deep learning, information processing, environment models, communication support, etc.<sup>[74][75][76][77]</sup>

### **cluster analysis**

The task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

### **Cobweb**

An incremental system for hierarchical conceptual clustering. COBWEB was invented by Professor Douglas H. Fisher, currently at Vanderbilt University.<sup>[78][79]</sup> COBWEB incrementally organizes observations into a classification tree. Each node in a classification tree represents a class (concept) and is labeled by a probabilistic concept that summarizes the attribute-value distributions of objects classified under the node. This classification tree can be used to predict missing attributes or the class of a new object.<sup>[80]</sup>

### **cognitive architecture**

The Institute of Creative Technologies defines cognitive architecture as: "hypothesis about the fixed structures that provide a mind, whether in natural or artificial systems, and how they work together – in conjunction with knowledge and skills embodied within the architecture – to yield intelligent behavior in a diversity of complex environments."<sup>[81]</sup>

### **cognitive computing**

In general, the term cognitive computing has been used to refer to new hardware and/or software that mimics the functioning of the human brain<sup>[82][83][84][85][86][87]</sup> and helps to improve human decision-making.<sup>[88]</sup> In this sense, CC is a new type of computing with the goal of more accurate models of how the human brain/mind senses, reasons, and responds to stimulus.

### **cognitive science**

The interdisciplinary scientific study of the mind and its processes.<sup>[89]</sup>

### **combinatorial optimization**

In Operations Research, applied mathematics and theoretical computer science, combinatorial optimization is a topic that consists of finding an optimal object from a finite set of objects.<sup>[90]</sup>

### **committee machine**

A type of artificial neural network using a divide and conquer strategy in which the responses of multiple neural networks (experts) are combined into a single response.<sup>[91]</sup>

The combined response of the committee machine is supposed to be superior to those of its constituent experts. Compare ensembles of classifiers.

### **commonsense knowledge**

In artificial intelligence research, commonsense knowledge consists of facts about the everyday world, such as "Lemons are sour", that all humans are expected to know. The first AI program to address common sense knowledge was Advice Taker in 1959 by John McCarthy.<sup>[92]</sup>

### **commonsense reasoning**

A branch of artificial intelligence concerned with simulating the human ability to make presumptions about the type and essence of ordinary situations they encounter every day.<sup>[93]</sup>

### **computational chemistry**

A branch of chemistry that uses computer simulation to assist in solving chemical problems.

### **computational complexity theory**

Focuses on classifying computational problems according to their inherent difficulty, and relating these classes to each other. A computational problem is a task solved by a computer. A computation problem is solvable by mechanical application of mathematical steps, such as an algorithm.

### **computational creativity**

A multidisciplinary endeavour that includes the fields of artificial intelligence, cognitive psychology, philosophy, and the arts.

### **computational cybernetics**

The integration of cybernetics and computational intelligence techniques.

### **computational humor**

A branch of computational linguistics and artificial intelligence which uses computers in humor research.<sup>[94]</sup>

### **computational intelligence (CI)**

Usually refers to the ability of a computer to learn a specific task from data or experimental observation.

### **computational learning theory**

In computer science, computational learning theory (or just learning theory) is a subfield of artificial intelligence devoted to studying the design and analysis of machine learning algorithms.<sup>[95]</sup>

### **computational linguistics**

An interdisciplinary field concerned with the statistical or rule-based modeling of natural language from a computational perspective, as well as the study of appropriate computational approaches to linguistic questions.

### **computational mathematics**

The mathematical research in areas of science where computing plays an essential role.

### **computational neuroscience**

A branch of neuroscience which employs mathematical models, theoretical analysis and abstractions of the brain to understand the principles that govern the development, structure, physiology, and cognitive abilities of the nervous system.<sup>[96][97][98][99]</sup>

### **computational number theory**

The study of algorithms for performing number theoretic computations.

## **computational problem**

In theoretical computer science, a computational problem is a mathematical object representing a collection of questions that computers might be able to solve.

## **computational statistics**

The interface between statistics and computer science.

## **computer-automated design (CAutoD)**

Design automation usually refers to electronic design automation, or Design Automation which is a Product Configurator. Extending Computer-Aided Design (CAD), automated design and computer-automated design<sup>[100][101][102]</sup> are concerned with a broader range of applications, such as automotive engineering, civil engineering,<sup>[103][104][105][106]</sup> composite material design, control engineering,<sup>[107]</sup> dynamic system identification and optimization,<sup>[108]</sup> financial systems, industrial equipment, mechatronic systems, steel construction,<sup>[109]</sup> structural optimisation,<sup>[110]</sup> and the invention of novel systems. More recently, traditional CAD simulation is seen to be transformed to CAutoD by biologically inspired machine learning,<sup>[111]</sup> including heuristic search techniques such as evolutionary computation,<sup>[112][113]</sup> and swarm intelligence algorithms.<sup>[114]</sup>

## **computer audition (CA)**

See machine listening.

## **computer science**

The theory, experimentation, and engineering that form the basis for the design and use of computers. It involves the study of algorithms that process, store, and communicate digital information. A computer scientist specializes in the theory of computation and the design of computational systems.<sup>[115]</sup>

## **computer vision**

An interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do.<sup>[116][117][118]</sup>

## **concept drift**

In predictive analytics and machine learning, the concept drift means that the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways. This causes problems because the predictions become less accurate as time passes.

## **connectionism**

An approach in the fields of cognitive science, that hopes to explain mental phenomena using artificial neural networks.<sup>[119]</sup>

## **consistent heuristic**

In the study of path-finding problems in artificial intelligence, a heuristic function is said to be consistent, or monotone, if its estimate is always less than or equal to the estimated distance from any neighboring vertex to the goal, plus the cost of reaching that neighbor.

## **constrained conditional model (CCM)**

A machine learning and inference framework that augments the learning of conditional (probabilistic or discriminative) models with declarative constraints.

## **constraint logic programming**

A form of constraint programming, in which logic programming is extended to include concepts from constraint satisfaction. A constraint logic program is a logic program that contains constraints in the body of clauses. An example of a clause including a constraint is  $A(X, Y) :- X+Y>0, B(X), C(Y)$ . In this clause,  $X+Y>0$  is a constraint;  $A(X, Y)$ ,  $B(X)$ , and  $C(Y)$  are literals as in regular logic programming. This clause states one

condition under which the statement  $A(X, Y)$  holds:  $X+Y$  is greater than zero and both  $B(X)$  and  $C(Y)$  are true.

### **constraint programming**

A programming paradigm wherein relations between variables are stated in the form of constraints. Constraints differ from the common primitives of imperative programming languages in that they do not specify a step or sequence of steps to execute, but rather the properties of a solution to be found.

### **constructed language**

A language whose phonology, grammar, and vocabulary are consciously devised, instead of having developed naturally. Constructed languages may also be referred to as artificial, planned, or invented languages.<sup>[120]</sup>

### **control theory**

In control systems engineering is a subfield of mathematics that deals with the control of continuously operating dynamical systems in engineered processes and machines. The objective is to develop a control model for controlling such systems using a control action in an optimum manner without *delay or overshoot* and ensuring control stability.

### **convolutional neural network**

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural network most commonly applied to image analysis. CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing.<sup>[121]</sup> They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.<sup>[122][123]</sup>

### **crossover**

In genetic algorithms and evolutionary computation, a genetic operator used to combine the genetic information of two parents to generate new offspring. It is one way to stochastically generate new solutions from an existing population, and analogous to the crossover that happens during sexual reproduction in biological organisms. Solutions can also be generated by cloning an existing solution, which is analogous to asexual reproduction. Newly generated solutions are typically mutated before being added to the population.

## **D**

---

### **Darkforest**

A computer go program developed by Facebook, based on deep learning techniques using a convolutional neural network. Its updated version Darkfores2 combines the techniques of its predecessor with Monte Carlo tree search.<sup>[124][125]</sup> The MCTS effectively takes tree search methods commonly seen in computer chess programs and randomizes them.<sup>[126]</sup> With the update, the system is known as Darkfmcts3.<sup>[127]</sup>

### **Dartmouth workshop**

The Dartmouth Summer Research Project on Artificial Intelligence was the name of a 1956 summer workshop now considered by many<sup>[128][129]</sup> (though not all<sup>[130]</sup>) to be the seminal event for artificial intelligence as a field.

### **data augmentation**

Data augmentation in data analysis are techniques used to increase the amount of data. It helps reduce overfitting when training a learning algorithm.

### **data fusion**

The process of integrating multiple data sources to produce more consistent, accurate, and useful information than that provided by any individual data source.<sup>[131]</sup>

### **data integration**

The process of combining data residing in different sources and providing users with a unified view of them.<sup>[132]</sup> This process becomes significant in a variety of situations, which include both commercial (such as when two similar companies need to merge their databases) and scientific (combining research results from different bioinformatics repositories, for example) domains. Data integration appears with increasing frequency as the volume (that is, big data) and the need to share existing data explodes.<sup>[133]</sup> It has become the focus of extensive theoretical work, and numerous open problems remain unsolved.

### **data mining**

The process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems.

### **data science**

An interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured,<sup>[134][135]</sup> similar to data mining. Data science is a "concept to unify statistics, data analysis, machine learning, and their related methods" in order to "understand and analyze actual phenomena" with data.<sup>[136]</sup> It employs techniques and theories drawn from many fields within the context of mathematics, statistics, information science, and computer science.

### **data set**

A collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. The data set may comprise data for one or more members, corresponding to the number of rows.

### **data warehouse (DW or DWH)**

A system used for reporting and data analysis.<sup>[137]</sup> DWs are central repositories of integrated data from one or more disparate sources. They store current and historical data in one single place<sup>[138]</sup>

### **Datalog**

A declarative logic programming language that syntactically is a subset of Prolog. It is often used as a query language for deductive databases. In recent years, Datalog has found new application in data integration, information extraction, networking, program analysis, security, and cloud computing.<sup>[139]</sup>

### **decision boundary**

In the case of backpropagation-based artificial neural networks or perceptrons, the type of decision boundary that the network can learn is determined by the number of hidden layers in the network. If it has no hidden layers, then it can only learn linear problems. If it has one hidden layer, then it can learn any continuous function on compact subsets of  $\mathbb{R}^n$  as shown by the Universal approximation theorem, thus it can have an arbitrary decision boundary.

### **decision support system (DSS)**

An information system that supports business or organizational decision-making activities. DSSs serve the management, operations and planning levels of an organization

(usually mid and higher management) and help people make decisions about problems that may be rapidly changing and not easily specified in advance—i.e. unstructured and semi-structured decision problems. Decision support systems can be either fully computerized or human-powered, or a combination of both.

### **decision theory**

The study of the reasoning underlying an agent's choices.<sup>[140]</sup> Decision theory can be broken into two branches: normative decision theory, which gives advice on how to make the best decisions given a set of uncertain beliefs and a set of values, and descriptive decision theory which analyzes how existing, possibly irrational agents actually make decisions.

### **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.

### **declarative programming**

A programming paradigm—a style of building the structure and elements of computer programs—that expresses the logic of a computation without describing its control flow.<sup>[141]</sup>

### **deductive classifier**

A type of artificial intelligence inference engine. It takes as input a set of declarations in a frame language about a domain such as medical research or molecular biology. For example, the names of classes, sub-classes, properties, and restrictions on allowable values.

### **Deep Blue**

was a chess-playing computer developed by IBM. It is known for being the first computer chess-playing system to win both a chess game and a chess match against a reigning world champion under regular time controls.

### **deep learning**

A subset of machine learning that focuses on utilizing neural networks to perform tasks such as classification, regression, and representation learning. The field takes inspiration from biological neuroscience and is centered around stacking artificial neurons into layers and "training" them to process data. The adjective "deep" refers to the use of multiple layers (ranging from three to several hundred or thousands) in the network. Methods used can be either supervised, semi-supervised, or unsupervised.

### **DeepMind Technologies**

A British artificial intelligence company founded in September 2010, currently owned by Alphabet Inc. The company is based in London, with research centres in Canada,<sup>[142]</sup> France,<sup>[143]</sup> and the United States. Acquired by Google in 2014, the company has created a neural network that learns how to play video games in a fashion similar to that of humans,<sup>[144]</sup> as well as a neural Turing machine,<sup>[145]</sup> or a neural network that may be able to access an external memory like a conventional Turing machine, resulting in a computer that mimics the short-term memory of the human brain.<sup>[146][147]</sup> The company made headlines in 2016 after its AlphaGo program beat human professional Go player Lee Sedol, the world champion, in a five-game match, which was the subject of a documentary film.<sup>[148]</sup> A more general program, AlphaZero, beat the most powerful programs playing Go, chess, and shogi (Japanese chess) after a few days of play against itself using reinforcement learning.<sup>[149]</sup>

### **default logic**



A non-monotonic logic proposed by Raymond Reiter to formalize reasoning with default assumptions.

### **Density-based spatial clustering of applications with noise (DBSCAN)**

A clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu in 1996.<sup>[150]</sup>

### **description logic (DL)**

A family of formal knowledge representation languages. Many DLs are more expressive than propositional logic but less expressive than first-order logic. In contrast to the latter, the core reasoning problems for DLs are (usually) decidable, and efficient decision procedures have been designed and implemented for these problems. There are general, spatial, temporal, spatiotemporal, and fuzzy descriptions logics, and each description logic features a different balance between DL expressivity and reasoning complexity by supporting different sets of mathematical constructors.<sup>[151]</sup>

### **developmental robotics (DevRob)**

A scientific field which aims at studying the developmental mechanisms, architectures, and constraints that allow lifelong and open-ended learning of new skills and new knowledge in embodied machines.

### **diagnosis**

Concerned with the development of algorithms and techniques that are able to determine whether the behaviour of a system is correct. If the system is not functioning correctly, the algorithm should be able to determine, as accurately as possible, which part of the system is failing, and which kind of fault it is facing. The computation is based on *observations*, which provide information on the current behaviour.

### **dialogue system**

A computer system intended to converse with a human with a coherent structure. Dialogue systems have employed text, speech, graphics, haptics, gestures, and other modes for communication on both the input and output channel.

### **diffusion model**

In machine learning, **diffusion models**, also known as **diffusion probabilistic models** or **score-based generative models**, are a class of latent variable models. They are Markov chains trained using variational inference.<sup>[152]</sup> The goal of diffusion models is to learn the latent structure of a dataset by modeling the way in which data points diffuse through the latent space. In computer vision, this means that a neural network is trained to denoise images blurred with Gaussian noise by learning to reverse the diffusion process.<sup>[153][154]</sup> It mainly consists of three major components: the forward process, the reverse process, and the sampling procedure.<sup>[155]</sup> Three examples of generic diffusion modeling frameworks used in computer vision are denoising diffusion probabilistic models, noise conditioned score networks, and stochastic differential equations.<sup>[156]</sup>

### **Dijkstra's algorithm**

An algorithm for finding the shortest paths between nodes in a weighted graph, which may represent, for example, road networks.

### **dimensionality reduction**

The process of reducing the number of random variables under consideration<sup>[157]</sup> by obtaining a set of principal variables. It can be divided into feature selection and feature extraction.<sup>[158]</sup>

### **discrete system**

Any system with a countable number of states. Discrete systems may be contrasted with continuous systems, which may also be called analog systems. A final discrete system is

often modeled with a directed graph and is analyzed for correctness and complexity according to computational theory. Because discrete systems have a countable number of states, they may be described in precise mathematical models. A computer is a finite-state machine that may be viewed as a discrete system. Because computers are often used to model not only other discrete systems but continuous systems as well, methods have been developed to represent real-world continuous systems as discrete systems. One such method involves sampling a continuous signal at discrete time intervals.

### **distributed artificial intelligence (DAI)**

A subfield of artificial intelligence research dedicated to the development of distributed solutions for problems. DAI is closely related to and a predecessor of the field of multi-agent systems.<sup>[159]</sup>

### **double descent**

A phenomenon in statistics and machine learning where a model with a small number of parameters and a model with an extremely large number of parameters have a small test error, but a model whose number of parameters is about the same as the number of data points used to train the model will have a large error.<sup>[160]</sup> This phenomenon has been considered surprising, as it contradicts assumptions about overfitting in classical machine learning.<sup>[161]</sup>

### **dropout**

A regularization technique for reducing overfitting in artificial neural networks by preventing complex co-adaptations on training data.

### **dynamic epistemic logic (DEL)**

A logical framework dealing with knowledge and information change. Typically, DEL focuses on situations involving multiple agents and studies how their knowledge changes when events occur.

## **E**

---

### **eager learning**

A learning method in which the system tries to construct a general, input-independent target function during training of the system, as opposed to lazy learning, where generalization beyond the training data is delayed until a query is made to the system.<sup>[162]</sup>

### **early stopping**

A regularization technique often used when training a machine learning model with an iterative method such as gradient descent.

### **Ebert test**

A test which gauges whether a computer-based synthesized voice<sup>[163][164]</sup> can tell a joke with sufficient skill to cause people to laugh.<sup>[165]</sup> It was proposed by film critic Roger Ebert at the 2011 TED conference as a challenge to software developers to have a computerized voice master the inflections, delivery, timing, and intonations of a speaking human.<sup>[163]</sup> The test is similar to the Turing test proposed by Alan Turing in 1950 as a way to gauge a computer's ability to exhibit intelligent behavior by generating performance indistinguishable from a human being.<sup>[166]</sup>

### **echo state network (ESN)**

A recurrent neural network with a sparsely connected hidden layer (with typically 1% connectivity). The connectivity and weights of hidden neurons are fixed and randomly assigned. The weights of output neurons can be learned so that the network can

(re)produce specific temporal patterns. The main interest of this network is that although its behaviour is non-linear, the only weights that are modified during training are for the synapses that connect the hidden neurons to output neurons. Thus, the error function is quadratic with respect to the parameter vector and can be differentiated easily to a linear system.<sup>[167][168]</sup>

### **embodied agent**

An intelligent agent that interacts with the environment through a physical body within that environment. Agents that are represented graphically with a body, for example a human or a cartoon animal, are also called embodied agents, although they have only virtual, not physical, embodiment.<sup>[169]</sup>

### **embodied cognitive science**

An interdisciplinary field of research, the aim of which is to explain the mechanisms underlying intelligent behavior. It comprises three main methodologies: 1) the modeling of psychological and biological systems in a holistic manner that considers the mind and body as a single entity, 2) the formation of a common set of general principles of intelligent behavior, and 3) the experimental use of robotic agents in controlled environments.

### **error-driven learning**

A sub-area of machine learning concerned with how an agent ought to take actions in an environment so as to minimize some error feedback. It is a type of reinforcement learning.

### **ensemble learning**

The use of multiple machine learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.<sup>[170][171][172]</sup>

### **epoch**

In machine learning, particularly in the creation of artificial neural networks, an epoch is training the model for one cycle through the full training dataset. Small models are typically trained for as many epochs as it takes to reach the best performance on the validation dataset. The largest models may train for only one epoch.

### **ethics of artificial intelligence**

The part of the ethics of technology specific to artificial intelligence.

### **evolutionary algorithm (EA)**

A subset of evolutionary computation,<sup>[173]</sup> a generic population-based metaheuristic optimization algorithm. An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solutions (see also loss function). Evolution of the population then takes place after the repeated application of the above operators.

### **evolutionary computation**

A family of algorithms for global optimization inspired by biological evolution, and the subfield of artificial intelligence and soft computing studying these algorithms. In technical terms, they are a family of population-based trial and error problem solvers with a metaheuristic or stochastic optimization character.

### **evolving classification function (ECF)**

Evolving classification functions are used for classifying and clustering in the field of machine learning and artificial intelligence, typically employed for data stream mining tasks in dynamic and changing environments.

### **existential risk**

The hypothesis that substantial progress in artificial general intelligence (AGI) could someday result in human extinction or some other unrecoverable global catastrophe.<sup>[174][175][176]</sup>

### **expert system**

A computer system that emulates the decision-making ability of a human expert.<sup>[177]</sup> Expert systems are designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as if-then rules rather than through conventional procedural code.<sup>[178]</sup>

## F

---

### **fast-and-frugal trees**

A type of classification tree. Fast-and-frugal trees can be used as decision-making tools which operate as lexicographic classifiers, and, if required, associate an action (decision) to each class or category.<sup>[179]</sup>

### **feature**

An individual measurable property or characteristic of a phenomenon.<sup>[180]</sup> In computer vision and image processing, a feature is a piece of information about the content of an image; typically about whether a certain region of the image has certain properties. Features may be specific structures in an image (such as points, edges, or objects), or the result of a general neighborhood operation or feature detection applied to the image.

### **feature extraction**

In machine learning, pattern recognition, and image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations.

### **feature learning**

In machine learning, feature learning or representation learning<sup>[181]</sup> is a set of techniques that allows a system to automatically discover the representations needed for feature detection or classification from raw data. This replaces manual feature engineering and allows a machine to both learn the features and use them to perform a specific task.

### **feature selection**

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction.

### **federated learning**

A machine learning technique that allows for training models on multiple devices with decentralized data, thus helping preserve the privacy of individual users and their data.

### **first-order logic**

A collection of formal systems used in mathematics, philosophy, linguistics, and computer science. First-order logic uses quantified variables over non-logical objects and allows the use of sentences that contain variables, so that rather than propositions such as *Socrates is a man* one can have expressions in the form "there exists X such that X is Socrates and X is a man" and *there exists* is a quantifier while X is a variable.<sup>[182]</sup> This distinguishes it from propositional logic, which does not use quantifiers or relations.<sup>[183]</sup>

### **fluent**

A condition that can change over time. In logical approaches to reasoning about actions, fluents can be represented in first-order logic by predicates having an argument that depends on time.

### **formal language**

A set of words whose letters are taken from an alphabet and are well-formed according to a specific set of rules.

### **forward chaining**

One of the two main methods of reasoning when using an inference engine and can be described logically as repeated application of *modus ponens*. Forward chaining is a popular implementation strategy for expert systems, businesses and production rule systems. The opposite of forward chaining is backward chaining. Forward chaining starts with the available data and uses inference rules to extract more data (from an end user, for example) until a goal is reached. An inference engine using forward chaining searches the inference rules until it finds one where the antecedent (If clause) is known to be true. When such a rule is found, the engine can conclude, or infer, the consequent (Then clause), resulting in the addition of new information to its data.<sup>[184]</sup>

### **frame**

An artificial intelligence data structure used to divide knowledge into substructures by representing "stereotyped situations". Frames are the primary data structure used in artificial intelligence frame language.

### **frame language**

A technology used for knowledge representation in artificial intelligence. Frames are stored as ontologies of sets and subsets of the frame concepts. They are similar to class hierarchies in object-oriented languages although their fundamental design goals are different. Frames are focused on explicit and intuitive representation of knowledge whereas objects focus on encapsulation and information hiding. Frames originated in AI research and objects primarily in software engineering. However, in practice the techniques and capabilities of frame and object-oriented languages overlap significantly.

### **frame problem**

The problem of finding adequate collections of axioms for a viable description of a robot environment.<sup>[185]</sup>

### **friendly artificial intelligence**

A hypothetical artificial general intelligence (AGI) that would have a positive effect on humanity. It is a part of the ethics of artificial intelligence and is closely related to machine ethics. While machine ethics is concerned with how an artificially intelligent agent should behave, friendly artificial intelligence research is focused on how to practically bring about this behaviour and ensuring it is adequately constrained.

### **futures studies**

The study of postulating possible, probable, and preferable futures and the worldviews and myths that underlie them.<sup>[186]</sup>

### **fuzzy control system**

A control system based on fuzzy logic—a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 1 or 0 (true or false, respectively).<sup>[187][188]</sup>

### **fuzzy logic**

A simple form for the many-valued logic, in which the truth values of variables may have any degree of "*Truthfulness*" that can be represented by any real number in the range between 0 (as in Completely False) and 1 (as in Completely True) inclusive.

Consequently, It is employed to handle the concept of partial truth, where the truth value may range between completely true and completely false. In contrast to Boolean logic, where the truth values of variables may have the integer values 0 or 1 only.

### **fuzzy rule**

A rule used within fuzzy logic systems to infer an output based on input variables.

### **fuzzy set**

In classical set theory, the membership of elements in a set is assessed in binary terms according to a bivalent condition — an element either belongs or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval [0, 1]. Fuzzy sets generalize classical sets, since the indicator functions (aka *characteristic functions*) of classical sets are special cases of the membership functions of fuzzy sets, if the latter only take values 0 or 1.<sup>[189]</sup> In fuzzy set theory, classical bivalent sets are usually called crisp sets. The fuzzy set theory can be used in a wide range of domains in which information is incomplete or imprecise, such as bioinformatics.<sup>[190]</sup>

## **G**

---

### **game theory**

The study of mathematical models of strategic interaction between rational decision-makers.<sup>[191]</sup>

### **general game playing (GGP)**

General game playing is the design of artificial intelligence programs to be able to run and play more than one game successfully.<sup>[192][193][194]</sup>

### **generalization**

The concept that humans, other animals, and artificial neural networks use past learning in present situations of learning if the conditions in the situations are regarded as similar.<sup>[195]</sup>

### **generalization error**

For supervised learning applications in machine learning and statistical learning theory, **generalization error**<sup>[196]</sup> (also known as the **out-of-sample error**<sup>[197]</sup> or the **risk**) is a measure of how accurately a learning algorithm is able to predict outcomes for previously unseen data.

### **generative adversarial network (GAN)**

A class of machine learning systems. Two neural networks contest with each other in a zero-sum game framework.

### **generative artificial intelligence**

Generative artificial intelligence is artificial intelligence capable of generating text, images, or other media in response to prompts.<sup>[198][199]</sup> Generative AI models learn the patterns and structure of their input training data and then generate new data that has similar characteristics, typically using transformer-based deep neural networks.<sup>[200][201]</sup>

### **generative pretrained transformer (GPT)**

A large language model based on the transformer architecture that generates text. It is first pretrained to predict the next token in texts (a token is typically a word, subword, or punctuation). After their pretraining, GPT models can generate human-like text by repeatedly predicting the token that they would expect to follow. GPT models are usually also fine-tuned, for example with reinforcement learning from human feedback to reduce hallucination or harmful behaviour, or to format the output in a conversational format.<sup>[202]</sup>

### **genetic algorithm (GA)**

A metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). Genetic algorithms are commonly used to generate



high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection.<sup>[203]</sup>

### **genetic operator**

An operator used in genetic algorithms to guide the algorithm towards a solution to a given problem. There are three main types of operators (mutation, crossover and selection), which must work in conjunction with one another in order for the algorithm to be successful.

### **glowworm swarm optimization**

A swarm intelligence optimization algorithm based on the behaviour of glowworms (also known as fireflies or lightning bugs).

### **gradient boosting**

A machine learning technique based on boosting in a functional space, where the target is *pseudo-residuals* instead of residuals as in traditional boosting.

### **graph (abstract data type)**

In computer science, a graph is an abstract data type that is meant to implement the undirected graph and directed graph concepts from mathematics; specifically, the field of graph theory.

### **graph (discrete mathematics)**

In mathematics, and more specifically in graph theory, a graph is a structure amounting to a set of objects in which some pairs of the objects are in some sense "related". The objects correspond to mathematical abstractions called vertices (also called *nodes* or *points*) and each of the related pairs of vertices is called an edge (also called an *arc* or *line*).<sup>[204]</sup>

### **graph database (GDB)**

A database that uses graph structures for semantic queries with nodes, edges, and properties to represent and store data. A key concept of the system is the *graph* (or *edge* or *relationship*), which directly relates data items in the store a collection of nodes of data and edges representing the relationships between the nodes. The relationships allow data in the store to be linked together directly, and in many cases retrieved with one operation. Graph databases hold the relationships between data as a priority. Querying relationships within a graph database is fast because they are perpetually stored within the database itself. Relationships can be intuitively visualized using graph databases, making it useful for heavily inter-connected data.<sup>[205][206]</sup>

### **graph theory**

The study of graphs, which are mathematical structures used to model pairwise relations between objects.

### **graph traversal**

The process of visiting (checking and/or updating) each vertex in a graph. Such traversals are classified by the order in which the vertices are visited. Tree traversal is a special case of graph traversal.

## **H**

---

### **hallucination**

A response generated by AI that contains false or misleading information presented as fact.

### **heuristic**

A technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution. This is achieved by trading optimality, completeness, accuracy, or precision for speed. In a way, it can be considered a shortcut. A heuristic function, also called simply a heuristic, is a function that ranks alternatives in search algorithms at each branching step based on available information to decide which branch to follow. For example, it may approximate the exact solution.<sup>[207]</sup>

### **hidden layer**

A layer of neurons in an [artificial neural network](#) that is neither an input layer nor an output layer.

### **hyper-heuristic**

A [heuristic](#) search method that seeks to automate the process of selecting, combining, generating, or adapting several simpler heuristics (or components of such heuristics) to efficiently solve computational search problems, often by the incorporation of [machine learning](#) techniques. One of the motivations for studying hyper-heuristics is to build systems which can handle classes of problems rather than solving just one problem.<sup>[208][209][210]</sup>

### **hyperparameter**

A [parameter](#) that can be set in order to define any configurable part of a [machine learning](#) model's learning process.

### **hyperparameter optimization**

The process of choosing a set of optimal [hyperparameters](#) for a learning [algorithm](#).

### **hyperplane**

A decision boundary in [machine learning](#) classifiers that partitions the input space into two or more sections, with each section corresponding to a unique class label.

## **I**

---

### **IEEE Computational Intelligence Society**

A professional society of the Institute of Electrical and Electronics Engineers (IEEE) focussing on "the theory, design, application, and development of biologically and linguistically motivated computational paradigms emphasizing [neural networks](#), connectionist systems, genetic algorithms, [evolutionary programming](#), fuzzy systems, and hybrid intelligent systems in which these paradigms are contained".<sup>[211]</sup>

### **incremental learning**

A method of [machine learning](#), in which input data is continuously used to extend the existing model's knowledge i.e. to further train the model. It represents a dynamic technique of [supervised](#) and [unsupervised learning](#) that can be applied when training data becomes available gradually over time or its size is out of system memory limits. Algorithms that can facilitate incremental learning are known as incremental machine learning algorithms.

### **inference engine**

A component of the system that applies logical rules to the knowledge base to deduce new information.

### **information integration (II)**

The merging of information from heterogeneous sources with differing conceptual, contextual and typographical representations. It is used in [data mining](#) and consolidation of data from unstructured or semi-structured resources. Typically, *information integration* refers to textual representations of knowledge but is sometimes applied to rich-media content. Information fusion, which is a related term, involves the combination of

information into a new set of information towards reducing redundancy and uncertainty.<sup>[131]</sup>

### **Information Processing Language (IPL)**

A programming language that includes features intended to help with programs that perform simple problem solving actions such as lists, dynamic memory allocation, data types, recursion, functions as arguments, generators, and cooperative multitasking. IPL invented the concept of list processing, albeit in an assembly-language style.

### **intelligence amplification (IA)**

The effective use of information technology in augmenting human intelligence.

### **intelligence explosion**

A possible outcome of humanity building artificial general intelligence (AGI). AGI would be capable of recursive self-improvement leading to rapid emergence of ASI (artificial superintelligence), the limits of which are unknown, at the time of the technological singularity.

### **intelligent agent (IA)**

An autonomous entity which acts, directing its activity towards achieving goals (i.e. it is an agent), upon an environment using observation through sensors and consequent actuators (i.e. it is intelligent). Intelligent agents may also learn or use knowledge to achieve their goals. They may be very simple or very complex.

### **intelligent control**

A class of control techniques that use various artificial intelligence computing approaches like neural networks, Bayesian probability, fuzzy logic, machine learning, reinforcement learning, evolutionary computation and genetic algorithms.<sup>[212]</sup>

### **intelligent personal assistant**

A software agent that can perform tasks or services for an individual based on verbal commands. Sometimes the term "chatbot" is used to refer to virtual assistants generally or specifically accessed by online chat (or in some cases online chat programs that are exclusively for entertainment purposes). Some virtual assistants are able to interpret human speech and respond via synthesized voices. Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other basic tasks such as email, to-do lists, and calendars with verbal commands.<sup>[213]</sup>

### **interpretation**

An assignment of meaning to the symbols of a formal language. Many formal languages used in mathematics, logic, and theoretical computer science are defined in solely syntactic terms, and as such do not have any meaning until they are given some interpretation. The general study of interpretations of formal languages is called formal semantics.

### **intrinsic motivation**

An intelligent agent is intrinsically motivated to act if the information content alone, of the experience resulting from the action, is the motivating factor. Information content in this context is measured in the information theory sense as quantifying uncertainty. A typical intrinsic motivation is to search for unusual (surprising) situations, in contrast to a typical extrinsic motivation such as the search for food. Intrinsically motivated artificial agents display behaviours akin to exploration and curiosity.<sup>[214]</sup>

### **issue tree**

A graphical breakdown of a question that dissects it into its different components vertically and that progresses into details as it reads to the right.<sup>[215]:47</sup> Issue trees are useful in problem solving to identify the root causes of a problem as well as to identify its potential

solutions. They also provide a reference point to see how each piece fits into the whole picture of a problem.<sup>[216]</sup>

## J

---

### **junction tree algorithm**

A method used in machine learning to extract marginalization in general graphs. In essence, it entails performing belief propagation on a modified graph called a junction tree. The graph is called a tree because it branches into different sections of data; nodes of variables are the branches.<sup>[217]</sup>

## K

---

### **kernel method**

In machine learning, kernel methods are a class of algorithms for pattern analysis, whose best known member is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations (e.g., cluster analysis, rankings, principal components, correlations, classifications) in datasets.

### **KL-ONE**

A well-known knowledge representation system in the tradition of semantic networks and frames; that is, it is a frame language. The system is an attempt to overcome semantic indistinctness in semantic network representations and to explicitly represent conceptual information as a structured inheritance network.<sup>[218][219][220]</sup>

### **k-nearest neighbors**

A non-parametric supervised learning method first developed by Evelyn Fix and Joseph Hodges in 1951,<sup>[221]</sup> and later expanded by Thomas Cover.<sup>[217]</sup> It is used for classification and regression.

### **knowledge acquisition**

The process used to define the rules and ontologies required for a knowledge-based system. The phrase was first used in conjunction with expert systems to describe the initial tasks associated with developing an expert system, namely finding and interviewing domain experts and capturing their knowledge via rules, objects, and frame-based ontologies.

### **knowledge-based system (KBS)**

A computer program that reasons and uses a knowledge base to solve complex problems. The term is broad and refers to many different kinds of systems. The one common theme that unites all knowledge based systems is an attempt to represent knowledge explicitly and a reasoning system that allows it to derive new knowledge. Thus, a knowledge-based system has two distinguishing features: a knowledge base and an inference engine.

### **knowledge distillation**

The process of transferring knowledge from a large machine learning model to a smaller one.

### **knowledge engineering (KE)**

All technical, scientific, and social aspects involved in building, maintaining, and using knowledge-based systems.

### **knowledge extraction**

The creation of knowledge from structured (relational databases, XML) and unstructured (text, documents, images) sources. The resulting knowledge needs to be in a machine-readable and machine-interpretable format and must represent knowledge in a manner that facilitates inferencing. Although it is methodically similar to information extraction and ETL, the main criterion is that the extraction result goes beyond the creation of structured information or the transformation into a relational schema. It requires either the reuse of existing formal knowledge (reusing identifiers or ontologies) or the generation of a schema based on the source data.

### **knowledge Interchange Format (KIF)**

A computer language designed to enable systems to share and reuse information from knowledge-based systems. KIF is similar to frame languages such as KL-ONE and LOOM but unlike such language its primary role is not intended as a framework for the expression or use of knowledge but rather for the interchange of knowledge between systems. The designers of KIF likened it to PostScript. PostScript was not designed primarily as a language to store and manipulate documents but rather as an interchange format for systems and devices to share documents. In the same way KIF is meant to facilitate sharing of knowledge across different systems that use different languages, formalisms, platforms, etc.

### **knowledge representation and reasoning (KR<sup>2</sup> or KR&R)**

The field of artificial intelligence dedicated to representing information about the world in a form that a computer system can utilize to solve complex tasks such as diagnosing a medical condition or having a dialog in a natural language. Knowledge representation incorporates findings from psychology<sup>[222]</sup> about how humans solve problems and represent knowledge in order to design formalisms that will make complex systems easier to design and build. Knowledge representation and reasoning also incorporates findings from logic to automate various kinds of *reasoning*, such as the application of rules or the relations of sets and subsets.<sup>[223]</sup> Examples of knowledge representation formalisms include semantic nets, systems architecture, frames, rules, and ontologies. Examples of automated reasoning engines include inference engines, theorem provers, and classifiers.

### **k-means clustering**

A method of vector quantization, originally from signal processing, that aims to partition  $n$  observations into  $k$  clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.

## **L**

---

### **language model**

A probabilistic model that manipulates natural language.

### **large language model (LLM)**

A language model with a large number of parameters (typically at least a billion) that are adjusted during training. Due to its size, it requires a lot of data and computing capability to train. Large language models are usually based on the transformer architecture.<sup>[224]</sup>

### **lazy learning**

In machine learning, lazy learning is a learning method in which generalization of the training data is, in theory, delayed until a query is made to the system, as opposed to in eager learning, where the system tries to generalize the training data before receiving queries.

### **Lisp (programming language) (LISP)**

A family of programming languages with a long history and a distinctive, fully parenthesized prefix notation.<sup>[225]</sup>

## **logic programming**

A type of programming paradigm which is largely based on formal logic. Any program written in a logic programming language is a set of sentences in logical form, expressing facts and rules about some problem domain. Major logic programming language families include Prolog, answer set programming (ASP), and Datalog.

## **long short-term memory (LSTM)**

An artificial recurrent neural network architecture<sup>[226]</sup> used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections that make it a "general purpose computer" (that is, it can compute anything that a Turing machine can).<sup>[227]</sup> It can not only process single data points (such as images), but also entire sequences of data (such as speech or video).

# **M**

---

## **machine vision (MV)**

The technology and methods used to provide imaging-based automatic inspection and analysis for such applications as automatic inspection, process control, and robot guidance, usually in industry. Machine vision is a term encompassing a large number of technologies, software and hardware products, integrated systems, actions, methods and expertise. Machine vision as a systems engineering discipline can be considered distinct from computer vision, a form of computer science. It attempts to integrate existing technologies in new ways and apply them to solve real world problems. The term is the prevalent one for these functions in industrial automation environments but is also used for these functions in other environments such as security and vehicle guidance.

## **Markov chain**

A stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.<sup>[228][229]</sup>

## **Markov decision process (MDP)**

A discrete time stochastic control process. It provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker. MDPs are useful for studying optimization problems solved via dynamic programming and reinforcement learning.

## **mathematical optimization**

In mathematics, computer science, and operations research, the selection of a best element (with regard to some criterion) from some set of available alternatives.<sup>[230]</sup>

## **machine learning (ML)**

The scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead.

## **machine listening**

A general field of study of algorithms and systems for audio understanding by machine.<sup>[231][232]</sup>

## **machine perception**

The capability of a computer system to interpret data in a manner that is similar to the way humans use their senses to relate to the world around them.<sup>[233][234][235]</sup>

## **mechanism design**



A field in economics and game theory that takes an engineering approach to designing economic mechanisms or incentives, toward desired objectives, in strategic settings, where players act rationally. Because it starts at the end of the game, then goes backwards, it is also called reverse game theory. It has broad applications, from economics and politics (markets, auctions, voting procedures) to networked-systems (internet interdomain routing, sponsored search auctions).

### **mechatronics**

A multidisciplinary branch of engineering that focuses on the engineering of both electrical and mechanical systems, and also includes a combination of robotics, electronics, computer, telecommunications, systems, control, and product engineering.<sup>[236][237]</sup>

### **metabolic network reconstruction and simulation**

Allows for an in-depth insight into the molecular mechanisms of a particular organism. In particular, these models correlate the genome with molecular physiology.<sup>[238]</sup>

### **metaheuristic**

In computer science and mathematical optimization, a metaheuristic is a higher-level procedure or heuristic designed to find, generate, or select a heuristic (partial search algorithm) that may provide a sufficiently good solution to an optimization problem, especially with incomplete or imperfect information or limited computation capacity.<sup>[239][240]</sup> Metaheuristics sample a set of solutions which is too large to be completely sampled.

### **model checking**

In computer science, model checking or property checking is, for a given model of a system, exhaustively and automatically checking whether this model meets a given specification. Typically, one has hardware or software systems in mind, whereas the specification contains safety requirements such as the absence of deadlocks and similar critical states that can cause the system to crash. Model checking is a technique for automatically verifying correctness properties of *finite-state* systems.

### **modus ponens**

In propositional logic, *modus ponens* is a rule of inference.<sup>[241]</sup> It can be summarized as "*P implies Q* and *P* is asserted to be true, therefore *Q* must be true."

### **modus tollens**

In propositional logic, *modus tollens* is a valid argument form and a rule of inference. It is an application of the general truth that if a statement is true, then so is its contrapositive. The inference rule *modus tollens* asserts that the inference from *P implies Q* to the negation of *Q* implies the negation of *P* is valid.

### **Monte Carlo tree search**

In computer science, Monte Carlo tree search (MCTS) is a heuristic search algorithm for some kinds of decision processes.

### **multi-agent system (MAS)**

A computerized system composed of multiple interacting intelligent agents. Multi-agent systems can solve problems that are difficult or impossible for an individual agent or a monolithic system to solve. Intelligence may include methodic, functional, procedural approaches, algorithmic search or reinforcement learning.

### **multilayer perceptron (MLP)**

In deep learning, a multilayer perceptron (MLP) is a name for a modern feedforward neural network consisting of fully connected neurons with nonlinear activation functions, organized in layers, notable for being able to distinguish data that is not linearly separable.<sup>[242]</sup>

### **multi-swarm optimization**

A variant of particle swarm optimization (PSO) based on the use of multiple sub-swarms instead of one (standard) swarm. The general approach in multi-swarm optimization is that each sub-swarm focuses on a specific region while a specific diversification method decides where and when to launch the sub-swarms. The multi-swarm framework is especially fitted for the optimization on multi-modal problems, where multiple (local) optima exist.

### **mutation**

A genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to a better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set too high, the search will turn into a primitive random search.

### **Mycin**

An early backward chaining expert system that used artificial intelligence to identify bacteria causing severe infections, such as bacteremia and meningitis, and to recommend antibiotics, with the dosage adjusted for patient's body weight – the name derived from the antibiotics themselves, as many antibiotics have the suffix "-mycin". The MYCIN system was also used for the diagnosis of blood clotting diseases.

## **N**

---

### **naive Bayes classifier**

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

### **naive semantics**

An approach used in computer science for representing basic knowledge about a specific domain, and has been used in applications such as the representation of the meaning of natural language sentences in artificial intelligence applications. In a general setting the term has been used to refer to the use of a limited store of generally understood knowledge about a specific domain in the world, and has been applied to fields such as the knowledge based design of data schemas.<sup>[243]</sup>

### **name binding**

In programming languages, name binding is the association of entities (data and/or code) with identifiers.<sup>[244]</sup> An identifier bound to an object is said to reference that object. Machine languages have no built-in notion of identifiers, but name-object bindings as a service and notation for the programmer is implemented by programming languages. Binding is intimately connected with scoping, as scope determines which names bind to which objects – at which locations in the program code (lexically) and in which one of the possible execution paths (temporally). Use of an identifier *id* in a context that establishes a binding for *id* is called a binding (or defining) occurrence. In all other occurrences (e.g., in expressions, assignments, and subprogram calls), an identifier stands for what it is bound to; such occurrences are called applied occurrences.

### **named-entity recognition (NER)**

A subtask of information extraction that seeks to locate and classify named entity mentions in unstructured text into pre-defined categories such as the person names,

organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.

### **named graph**

A key concept of Semantic Web architecture in which a set of Resource Description Framework statements (a graph) are identified using a URI,<sup>[245]</sup> allowing descriptions to be made of that set of statements such as context, provenance information or other such metadata. Named graphs are a simple extension of the RDF data model<sup>[246]</sup> through which graphs can be created but the model lacks an effective means of distinguishing between them once published on the Web at large.

### **natural language generation (NLG)**

A software process that transforms structured data into plain-English content. It can be used to produce long-form content for organizations to automate custom reports, as well as produce custom content for a web or mobile application. It can also be used to generate short blurbs of text in interactive conversations (a chatbot) which might even be read out loud by a text-to-speech system.

### **natural language processing (NLP)**

A subfield of computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data.

### **natural language programming**

An ontology-assisted way of programming in terms of natural-language sentences, e.g. English.<sup>[247]</sup>

### **network motif**

All networks, including biological networks, social networks, technological networks (e.g., computer networks and electrical circuits) and more, can be represented as graphs, which include a wide variety of subgraphs. One important local property of networks are so-called network motifs, which are defined as recurrent and statistically significant subgraphs or patterns.

### **neural machine translation (NMT)**

An approach to machine translation that uses a large artificial neural network to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model.

### **neural network**

A **neural network** can refer to either a neural circuit of biological neurons (sometimes also called a *biological neural network*), or a network of artificial neurons or nodes in the case of an artificial neural network.<sup>[248]</sup> Artificial neural networks are used for solving artificial intelligence (AI) problems; they model connections of biological neurons as weights between nodes. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred to as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be -1 and 1.

### **neural Turing machine (NTM)**

A recurrent neural network model. NTMs combine the fuzzy pattern matching capabilities of neural networks with the algorithmic power of programmable computers. An NTM has a neural network controller coupled to external memory resources, which it interacts with through attentional mechanisms. The memory interactions are differentiable end-to-end, making it possible to optimize them using gradient descent.<sup>[249]</sup> An NTM with a long short-term memory (LSTM) network controller can infer simple algorithms such as copying, sorting, and associative recall from examples alone.<sup>[250]</sup>

## **neuro-fuzzy**

Combinations of artificial neural networks and fuzzy logic.

## **neurocybernetics**

A direct communication pathway between an enhanced or wired brain and an external device. BCI differs from neuromodulation in that it allows for bidirectional information flow. BCIs are often directed at researching, mapping, assisting, augmenting, or repairing human cognitive or sensory-motor functions.<sup>[251]</sup>

## **neuromorphic engineering**

A concept describing the use of very-large-scale integration (VLSI) systems containing electronic analog circuits to mimic neuro-biological architectures present in the nervous system.<sup>[252]</sup> In recent times, the term *neuromorphic* has been used to describe analog, digital, mixed-mode analog/digital VLSI, and software systems that implement models of neural systems (for perception, motor control, or multisensory integration). The implementation of neuromorphic computing on the hardware level can be realized by oxide-based memristors,<sup>[253]</sup> spintronic memories,<sup>[254]</sup> threshold switches, and transistors.<sup>[255][256][257][258]</sup>

## **node**

A basic unit of a data structure, such as a linked list or tree data structure. Nodes contain data and also may link to other nodes. Links between nodes are often implemented by pointers.

## **nondeterministic algorithm**

An algorithm that, even for the same input, can exhibit different behaviors on different runs, as opposed to a deterministic algorithm.

## **nouvelle AI**

Nouvelle AI differs from classical AI by aiming to produce robots with intelligence levels similar to insects. Researchers believe that intelligence can emerge organically from simple behaviors as these intelligences interacted with the "real world", instead of using the constructed worlds which symbolic AIs typically needed to have programmed into them.<sup>[259]</sup>

## **NP**

In computational complexity theory, NP (nondeterministic polynomial time) is a complexity class used to classify decision problems. NP is the set of decision problems for which the problem instances, where the answer is "yes", have proofs verifiable in polynomial time.<sup>[260][Note 1]</sup>

## **NP-completeness**

In computational complexity theory, a problem is NP-complete when it can be solved by a restricted class of brute force search algorithms and it can be used to simulate any other problem with a similar algorithm. More precisely, each input to the problem should be associated with a set of solutions of polynomial length, whose validity can be tested quickly (in polynomial time<sup>[261]</sup>), such that the output for any input is "yes" if the solution set is non-empty and "no" if it is empty.

## **NP-hardness**

In computational complexity theory, the defining property of a class of problems that are, informally, "at least as hard as the hardest problems in NP". A simple example of an NP-hard problem is the subset sum problem.

# O

---

## **Occam's razor**

The problem-solving principle that states that when presented with competing hypotheses that make the same predictions, one should select the solution with the fewest assumptions;<sup>[262]</sup> the principle is not meant to filter out hypotheses that make different predictions. The idea is attributed to the English Franciscan friar William of Ockham (c. 1287–1347), a scholastic philosopher and theologian.

## **offline learning**

A machine learning training approach in which a model is trained on a fixed dataset that is not updated during the learning process.

## **online machine learning**

A method of machine learning in which data becomes available in a sequential order and is used to update the best predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once. Online learning is a common technique used in areas of machine learning where it is computationally infeasible to train over the entire dataset, requiring the need of out-of-core algorithms. It is also used in situations where it is necessary for the algorithm to dynamically adapt to new patterns in the data, or when the data itself is generated as a function of time.

## **ontology learning**

The automatic or semi-automatic creation of ontologies, including extracting the corresponding domain's terms and the relationships between the concepts that these terms represent from a corpus of natural language text, and encoding them with an ontology language for easy retrieval.

## **OpenAI**

The for-profit corporation OpenAI LP, whose parent organization is the non-profit organization OpenAI Inc<sup>[263]</sup> that conducts research in the field of artificial intelligence (AI) with the stated aim to promote and develop friendly AI in such a way as to benefit humanity as a whole.

## **OpenCog**

A project that aims to build an open-source artificial intelligence framework. OpenCog Prime is an architecture for robot and virtual embodied cognition that defines a set of interacting components designed to give rise to human-equivalent artificial general intelligence (AGI) as an emergent phenomenon of the whole system.<sup>[264]</sup>

## **Open Mind Common Sense**

An artificial intelligence project based at the Massachusetts Institute of Technology (MIT) Media Lab whose goal is to build and utilize a large commonsense knowledge base from the contributions of many thousands of people across the Web.

## **open-source software (OSS)**

A type of computer software in which source code is released under an license in which the copyright holder grants users the rights to study, change, and distribute the software to anyone and for any purpose.<sup>[265]</sup> Open-source software may be developed in an collaborative public manner. Open-source software is a prominent example of open collaboration.<sup>[266]</sup>

## **overfitting**

"The production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit to additional data or predict future observations reliably".<sup>[267]</sup> In other words, an overfitted model memorizes training data details but cannot generalize to new data. Conversely, an **underfitted** model is too simple to capture the complexity of the training data.

## P

---

### **partial order reduction**

A technique for reducing the size of the state-space to be searched by a model checking or automated planning and scheduling algorithm. It exploits the commutativity of concurrently executed transitions, which result in the same state when executed in different orders.

### **partially observable Markov decision process (POMDP)**

A generalization of a Markov decision process (MDP). A POMDP models an agent decision process in which it is assumed that the system dynamics are determined by an MDP, but the agent cannot directly observe the underlying state. Instead, it must maintain a probability distribution over the set of possible states, based on a set of observations and observation probabilities, and the underlying MDP.

### **particle swarm optimization (PSO)**

A computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position, but is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

### **pathfinding**

The plotting, by a computer application, of the shortest route between two points. It is a more practical variant on solving mazes. This field of research is based heavily on Dijkstra's algorithm for finding a shortest path on a weighted graph.

### **pattern recognition**

Concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories.<sup>[268]</sup>

### **perceptron**

An algorithm for supervised learning of binary classifiers.

### **predicate logic**

A collection of formal systems used in mathematics, philosophy, linguistics, and computer science. First-order logic uses quantified variables over non-logical objects and allows the use of sentences that contain variables, so that rather than propositions such as *Socrates is a man* one can have expressions in the form "there exists x such that x is Socrates and x is a man" and *there exists* is a quantifier while x is a variable.<sup>[182]</sup> This distinguishes it from propositional logic, which does not use quantifiers or relations,<sup>[269]</sup> in this sense, propositional logic is the foundation of first-order logic.

### **predictive analytics**



A variety of statistical techniques from data mining, predictive modelling, and machine learning, that analyze current and historical facts to make predictions about future or otherwise unknown events.<sup>[270][271]</sup>

### **principal component analysis (PCA)**

A statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component, in turn, has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing  $n$  observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

### **principle of rationality**

A principle coined by Karl R. Popper in his Harvard Lecture of 1963, and published in his book *Myth of Framework*.<sup>[272]</sup> It is related to what he called the 'logic of the situation' in an *Economica* article of 1944/1945, published later in his book *The Poverty of Historicism*.<sup>[273]</sup> According to Popper's rationality principle, agents act in the most adequate way according to the objective situation. It is an idealized conception of human behavior which he used to drive his model of situational logic.

### **probabilistic programming (PP)**

A programming paradigm in which probabilistic models are specified and inference for these models is performed automatically.<sup>[274]</sup> It represents an attempt to unify probabilistic modeling and traditional general-purpose programming in order to make the former easier and more widely applicable.<sup>[275][276]</sup> It can be used to create systems that help make decisions in the face of uncertainty. Programming languages used for probabilistic programming are referred to as "Probabilistic programming languages" (PPLs).

### **production system**

A computer program typically used to provide some form of AI, which consists primarily of a set of rules about behavior, but also includes the mechanism necessary to follow those rules as the system responds to states of the world.

### **programming language**

A formal language, which comprises a set of instructions that produce various kinds of output. Programming languages are used in computer programming to implement algorithms.

### **Prolog**

A logic programming language associated with artificial intelligence and computational linguistics.<sup>[277][278][279]</sup> Prolog has its roots in first-order logic, a formal logic, and unlike many other programming languages, Prolog is intended primarily as a declarative programming language: the program logic is expressed in terms of relations, represented as facts and rules. A computation is initiated by running a query over these relations.<sup>[280]</sup>

### **propositional calculus**

A branch of logic which deals with propositions (which can be true or false) and argument flow. Compound propositions are formed by connecting propositions by logical connectives. The propositions without logical connectives are called atomic propositions. Unlike first-order logic, propositional logic does not deal with non-logical objects, predicates about them, or quantifiers. However, all the machinery of propositional logic is included in first-order logic and higher-order logics. In this sense, propositional logic is the foundation of first-order logic and higher-order logic.

## **proximal policy optimization (PPO)**

A reinforcement learning algorithm for training an intelligent agent's decision function to accomplish difficult tasks.

## **Python**

An interpreted, high-level, general-purpose programming language created by Guido van Rossum and first released in 1991. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.<sup>[281]</sup>

## **PyTorch**

A machine learning library based on the Torch library,<sup>[282][283][284]</sup> used for applications such as computer vision and natural language processing,<sup>[285]</sup> originally developed by Meta AI and now part of the Linux Foundation umbrella.<sup>[286][287][288][289]</sup>

# Q

---

## **Q-learning**

A model-free reinforcement learning algorithm for learning the value of an action in a particular state.

## **qualification problem**

In philosophy and artificial intelligence (especially knowledge-based systems), the qualification problem is concerned with the impossibility of listing *all* of the preconditions required for a real-world action to have its intended effect.<sup>[290][291]</sup> It might be posed as *how to deal with the things that prevent me from achieving my intended result*. It is strongly connected to, and opposite the ramification side of, the frame problem.<sup>[290]</sup>

## **quantifier**

In logic, quantification specifies the quantity of specimens in the domain of discourse that satisfy an open formula. The two most common quantifiers mean "for all" and "there exists". For example, in arithmetic, quantifiers allow one to say that the natural numbers go on forever, by writing that *for all*  $n$  (where  $n$  is a natural number), there is another number (say, the successor of  $n$ ) which is one bigger than  $n$ .

## **quantum computing**

The use of quantum-mechanical phenomena such as superposition and entanglement to perform computation. A quantum computer is used to perform such computation, which can be implemented theoretically or physically.<sup>[292]:1-5</sup>

## **query language**

Query languages or data query languages (DQLs) are computer languages used to make queries in databases and information systems. Broadly, query languages can be classified according to whether they are database query languages or information retrieval query languages. The difference is that a database query language attempts to give factual answers to factual questions, while an information retrieval query language attempts to find documents containing information that is relevant to an area of inquiry.

# R

---

## **R programming language**

A programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing.<sup>[293]</sup> The R language is

widely used among statisticians and data miners for developing statistical software<sup>[294]</sup> and data analysis.<sup>[295]</sup>

### **radial basis function network**

In the field of mathematical modeling, a radial basis function network is an artificial neural network that uses radial basis functions as activation functions. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters. Radial basis function networks have many uses, including function approximation, time series prediction, classification, and system control. They were first formulated in a 1988 paper by Broomhead and Lowe, both researchers at the Royal Signals and Radar Establishment.<sup>[296][297][298]</sup>

### **random forest**

An ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.<sup>[299][300]</sup> Random decision forests correct for decision trees' habit of overfitting to their training set.<sup>[301]</sup>

### **reasoning system**

In information technology a reasoning system is a software system that generates conclusions from available knowledge using logical techniques such as deduction and induction. Reasoning systems play an important role in the implementation of artificial intelligence and knowledge-based systems.

### **recurrent neural network (RNN)**

A class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition<sup>[302]</sup> or speech recognition.<sup>[303][304]</sup>

### **regression analysis**

A set of statistical processes for estimating the relationships between a dependent variable (often called the outcome or response variable, or *label* in machine learning) and one or more error-free independent variables (often called regressors, predictors, covariates, explanatory variables, or features). The most common form of regression analysis is linear regression, in which one finds the line (or a more complex linear combination) that most closely fits the data according to a specific mathematical criterion.

### **regularization**

A set of techniques such as dropout, early stopping, and L1 and L2 regularization to reduce overfitting and underfitting when training a learning algorithm.

### **reinforcement learning (RL)**

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. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised and unsupervised learning. It differs from supervised learning in that labelled input/output pairs need not be presented, and sub-optimal actions need not be explicitly corrected. Instead the focus is finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge).<sup>[305]</sup>

### **reinforcement learning from human feedback (RLHF)**

A technique that involve training a "reward model" to predict how humans rate the quality of generated content, and then training a generative AI model to satisfy this reward model via reinforcement learning. It can be used for example to make the generative AI model more truthful or less harmful.<sup>[306]</sup>

## **representation learning**

See [feature learning](#).

## **reservoir computing**

A framework for computation that may be viewed as an extension of neural networks.<sup>[307]</sup> Typically an input signal is fed into a fixed (random) [dynamical system](#) called a *reservoir* and the dynamics of the reservoir map the input to a higher dimension. Then a simple *readout* mechanism is trained to read the state of the reservoir and map it to the desired output. The main benefit is that training is performed only at the readout stage and the reservoir is fixed. [Liquid-state machines](#)<sup>[308]</sup> and [echo state networks](#)<sup>[309]</sup> are two major types of reservoir computing.<sup>[310]</sup>

## **Resource Description Framework (RDF)**

A family of [World Wide Web Consortium](#) (W3C) [specifications](#)<sup>[311]</sup> originally designed as a [metadata data model](#). It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax notations and [data serialization](#) formats. It is also used in [knowledge management](#) applications.

## **restricted Boltzmann machine (RBM)**

A [generative stochastic artificial neural network](#) that can learn a [probability distribution](#) over its set of inputs.

## **Rete algorithm**

A [pattern matching algorithm](#) for implementing rule-based systems. The algorithm was developed to efficiently apply many [rules](#) or patterns to many objects, or [facts](#), in a [knowledge base](#). It is used to determine which of the system's rules should fire based on its data store, its facts.

## **robotics**

An interdisciplinary branch of science and engineering that includes [mechanical engineering](#), [electronic engineering](#), [information engineering](#), [computer science](#), and others. Robotics deals with the design, construction, operation, and use of [robots](#), as well as [computer systems](#) for their control, [sensory feedback](#), and [information processing](#).

## **rule-based system**

In [computer science](#), a rule-based system is used to store and manipulate knowledge to interpret information in a useful way. It is often used in artificial intelligence applications and research. Normally, the term *rule-based system* is applied to systems involving human-crafted or curated rule sets. Rule-based systems constructed using automatic rule inference, such as [rule-based machine learning](#), are normally excluded from this system type.

# **S**

---

## **satisfiability**

In [mathematical logic](#), satisfiability and [validity](#) are elementary concepts of [semantics](#). A formula is *satisfiable* if it is possible to find an [interpretation](#) (model) that makes the formula true.<sup>[312]</sup> A formula is *valid* if all interpretations make the formula true. The opposites of these concepts are unsatisfiability and invalidity, that is, a formula is *unsatisfiable* if none of the interpretations make the formula true, and *invalid* if some such interpretation makes the formula false. These four concepts are related to each other in a manner exactly analogous to [Aristotle's square of opposition](#).

## **search algorithm**

Any [algorithm](#) which solves the [search problem](#), namely, to retrieve information stored within some data structure, or calculated in the [search space](#) of a [problem domain](#), either

with discrete or continuous values.

### **selection**

The stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (using the crossover operator).

### **self-management**

The process by which computer systems manage their own operation without human intervention.

### **semantic network**

A knowledge base that represents semantic relations between concepts in a network. This is often used as a form of knowledge representation. It is a directed or undirected graph consisting of vertices, which represent concepts, and edges, which represent semantic relations between concepts,<sup>[313]</sup> mapping or connecting semantic fields.

### **semantic reasoner**

A piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with. The inference rules are commonly specified by means of an ontology language, and often a description logic language. Many reasoners use first-order predicate logic to perform reasoning; inference commonly proceeds by forward chaining and backward chaining.

### **semantic query**

Allows for queries and analytics of associative and contextual nature. Semantic queries enable the retrieval of both explicitly and implicitly derived information based on syntactic, semantic and structural information contained in data. They are designed to deliver precise results (possibly the distinctive selection of one single piece of information) or to answer more fuzzy and wide-open questions through pattern matching and digital reasoning.

### **semantics**

In programming language theory, semantics is the field concerned with the rigorous mathematical study of the meaning of programming languages. It does so by evaluating the meaning of syntactically valid strings defined by a specific programming language, showing the computation involved. In such a case that the evaluation would be of syntactically invalid strings, the result would be non-computation. Semantics describes the processes a computer follows when executing a program in that specific language. This can be shown by describing the relationship between the input and output of a program, or an explanation of how the program will be executed on a certain platform, hence creating a model of computation.

### **semi-supervised learning**

A machine learning training paradigm characterized by using a combination of a small amount of human-labeled data (used exclusively in supervised learning), followed by a large amount of unlabeled data (used exclusively in unsupervised learning).

### **sensor fusion**

The combining of sensory data or data derived from disparate sources such that the resulting information has less uncertainty than would be possible when these sources were used individually.

### **separation logic**

An extension of Hoare logic, a way of reasoning about programs. The assertion language of separation logic is a special case of the logic of bunched implications (BI).<sup>[314]</sup>

### **similarity learning**

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

### **simulated annealing (SA)**

A probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space for an optimization problem.

### **situated approach**

In artificial intelligence research, the situated approach builds agents that are designed to behave effectively successfully in their environment. This requires designing AI "from the bottom-up" by focussing on the basic perceptual and motor skills required to survive. The situated approach gives a much lower priority to abstract reasoning or problem-solving skills.

### **situation calculus**

A logic formalism designed for representing and reasoning about dynamical domains.

### **Selective Linear Definite clause resolution**

The basic inference rule used in logic programming. It is a refinement of resolution, which is both sound and refutation complete for Horn clauses.

### **software**

A collection of data or computer instructions that tell the computer how to work. This is in contrast to physical hardware, from which the system is built and actually performs the work. In computer science and software engineering, computer software is all information processed by computer systems, programs and data. Computer software includes computer programs, libraries and related non-executable data, such as online documentation or digital media.

### **software engineering**

The application of engineering to the development of software in a systematic method.<sup>[315][316][317]</sup>

### **spatial-temporal reasoning**

An area of artificial intelligence which draws from the fields of computer science, cognitive science, and cognitive psychology. The theoretic goal—on the cognitive side—involves representing and reasoning spatial-temporal knowledge in mind. The applied goal—on the computing side—involves developing high-level control systems of automata for navigating and understanding time and space.

### **SPARQL**

An RDF query language—that is, a semantic query language for databases—able to retrieve and manipulate data stored in Resource Description Framework (RDF) format.<sup>[318][319]</sup>

### **sparse dictionary learning**

A feature learning method aimed at finding a sparse representation of the input data in the form of a linear combination of basic elements as well as those basic elements themselves.

### **speech recognition**

An interdisciplinary subfield of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as automatic speech recognition (ASR), computer speech recognition or speech to text (STT). It incorporates knowledge and research in the linguistics, computer science, and electrical engineering fields.



### **spiking neural network (SNN)**

An artificial neural network that more closely mimics a natural neural network.<sup>[320]</sup> In addition to neuronal and synaptic state, SNNs incorporate the concept of time into their Operating Model.

### **state**

In information technology and computer science, a program is described as stateful if it is designed to remember preceding events or user interactions,<sup>[321]</sup> the remembered information is called the state of the system.

### **statistical classification**

In machine learning and statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. Examples are assigning a given email to the "spam" or "non-spam" class, and assigning a diagnosis to a given patient based on observed characteristics of the patient (sex, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition.

### **state-action-reward-state-action (SARSA)**

A reinforcement learning algorithm for learning a Markov decision process policy.

### **statistical relational learning (SRL)**

A subdiscipline of artificial intelligence and machine learning that is concerned with domain models that exhibit both uncertainty (which can be dealt with using statistical methods) and complex, relational structure.<sup>[322][323]</sup> Note that SRL is sometimes called Relational Machine Learning (RML) in the literature. Typically, the knowledge representation formalisms developed in SRL use (a subset of) first-order logic to describe relational properties of a domain in a general manner (universal quantification) and draw upon probabilistic graphical models (such as Bayesian networks or Markov networks) to model the uncertainty; some also build upon the methods of inductive logic programming.

### **stochastic optimization (SO)**

Any optimization method that generates and uses random variables. For stochastic problems, the random variables appear in the formulation of the optimization problem itself, which involves random objective functions or random constraints. Stochastic optimization methods also include methods with random iterates. Some stochastic optimization methods use random iterates to solve stochastic problems, combining both meanings of stochastic optimization.<sup>[324]</sup> Stochastic optimization methods generalize deterministic methods for deterministic problems.

### **stochastic semantic analysis**

An approach used in computer science as a semantic component of natural language understanding. Stochastic models generally use the definition of segments of words as basic semantic units for the semantic models, and in some cases involve a two layered approach.<sup>[325]</sup>

### **Stanford Research Institute Problem Solver (STRIPS)**

An automated planner developed by Richard Fikes and Nils Nilsson in 1971 at SRI International.

### **subject-matter expert (SME)**

A person who has accumulated great knowledge in a particular field or topic, demonstrated by the person's degree, licensure, and/or through years of professional experience with the subject.

### **superintelligence**

A hypothetical agent that possesses intelligence far surpassing that of the brightest and most gifted human minds. Superintelligence may also refer to a property of problem-solving systems (e.g., superintelligent language translators or engineering assistants)

whether or not these high-level intellectual competencies are embodied in agents that act within the physical world. A superintelligence may or may not be created by an intelligence explosion and be associated with a technological singularity.

### **supervised learning**

The machine learning task of learning a function that maps an input to an output based on example input-output pairs.<sup>[326]</sup> It infers a function from *labeled training data* consisting of a set of *training examples*.<sup>[327]</sup> In supervised learning, each example is a *pair* consisting of an input object (typically a vector) and a desired output value (also called the *supervisory signal*). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way (see inductive bias).

### **support vector machines**

In machine learning, support vector machines (SVMs, also support vector networks<sup>[328]</sup>) are supervised learning models with associated learning algorithms that analyze data used for classification and regression.

### **swarm intelligence (SI)**

The collective behavior of decentralized, self-organized systems, either natural or artificial. The expression was introduced in the context of cellular robotic systems.<sup>[329]</sup>

### **symbolic artificial intelligence**

The term for the collection of all methods in artificial intelligence research that are based on high-level "symbolic" (human-readable) representations of problems, logic, and search.

### **synthetic intelligence (SI)**

An alternative term for artificial intelligence which emphasizes that the intelligence of machines need not be an imitation or in any way artificial; it can be a genuine form of intelligence.<sup>[330][331]</sup>

### **systems neuroscience**

A subdiscipline of neuroscience and systems biology that studies the structure and function of neural circuits and systems. It is an umbrella term, encompassing a number of areas of study concerned with how nerve cells behave when connected together to form neural pathways, neural circuits, and larger brain networks.

## **T**

---

### **technological singularity**

A hypothetical point in the future when technological growth becomes uncontrollable and irreversible, resulting in unfathomable changes to human civilization.<sup>[332][333][334]</sup>

### **temporal difference learning**

A class of model-free reinforcement learning methods which learn by bootstrapping from the current estimate of the value function. These methods sample from the environment, like Monte Carlo methods, and perform updates based on current estimates, like dynamic programming methods.<sup>[335]</sup>

### **tensor network theory**

A theory of brain function (particularly that of the cerebellum) that provides a mathematical model of the transformation of sensory space-time coordinates into motor coordinates and vice versa by cerebellar neuronal networks. The theory was developed as a

geometrization of brain function (especially of the central nervous system) using tensors.<sup>[336][337]</sup>

### **TensorFlow**

A free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.<sup>[338]</sup>

### **theoretical computer science (TCS)**

A subset of general computer science and mathematics that focuses on more mathematical topics of computing and includes the theory of computation.

### **theory of computation**

In theoretical computer science and mathematics, the theory of computation is the branch that deals with how efficiently problems can be solved on a model of computation, using an algorithm. The field is divided into three major branches: automata theory and languages, computability theory, and computational complexity theory, which are linked by the question: "*What are the fundamental capabilities and limitations of computers?*".<sup>[339]</sup>

### **Thompson sampling**

A heuristic for choosing actions that addresses the exploration-exploitation dilemma in the multi-armed bandit problem. It consists in choosing the action that maximizes the expected reward with respect to a randomly drawn belief.<sup>[340][341]</sup>

### **time complexity**

The computational complexity that describes the amount of time it takes to run an algorithm. Time complexity is commonly estimated by counting the number of elementary operations performed by the algorithm, supposing that each elementary operation takes a fixed amount of time to perform. Thus, the amount of time taken and the number of elementary operations performed by the algorithm are taken to differ by at most a constant factor.

### **transfer learning**

A machine learning technique in which knowledge learned from a task is reused in order to boost performance on a related task.<sup>[342]</sup> For example, for image classification, knowledge gained while learning to recognize cars could be applied when trying to recognize trucks.

### **transformer**

A type of deep learning architecture that exploits a multi-head attention mechanism. Transformers address some of the limitations of long short-term memory, and became widely used in natural language processing, although it can also process other types of data such as images in the case of vision transformers.<sup>[343]</sup>

### **transhumanism**

An international philosophical movement that advocates for the transformation of the human condition by developing and making widely available sophisticated technologies to greatly enhance human intellect and physiology.<sup>[344][345]</sup>

### **transition system**

In theoretical computer science, a transition system is a concept used in the study of computation. It is used to describe the potential behavior of discrete systems. It consists of states and transitions between states, which may be labeled with labels chosen from a set; the same label may appear on more than one transition. If the label set is a singleton, the system is essentially unlabeled, and a simpler definition that omits the labels is possible.

### **tree traversal**

A form of graph traversal and refers to the process of visiting (checking and/or updating) each node in a tree data structure, exactly once. Such traversals are classified by the order in which the nodes are visited.

### **true quantified Boolean formula**

In computational complexity theory, the language TQBF is a formal language consisting of the true quantified Boolean formulas. A (fully) quantified Boolean formula is a formula in quantified propositional logic where every variable is quantified (or bound), using either existential or universal quantifiers, at the beginning of the sentence. Such a formula is equivalent to either true or false (since there are no free variables). If such a formula evaluates to true, then that formula is in the language TQBF. It is also known as QSAT (Quantified SAT).

### **Turing machine**

A mathematical model of computation describing an abstract machine<sup>[346]</sup> that manipulates symbols on a strip of tape according to a table of rules.<sup>[347]</sup> Despite the model's simplicity, it is capable of implementing any algorithm.<sup>[348]</sup>

### **Turing test**

A test of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, that of a human, developed by Alan Turing in 1950. Turing proposed that a human evaluator would judge natural language conversations between a human and a machine designed to generate human-like responses. The evaluator would be aware that one of the two partners in conversation is a machine, and all participants would be separated from one another. The conversation would be limited to a text-only channel such as a computer keyboard and screen so the result would not depend on the machine's ability to render words as speech.<sup>[349]</sup> If the evaluator cannot reliably tell the machine from the human, the machine is said to have passed the test. The test results do not depend on the machine's ability to give correct answers to questions, only how closely its answers resemble those a human would give.

### **type system**

In programming languages, a set of rules that assigns a property called type to the various constructs of a computer program, such as variables, expressions, functions, or modules.<sup>[350]</sup> These types formalize and enforce the otherwise implicit categories the programmer uses for algebraic data types, data structures, or other components (e.g. "string", "array of float", "function returning boolean"). The main purpose of a type system is to reduce possibilities for bugs in computer programs<sup>[351]</sup> by defining interfaces between different parts of a computer program, and then checking that the parts have been connected in a consistent way. This checking can happen statically (at compile time), dynamically (at run time), or as a combination of static and dynamic checking. Type systems have other purposes as well, such as expressing business rules, enabling certain compiler optimizations, allowing for multiple dispatch, providing a form of documentation, etc.

## **U**

---

### **unsupervised learning**

A type of self-organized Hebbian learning that helps find previously unknown patterns in data set without pre-existing labels. It is also known as self-organization and allows modeling probability densities of given inputs.<sup>[352]</sup> It is one of the three basic paradigms of machine learning, alongside supervised and reinforcement learning. Semi-supervised

learning has also been described and is a hybridization of supervised and unsupervised techniques.

## V

---

### **vision processing unit (VPU)**

A type of microprocessor designed to accelerate machine vision tasks.<sup>[353][354]</sup>

### **Value-alignment complete**

Analogous to an AI-complete problem, a value-alignment complete problem is a problem where the AI control problem needs to be fully solved to solve it.

## W

---

### **Watson**

A question-answering computer system capable of answering questions posed in natural language,<sup>[355]</sup> developed in IBM's DeepQA project by a research team led by principal investigator David Ferrucci.<sup>[356]</sup> Watson was named after IBM's first CEO, industrialist Thomas J. Watson.<sup>[357][358]</sup>

### **weak AI**

Artificial intelligence that is focused on one narrow task.<sup>[359][360][361]</sup>

### **weak supervision**

See semi-supervised learning.

### **word embedding**

A representation of a word in natural language processing. Typically, the representation is a real-valued vector that encodes the meaning of the word in such a way that words that are closer in the vector space are expected to be similar in meaning.<sup>[362]</sup>

## X

---

### **XGBoost**

Short for eXtreme Gradient Boosting, XGBoost<sup>[363]</sup> is an open-source software library which provides a regularizing gradient boosting framework for multiple programming languages.

## References

---

1. For example: Josephson, John R.; Josephson, Susan G., eds. (1994). *Abductive Inference: Computation, Philosophy, Technology*. Cambridge, UK; New York: Cambridge University Press. doi:10.1017/CBO9780511530128 (<https://doi.org/10.1017%2FCBO9780511530128>). ISBN 978-0521434614. OCLC 28149683 (<https://search.worldcat.org/oclc/28149683>).

2. "Retroduction" (<https://web.archive.org/web/20220705094016/http://www.commens.org/dictionary/term/retroduction>). *Commens – Digital Companion to C. S. Peirce*. Mats Bergman, Sami Paavola & João Queiroz. Archived from the original (<https://commens.org/dictionary/term/retroduction>) on 5 July 2022. Retrieved 24 August 2014.
3. Sheikholeslami, Sina (2019). *Ablation Programming for Machine Learning* (<https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-258413>).
4. Colburn, Timothy; Shute, Gary (5 June 2007). "Abstraction in Computer Science". *Minds and Machines*. **17** (2): 169–184. doi:10.1007/s11023-007-9061-7 (<https://doi.org/10.1007%2Fs11023-007-9061-7>). ISSN 0924-6495 (<https://search.worldcat.org/issn/0924-6495>). S2CID 5927969 (<https://api.semanticscholar.org/CorpusID:5927969>).
5. Kramer, Jeff (1 April 2007). "Is abstraction the key to computing?". *Communications of the ACM*. **50** (4): 36–42. CiteSeerX 10.1.1.120.6776 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.120.6776>). doi:10.1145/1232743.1232745 (<https://doi.org/10.1145%2F1232743.1232745>). ISSN 0001-0782 (<https://search.worldcat.org/issn/0001-0782>). S2CID 12481509 (<https://api.semanticscholar.org/CorpusID:12481509>).
6. Michael Gelfond, Vladimir Lifschitz (1998) "Action Languages (<https://ep.liu.se/ea/cis/1998/016/>)", *Linköping Electronic Articles in Computer and Information Science*, vol 3, nr 16.
7. Jang, Jyh-Shing R (1991). *Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm* (<https://aaai.org/Papers/AAAI/1991/AAAI91-119.pdf>) (PDF). Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA, USA, July 14–19. Vol. 2. pp. 762–767.
8. Jang, J.-S.R. (1993). "ANFIS: adaptive-network-based fuzzy inference system". *IEEE Transactions on Systems, Man, and Cybernetics*. **23** (3): 665–685. doi:10.1109/21.256541 (<https://doi.org/10.1109%2F21.256541>). S2CID 14345934 (<https://api.semanticscholar.org/CorpusID:14345934>).
9. Abraham, A. (2005), "Adaptation of Fuzzy Inference System Using Neural Learning", in Nedjah, Nadia; De Macedo Mourelle, Luiza (eds.), *Fuzzy Systems Engineering: Theory and Practice*, Studies in Fuzziness and Soft Computing, vol. 181, Germany: Springer Verlag, pp. 53–83, CiteSeerX 10.1.1.161.6135 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.161.6135>), doi:10.1007/11339366\_3 ([https://doi.org/10.1007%2F11339366\\_3](https://doi.org/10.1007%2F11339366_3)), ISBN 978-3-540-25322-8
10. Jang, Sun, Mizutani (1997) – *Neuro-Fuzzy and Soft Computing* – Prentice Hall, pp 335–368, ISBN 0-13-261066-3
11. Tahmasebi, P. (2012). "A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4268588>). *Computers & Geosciences*. **42**: 18–27. Bibcode:2012CG....42...18T (<https://ui.adsabs.harvard.edu/abs/2012CG....42...18T>). doi:10.1016/j.cageo.2012.02.004 (<https://doi.org/10.1016%2Fj.cageo.2012.02.004>). PMC 4268588 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4268588>). PMID 25540468 (<https://pubmed.ncbi.nlm.nih.gov/25540468>).
12. Tahmasebi, P. (2010). "Comparison of optimized neural network with fuzzy logic for ore grade estimation" (<https://www.researchgate.net/publication/266881168>). *Australian Journal of Basic and Applied Sciences*. **4**: 764–772.
13. Russell, S.J.; Norvig, P. (2002). *Artificial Intelligence: A Modern Approach*. Prentice Hall. ISBN 978-0-13-790395-5.
14. Tao, Jianhua; Tieniu Tan (2005). "Affective Computing: A Review". *Affective Computing and Intelligent Interaction*. Vol. LNCS 3784. Springer. pp. 981–995. doi:10.1007/11573548 (<https://doi.org/10.1007%2F11573548>).
15. El Kaliouby, Rana (November–December 2017). "We Need Computers with Empathy" (<https://wayback.archive-it.org/all/20180707140902/https://technologyreview.com/s/609071/we-need-computers-with-empathy/>). *Technology Review*. Vol. 120, no. 6. p. 8. Archived from the original (<https://technologyreview.com/s/609071/we-need-computers-with-empathy/>) on 7 July 2018. Retrieved 6 November 2018.



16. Comparison of Agent Architectures (<https://hri.cogs.indiana.edu/publications/aaai04ws.pdf>) Archived (<https://web.archive.org/web/20080827222057/https://hri.cogs.indiana.edu/publications/aaai04ws.pdf>) August 27, 2008, at the Wayback Machine
17. "Intel unveils Movidius Compute Stick USB AI Accelerator" (<https://web.archive.org/web/20170811193632/https://v3.co.uk/v3-uk/news/3014293/intel-unveils-movidius-compute-stick-usb-ai-accelerator>). 21 July 2017. Archived from the original (<https://v3.co.uk/v3-uk/news/3014293/intel-unveils-movidius-compute-stick-usb-ai-accelerator>) on 11 August 2017. Retrieved 28 November 2018.
18. "Inspurs unveils GX4 AI Accelerator" (<https://insidehpc.com/2017/06/inspurs-unveils-gx4-ai-accelerator/>). 21 June 2017.
19. Shapiro, Stuart C. (1992). Artificial Intelligence (<https://cse.buffalo.edu/~shapiro/Papers/ai.pdf>) In Stuart C. Shapiro (Ed.), *Encyclopedia of Artificial Intelligence* (Second Edition, pp. 54–57). New York: John Wiley. (Section 4 is on "AI-Complete Tasks".)
20. Solomonoff, R., "A Preliminary Report on a General Theory of Inductive Inference (<https://world.std.com/~rjs/z138.pdf>)", Report V-131, Zator Co., Cambridge, Ma. (Nov. 1960 revision of the Feb. 4, 1960 report).
21. "Artificial intelligence: Google's AlphaGo beats Go master Lee Se-dol" (<https://bbc.com/news/technology-35785875>). *BBC News*. 12 March 2016. Retrieved 17 March 2016.
22. "AlphaGo | DeepMind" (<https://deepmind.com/research/alphago/>). *DeepMind*.
23. "Research Blog: AlphaGo: Mastering the ancient game of Go with Machine Learning" (<https://googleresearch.blogspot.com/2016/01/alphago-mastering-ancient-game-of-go.html>). *Google Research Blog*. 27 January 2016.
24. "Google achieves AI 'breakthrough' by beating Go champion" (<https://bbc.com/news/technology-35420579>). *BBC News*. 27 January 2016.
25. See Dung (1995)
26. See Besnard and Hunter (2001)
27. see Bench-Capon (2002)
28. Definition of AI as the study of intelligent agents:
  - Poole, Mackworth & Goebel 1998, p. 1 (<https://people.cs.ubc.ca/~poole/ci/ch1.pdf>), which provides the version that is used in this article. Note that they use the term "computational intelligence" as a synonym for artificial intelligence.
  - Russell & Norvig (2003) (who prefer the term "rational agent") and write "The whole-agent view is now widely accepted in the field" (Russell & Norvig 2003, p. 55).
  - Nilsson 1998
  - Legg & Hutter 2007.
29. Russell & Norvig 2009, p. 2.
30. "AAAI Corporate Bylaws" (<https://aaai.org/Organization/bylaws.php>).
31. Cipresso, Pietro; Giglioli, Irene Alice Chicchi; Raya, iz; Riva, Giuseppe (7 December 2011). "The Past, Present, and Future of Virtual and Augmented Reality Research: A Network and Cluster Analysis of the Literature" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6232426>). *Frontiers in Psychology*. **9**: 2086. doi:10.3389/fpsyg.2018.02086 (<https://doi.org/10.3389/fpsyg.2018.02086>). PMC 6232426 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6232426>). PMID 30459681 (<https://pubmed.ncbi.nlm.nih.gov/30459681>).
32. Ghallab, Malik; Nau, Dana S.; Traverso, Paolo (2004), *Automated Planning: Theory and Practice* (<https://laas.fr/planning/>), Morgan Kaufmann, ISBN 978-1-55860-856-6
33. Kephart, J.O.; Chess, D.M. (2003), "The vision of autonomic computing", *Computer*, **36**: 41–52, CiteSeerX 10.1.1.70.613 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.70.613>), doi:10.1109/MC.2003.1160055 (<https://doi.org/10.1109/MC.2003.1160055>)

34. Gehrig, Stefan K.; Stein, Fridtjof J. (1999). *Dead reckoning and cartography using stereo vision for an automated car*. IEEE/RSJ International Conference on Intelligent Robots and Systems. Vol. 3. Kyongju. pp. 1507–1512. doi:10.1109/IROS.1999.811692 (<https://doi.org/10.1109%2FIROS.1999.811692>). ISBN 0-7803-5184-3.
35. "Self-driving Uber car kills Arizona woman crossing street" (<https://reuters.com/article/us-autos-selfdriving-uber-idUSKBN1GV296>). Reuters. 20 March 2018.
36. Thrun, Sebastian (2010). "Toward Robotic Cars". *Communications of the ACM*. **53** (4): 99–106. doi:10.1145/1721654.1721679 (<https://doi.org/10.1145%2F1721654.1721679>). S2CID 207177792 (<https://api.semanticscholar.org/CorpusID:207177792>).
37. "Information Engineering Main/Home Page" (<https://web.archive.org/web/20220703164507/http://arno@robots.ox.ac.uk/>). University of Oxford. Archived from the original (<https://robots.ox.ac.uk/>) on 3 July 2022. Retrieved 3 October 2018.
38. Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2016) *Deep Learning*. MIT Press. p. 196. ISBN 9780262035613
39. Nielsen, Michael A. (2015). "Chapter 6" (<https://web.archive.org/web/20220808100458/http://neuralnetworksanddeeplearning.com/chap6.html>). *Neural Networks and Deep Learning*. Archived from the original (<https://neuralnetworksanddeeplearning.com/chap6.html>) on 8 August 2022. Retrieved 5 July 2022.
40. "Deep Networks: Overview – Ufldl" ([https://web.archive.org/web/20220316074444/http://ufldl.stanford.edu/wiki/index.php/Deep\\_Networks:\\_Overview](https://web.archive.org/web/20220316074444/http://ufldl.stanford.edu/wiki/index.php/Deep_Networks:_Overview)). *ufldl.stanford.edu*. Archived from the original ([https://ufldl.stanford.edu/wiki/index.php/Deep\\_Networks:\\_Overview](https://ufldl.stanford.edu/wiki/index.php/Deep_Networks:_Overview)) on 16 March 2022. Retrieved 4 August 2017.
41. Goller, Christoph; Küchler, Andreas (1996). "Learning Task-Dependent Distributed Representations by Backpropagation Through Structure". *Proceedings of International Conference on Neural Networks (ICNN'96)*. Vol. 1. pp. 347–352. CiteSeerX 10.1.1.49.1968 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.1968>). doi:10.1109/ICNN.1996.548916 (<https://doi.org/10.1109%2FICNN.1996.548916>). ISBN 0-7803-3210-5. S2CID 6536466 (<https://api.semanticscholar.org/CorpusID:6536466>).
42. Mozer, M. C. (1995). "A Focused Backpropagation Algorithm for Temporal Pattern Recognition" (<https://www.researchgate.net/publication/243781476>). In Chauvin, Y.; Rumelhart, D. (eds.). *Backpropagation: Theory, architectures, and applications*. Hillsdale, NJ: Lawrence Erlbaum Associates. pp. 137–169. Retrieved 21 August 2017.
43. Robinson, A. J. & Fallside, F. (1987). *The utility driven dynamic error propagation network* (<https://bibsonomy.org/bibtex/269a88ecbac9a51cbf0b4be189c412820/idsia>) (Technical report). Cambridge University, Engineering Department. CUED/F-INFENG/TR.1.
44. Werbos, Paul J. (1988). "Generalization of backpropagation with application to a recurrent gas market model" (<https://zenodo.org/record/1258627>). *Neural Networks*. **1** (4): 339–356. doi:10.1016/0893-6080(88)90007-x (<https://doi.org/10.1016%2F0893-6080%2888%2990007-x>).
45. Feigenbaum, Edward (1988). *The Rise of the Expert Company* (<https://archive.org/details/riseofexpertco00feig>). Times Books. p. 317 (<https://archive.org/details/riseofexpertco00feig/page/317>). ISBN 978-0-8129-1731-4.
46. Sivic, Josef (April 2009). "Efficient visual search of videos cast as text retrieval" (<https://web.archive.org/web/20220331011840/https://www.di.ens.fr/~josef/publications/sivic09a.pdf>) (PDF). *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **31** (4): 591–605. CiteSeerX 10.1.1.174.6841 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.174.6841>). doi:10.1109/TPAMI.2008.111 (<https://doi.org/10.1109%2FTPAMI.2008.111>). PMID 19229077 (<https://pubmed.ncbi.nlm.nih.gov/19229077>). S2CID 9899337 (<https://api.semanticscholar.org/CorpusID:9899337>). Archived from the original (<https://di.ens.fr/~josef/publications/sivic09a.pdf>) (PDF) on 31 March 2022. Retrieved 5 July 2022.
47. McTear et al 2016, p. 167.

48. "Understanding the backward pass through Batch Normalization Layer" (<https://kratzert.github.io/2016/02/12/understanding-the-gradient-flow-through-the-batch-normalization-layer.html>). *kratzert.github.io*. Retrieved 24 April 2018.
49. Ioffe, Sergey; Szegedy, Christian (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". arXiv:1502.03167 (<https://arxiv.org/abs/1502.03167>) [cs.LG ([https://arxiv.org/archive/cs.LG](https://arxiv.org/archive/cs/LG))].
50. "Glossary of Deep Learning: Batch Normalisation" (<https://medium.com/deeper-learning/glossary-of-deep-learning-batch-normalisation-8266dcd2fa82>). *medium.com*. 27 June 2017. Retrieved 24 April 2018.
51. Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S and Zaidi M. The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.
52. Pham, D.T., Castellani, M. (2009), The Bees Algorithm – Modelling Foraging Behaviour to Solve Continuous Optimisation Problems (<https://pic.sagepub.com/content/223/12/2919.short>) Archived (<https://web.archive.org/web/20161109125453/https://pic.sagepub.com/content/223/12/2919.short>) 9 November 2016 at the Wayback Machine. Proc. ImechE, Part C, 223(12), 2919–2938.
53. Pham, D. T.; Castellani, M. (2014). "Benchmarking and comparison of nature-inspired population-based continuous optimisation algorithms". *Soft Computing*. **18** (5): 871–903. doi:10.1007/s00500-013-1104-9 (<https://doi.org/10.1007%2Fs00500-013-1104-9>). S2CID 35138140 (<https://api.semanticscholar.org/CorpusID:35138140>).
54. Pham, Duc Truong; Castellani, Marco (2015). "A comparative study of the Bees Algorithm as a tool for function optimisation" (<https://doi.org/10.1080%2F23311916.2015.1091540>). *Cogent Engineering*. **2**. doi:10.1080/23311916.2015.1091540 (<https://doi.org/10.1080%2F23311916.2015.1091540>).
55. Nasrinpour, H. R.; Massah Bavani, A.; Teshnehlab, M. (2017). "Grouped Bees Algorithm: A Grouped Version of the Bees Algorithm" (<https://doi.org/10.3390%2Fcomputers6010005>). *Computers*. **6** (1): 5. doi:10.3390/computers6010005 (<https://doi.org/10.3390%2Fcomputers6010005>).
56. Cao, Longbing (2010). "In-depth Behavior Understanding and Use: the Behavior Informatics Approach". *Information Science*. **180** (17): 3067–3085. arXiv:2007.15516 (<https://arxiv.org/abs/2007.15516>). doi:10.1016/j.ins.2010.03.025 (<https://doi.org/10.1016%2Fj.ins.2010.03.025>). S2CID 7400761 (<https://api.semanticscholar.org/CorpusID:7400761>).
57. Colledanchise Michele, and Ögren Petter 2016. How Behavior Trees Modularize Hybrid Control Systems and Generalize Sequential Behavior Compositions, the Subsumption Architecture, and Decision Trees. In IEEE Transactions on Robotics vol.PP, no.99, pp.1–18 (2016) (<https://michelecolledanchise.com/tro16colledanchise.pdf>)
58. Colledanchise, Michele; Ögren, Petter (2018). *Behavior Trees in Robotics and AI*. arXiv:1709.00084 (<https://arxiv.org/abs/1709.00084>). doi:10.1201/9780429489105 (<https://doi.org/10.1201%2F9780429489105>). ISBN 9780429950902. S2CID 27470659 (<https://api.semanticscholar.org/CorpusID:27470659>).
59. Breur, Tom (July 2016). "Statistical Power Analysis and the contemporary "crisis" in social sciences" (<https://doi.org/10.1057%2Fs41270-016-0001-3>). *Journal of Marketing Analytics*. **4** (2–3): 61–65. doi:10.1057/s41270-016-0001-3 (<https://doi.org/10.1057%2Fs41270-016-0001-3>). ISSN 2050-3318 (<https://search.worldcat.org/issn/2050-3318>).
60. Bachmann, Paul (1894). *Analytische Zahlentheorie* (<https://archive.org/stream/dieanalytischezahlentheorie/page/402/mode/2up>) [*Analytic Number Theory*] (in German). Vol. 2. Leipzig: Teubner.
61. Landau, Edmund (1909). *Handbuch der Lehre von der Verteilung der Primzahlen* (<https://archive.org/details/handbuchderlehre01landuoft>) [*Handbook on the theory of the distribution of the primes*] (in German). Leipzig: B. G. Teubner. p. 883.

62. John, Taylor (2009). Garnier, Rowan (ed.). *Discrete Mathematics: Proofs, Structures and Applications, Third Edition* (<https://books.google.com/books?id=WnkZSSc4lkoC&pg=PA620>). CRC Press. p. 620. ISBN 978-1-4398-1280-8.
63. Skiena, Steven S (2009). *The Algorithm Design Manual* (<https://books.google.com/books?id=7XUSn0IKQEGC&pg=PA77>). Springer Science & Business Media. p. 77. ISBN 978-1-84800-070-4.
64. Erman, L. D.; Hayes-Roth, F.; Lesser, V. R.; Reddy, D. R. (1980). "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty". *ACM Computing Surveys*. **12** (2): 213. doi:10.1145/356810.356816 (<https://doi.org/10.1145%2F356810.356816>). S2CID 118556 (<https://api.semanticscholar.org/CorpusID:118556>).
65. Corkill, Daniel D. (September 1991). "Blackboard Systems" (<https://web.archive.org/web/20120416034609/http://www.bbtech.com/papers/ai-expert.pdf>) (PDF). *AI Expert*. **6** (9): 40–47. Archived from the original (<https://bbtech.com/papers/ai-expert.pdf>) (PDF) on 16 April 2012. Retrieved 5 July 2022.
66. \* Nii, H. Yenny (1986). *Blackboard Systems* (<https://i.stanford.edu/pub/ctr/reports/cs/tr/86/1123/CS-TR-86-1123.pdf>) (PDF) (Technical report). Department of Computer Science, Stanford University. STAN-CS-86-1123. Retrieved 12 April 2013.
67. Hayes-Roth, B. (1985). "A blackboard architecture for control". *Artificial Intelligence*. **26** (3): 251–321. doi:10.1016/0004-3702(85)90063-3 (<https://doi.org/10.1016%2F0004-3702%2885%2990063-3>).
68. Hinton, Geoffrey E. (24 May 2007). "Boltzmann machine" (<https://doi.org/10.4249%2Fscholarpedia.1668>). *Scholarpedia*. **2** (5): 1668. Bibcode:2007SchpJ...2.1668H (<https://ui.adsabs.harvard.edu/abs/2007SchpJ...2.1668H>). doi:10.4249/scholarpedia.1668 (<https://doi.org/10.4249%2Fscholarpedia.1668>). ISSN 1941-6016 (<https://search.worldcat.org/issn/1941-6016>).
69. *NZZ- Die Zangengeburt eines möglichen Stammvaters* (<https://nzz.ch/aktuell/zuerich/uebersicht/die-zangengeburt-eines-designierten-stammvaters-1.18029566>). Website Neue Zürcher Zeitung. Seen 16. August 2013.
70. *Official Homepage Roboy* (<https://roboy.org/mediaundnews.html>) Archived (<https://web.archive.org/web/20130803052035/https://roboy.org/mediaundnews.html>) 2013-08-03 at the Wayback Machine. Website Roboy. Seen 16. August 2013.
71. *Official Homepage Starmind* (<https://starmind.com/>). Website Starmind. Seen 16. August 2013.
72. Sabour, Sara; Frosst, Nicholas; Hinton, Geoffrey E. (26 October 2017). "Dynamic Routing Between Capsules". arXiv:1710.09829 (<https://arxiv.org/abs/1710.09829>) [cs.CV (<https://arxiv.org/archive/cs.CV>)].
73. "What is a chatbot?" ([https://searchdomino.techtarget.com/sDefinition/0,,sid4\\_gci935566,00.html](https://searchdomino.techtarget.com/sDefinition/0,,sid4_gci935566,00.html)). *techtarget.com*. Retrieved 30 January 2017.
74. Civera, Javier; Ciocarlie, Matei; Aydemir, Alper; Bekris, Kostas; Sarma, Sanjay (2015). "Guest Editorial Special Issue on Cloud Robotics and Automation" (<https://doi.org/10.1109%2FTASE.2015.2409511>). *IEEE Transactions on Automation Science and Engineering*. **12** (2): 396–397. doi:10.1109/TASE.2015.2409511 (<https://doi.org/10.1109%2FTASE.2015.2409511>). S2CID 16080778 (<https://api.semanticscholar.org/CorpusID:16080778>).
75. "Robo Earth - Tech News" (<https://roboearth.org/>). *Robo Earth*.
76. Goldberg, Ken. "Cloud Robotics and Automation" (<https://goldberg.berkeley.edu/cloud-robotics>).
77. Li, R. "Cloud Robotics-Enable cloud computing for robots" (<https://sites.google.com/site/ruiji-aoli/blogs/page>). Retrieved 7 December 2014.
78. Fisher, Douglas (1987). "Knowledge acquisition via incremental conceptual clustering" (<https://doi.org/10.1007%2FBF00114265>). *Machine Learning*. **2** (2): 139–172. doi:10.1007/BF00114265 (<https://doi.org/10.1007%2FBF00114265>).

79. Fisher, Douglas H. (July 1987). "Improving inference through conceptual clustering". *Proceedings of the 1987 AAAI Conferences*. AAAI Conference. Seattle Washington. pp. 461–465.
80. Iba, William; Langley, Pat (27 January 2011). "Cobweb models of categorization and probabilistic concept formation". In Pothos, Emmanuel M.; Wills, Andy J. (eds.). *Formal approaches in categorization*. Cambridge: Cambridge University Press. pp. 253–273. ISBN 9780521190480.
81. Refer to the ICT website: <https://cogarch.ict.usc.edu/>
82. "Hewlett Packard Labs" (<https://web.archive.org/web/20161030143900/http://www.labs.hpe.com/research/next-next/brain/>). Archived from the original (<https://labs.hpe.com/research/next-next/brain/>) on 30 October 2016. Retrieved 5 July 2022.
83. Terdiman, Daniel (2014) .IBM's TrueNorth processor mimics the human brain.<https://cnet.com/news/ibms-truenorth-processor-mimics-the-human-brain/>
84. Knight, Shawn (2011). *IBM unveils cognitive computing chips that mimic human brain* (<http://techspot.com/news/45138-ibm-unveils-cognitive-computing-chips-that-mimic-human-brain.html>) TechSpot: August 18, 2011, 12:00 PM
85. Hamill, Jasper (2013). *Cognitive computing: IBM unveils software for its brain-like SyNAPSE chips* ([https://theregister.co.uk/2013/08/08/ibm\\_unveils\\_computer\\_architecture\\_based\\_upon\\_your\\_brain/](https://theregister.co.uk/2013/08/08/ibm_unveils_computer_architecture_based_upon_your_brain/)) The Register: August 8, 2013
86. Denning., P.J. (2014). "Surfing Toward the Future". *Communications of the ACM*. **57** (3): 26–29. doi:10.1145/2566967 (<https://doi.org/10.1145/2566967>). S2CID 20681733 (<https://api.semanticscholar.org/CorpusID:20681733>).
87. Ludwig, Lars (2013). *Extended Artificial Memory: Toward an integral cognitive theory of memory and technology* (<https://kluedo.ub.uni-kl.de/frontdoor/index/index/docId/3662>) (pdf) (Thesis). Technical University of Kaiserslautern. Retrieved 7 February 2017.
88. "Research at HP Labs" (<https://web.archive.org/web/20220307214522/https://www.hpl.hp.com/research/>). Archived from the original (<https://hpl.hp.com/research/>) on 7 March 2022. Retrieved 5 July 2022.
89. Cognitive science is an interdisciplinary field of researchers from Linguistics, psychology, neuroscience, philosophy, computer science, and anthropology that seek to understand the mind. How We Learn: Ask the Cognitive Scientist (<https://aft.org/newspubs/periodicals/ae/summer2002/willingham.cfm>)
90. Schrijver, Alexander (February 1, 2006). A Course in Combinatorial Optimization (PDF), page 1.
91. HAYKIN, S. Neural Networks – A Comprehensive Foundation. Second edition. Pearson Prentice Hall: 1999.
92. "PROGRAMS WITH COMMON SENSE" (<https://www-formal.stanford.edu/jmc/mcc59/mcc59.html>). *www-formal.stanford.edu*. Retrieved 11 April 2018.
93. Davis, Ernest; Marcus, Gary (2015). "Commonsense reasoning" (<https://cacm.acm.org/magazines/2015/9/191169-commonsense-reasoning-and-commonsense-knowledge-in-artificial-intelligence/fulltext>). *Communications of the ACM*. Vol. 58, no. 9. pp. 92–103. doi:10.1145/2701413 (<https://doi.org/10.1145/2701413>).
94. Hulstijn, J, and Nijholt, A. (eds.). Proceedings of the International Workshop on Computational Humor. Number 12 in Twente Workshops on Language Technology, Enschede, Netherlands. University of Twente, 1996.
95. "ACL – Association for Computational Learning" (<https://learningtheory.org/>).
96. Trappenberg, Thomas P. (2002). Fundamentals of Computational Neuroscience. United States: Oxford University Press Inc. p. 1. ISBN 978-0-19-851582-1.

97. What is computational neuroscience? Patricia S. Churchland, Christof Koch, Terrence J. Sejnowski. in *Computational Neuroscience* pp.46–55. Edited by Eric L. Schwartz. 1993. MIT Press "Computational Neuroscience Edited by Eric L. Schwartz" (<https://web.archive.org/web/20110604124206/https://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=7195>). Archived from the original (<https://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=7195>) on 4 June 2011. Retrieved 11 June 2009.
98. "Theoretical Neuroscience" (<https://web.archive.org/web/20180531150713/https://mitpress.mit.edu/books/theoretical-neuroscience>). *The MIT Press*. Archived from the original (<https://mitpress.mit.edu/books/theoretical-neuroscience>) on 31 May 2018. Retrieved 24 May 2018.
99. Gerstner, W.; Kistler, W.; Naud, R.; Paninski, L. (2014). *Neuronal Dynamics*. Cambridge, UK: Cambridge University Press. ISBN 9781107447615.
100. Kamentsky, L.A.; Liu, C.-N. (1963). "Computer-Automated Design of Multifont Print Recognition Logic" (<https://web.archive.org/web/20160303202147/http://domino.research.ibm.com/tchjr/journalindex.nsf/0/a5cb0910ea78194885256bfa00683e5a?OpenDocument>). *IBM Journal of Research and Development*. **7** (1): 2. doi:10.1147/rd.71.0002 (<https://doi.org/10.1147%2Frd.71.0002>). Archived from the original (<https://domino.research.ibm.com/tchjr/journalindex.nsf/0/a5cb0910ea78194885256bfa00683e5a?OpenDocument>) on 3 March 2016. Retrieved 5 July 2022.
101. Brncick, M (2000). "Computer automated design and computer automated manufacture". *Phys Med Rehabil Clin N Am*. **11** (3): 701–13. doi:10.1016/s1047-9651(18)30806-4 (<https://doi.org/10.1016%2Fs1047-9651%2818%2930806-4>). PMID 10989487 (<https://pubmed.ncbi.nlm.nih.gov/10989487>).
102. Li, Y.; et al. (2004). "CAutoCSD - Evolutionary search and optimisation enabled computer automated control system design" (<https://link.springer.com/article/10.1007%2Fs11633-004-0076-8>). *International Journal of Automation and Computing*. **1** (1): 76–88. doi:10.1007/s11633-004-0076-8 (<https://doi.org/10.1007%2Fs11633-004-0076-8>). S2CID 55417415 (<https://api.semanticscholar.org/CorpusID:55417415>).
103. Kramer, GJE; Grierson, DE (1989). "Computer automated design of structures under dynamic loads". *Computers & Structures*. **32** (2): 313–325. doi:10.1016/0045-7949(89)90043-6 (<https://doi.org/10.1016%2F0045-7949%2889%2990043-6>).
104. Moharrami, H; Grierson, DE (1993). "Computer-Automated Design of Reinforced Concrete Frameworks". *Journal of Structural Engineering*. **119** (7): 2036–2058. doi:10.1061/(asce)0733-9445(1993)119:7(2036) (<https://doi.org/10.1061%2F%28asce%290733-9445%281993%29119%3A7%282036%29>).
105. Xu, L; Grierson, DE (1993). "Computer-Automated Design of Semirigid Steel Frameworks". *Journal of Structural Engineering*. **119** (6): 1740–1760. doi:10.1061/(asce)0733-9445(1993)119:6(1740) (<https://doi.org/10.1061%2F%28asce%290733-9445%281993%29119%3A6%281740%29>).
106. Barsan, GM; Dinsoreanu, M, (1997). Computer-automated design based on structural performance criteria, Mouchel Centenary Conference on Innovation in Civil and Structural Engineering, Aug 19-21, Cambridge England, Innovation in Civil and Structural Engineering, 167–172
107. Li, Yun (1996). "Genetic algorithm automated approach to the design of sliding mode control systems". *International Journal of Control*. **63** (4): 721–739. doi:10.1080/00207179608921865 (<https://doi.org/10.1080%2F00207179608921865>).
108. Li, Yun; Chwee Kim, Ng; Chen Kay, Tan (1995). "Automation of Linear and Nonlinear Control Systems Design by Evolutionary Computation" (<https://sciencedirect.com/science/article/pii/S1474667017451585/pdf?md5=b7aedef998282848dfcf44a1ea2f003dd&pid=1-s2.0-S1474667017451585-main.pdf>) (PDF). *IFAC Proceedings Volumes*. **28** (16): 85–90. doi:10.1016/S1474-6670(17)45158-5 (<https://doi.org/10.1016%2FS1474-6670%2817%2945158-5>).



109. Barsan, GM, (1995) Computer-automated design of semirigid steel frameworks according to EUROCODE-3, Nordic Steel Construction Conference 95, JUN 19–21, 787–794
110. Gray, Gary J.; Murray-Smith, David J.; Li, Yun; et al. (1998). "Nonlinear model structure identification using genetic programming" (<https://sciencedirect.com/science/article/pii/S0967066198000872/pdf?md5=5ad89d3029a3ebad83086271f3c78f75&pid=1-s2.0-S0967066198000872-main.pdf>) (PDF). *Control Engineering Practice*. **6** (11): 1341–1352. doi:10.1016/s0967-0661(98)00087-2 (<https://doi.org/10.1016%2Fs0967-0661%2898%2900087-2>).
111. 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" (<https://ieeexplore.ieee.org/document/6052374>). *IEEE 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>).
112. Gregory S. Hornby (2003). Generative Representations for Computer-Automated Design Systems, NASA Ames Research Center, Mail Stop 269–3, Moffett Field, CA 94035-1000 ([https://ti.arc.nasa.gov/m/pub-archive/768h/0768%20\(Hornby\).pdf](https://ti.arc.nasa.gov/m/pub-archive/768h/0768%20(Hornby).pdf))
113. J. Clune and H. Lipson (2011). Evolving three-dimensional objects with a generative encoding inspired by developmental biology. Proceedings of the European Conference on Artificial Life. 2011. (<https://msu.edu/~jclune/webfiles/publications/2011-CluneLipson-Evolving3DObjectsWithCPPNs-ECAL.pdf>)
114. Zhan, Z.H.; et al. (2009). "Adaptive Particle Swarm Optimization" (<https://eprints.gla.ac.uk/7645/1/7645.pdf>) (PDF). *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*. **39** (6): 1362–1381. doi:10.1109/tsmcb.2009.2015956 (<https://doi.org/10.1109%2Ftsmcb.2009.2015956>). PMID 19362911 (<https://pubmed.ncbi.nlm.nih.gov/19362911>). S2CID 11191625 (<https://api.semanticscholar.org/CorpusID:11191625>).
115. "WordNet Search—3.1" (<https://web.archive.org/web/20130114223028/http://wordnetweb.princeton.edu/perl/webwn?s=computer%20scientist>). Wordnetweb.princeton.edu. Archived from the original (<https://wordnetweb.princeton.edu/perl/webwn?s=computer%20scientist>) on 14 January 2013. Retrieved 14 May 2012.
116. Dana H. Ballard; Christopher M. Brown (1982). Computer Vision. Prentice Hall. ISBN 0-13-165316-4.
117. Huang, T. (1996-11-19). Vandoni, Carlo, E, ed. Computer Vision : Evolution And Promise (PDF). 19th CERN School of Computing. Geneva: CERN. pp. 21–25. doi:10.5170/CERN-1996-008.21 (<https://doi.org/10.5170%2FCERN-1996-008.21>). ISBN 978-9290830955.
118. Milan Sonka; Vaclav Hlavac; Roger Boyle (2008). Image Processing, Analysis, and Machine Vision. Thomson. ISBN 0-495-08252-X.
119. Garson, James (27 November 2018). Zalta, Edward N. (ed.). *The Stanford Encyclopedia of Philosophy* (<https://plato.stanford.edu/archives/fall2018/entries/connectionism/>). Metaphysics Research Lab, Stanford University – via Stanford Encyclopedia of Philosophy.
120. "Ishtar for Belgium to Belgrade" ([https://eurovision.tv/page/news?id=554&t=ishtar\\_for\\_belgium\\_to\\_belgrade](https://eurovision.tv/page/news?id=554&t=ishtar_for_belgium_to_belgrade)). European Broadcasting Union. Retrieved 19 May 2013.
121. LeCun, Yann. "LeNet-5, convolutional neural networks" (<https://yann.lecun.com/exdb/lenet/>). Retrieved 16 November 2013.
122. Zhang, Wei (1988). "Shift-invariant pattern recognition neural network and its optical architecture". Proceedings of annual conference of the Japan Society of Applied Physics.
123. Zhang, Wei (1990). "Parallel distributed processing model with local space-invariant interconnections and its optical architecture". *Applied Optics*. **29** (32): 4790–7. Bibcode:1990ApOpt..29.4790Z (<https://ui.adsabs.harvard.edu/abs/1990ApOpt..29.4790Z>). doi:10.1364/AO.29.004790 (<https://doi.org/10.1364%2FAO.29.004790>). PMID 20577468 (<https://pubmed.ncbi.nlm.nih.gov/20577468>).

124. Tian, Yuandong; Zhu, Yan (2015). "Better Computer Go Player with Neural Network and Long-term Prediction". arXiv:1511.06410v1 (<https://arxiv.org/abs/1511.06410v1>) [cs.LG (<https://arxiv.org/archive/cs.LG>)].
125. "How Facebook's AI Researchers Built a Game-Changing Go Engine" (<https://technologyreview.com/s/544181/how-facebooks-ai-researchers-built-a-game-changing-go-engine/>). *MIT Technology Review*. 4 December 2015. Retrieved 3 February 2016.
126. "Facebook AI Go Player Gets Smarter With Neural Network And Long-Term Prediction To Master World's Hardest Game" (<https://techtimes.com/articles/128636/20160128/facebook-ai-go-player-gets-smarter-with-neural-network-and-long-term-prediction-to-master-worlds-hardest-game.htm>). *Tech Times*. 28 January 2016. Retrieved 24 April 2016.
127. "Facebook's artificially intelligent Go player is getting smarter" (<https://venturebeat.com/2016/01/26/facebooks-artificially-intelligent-go-player-is-getting-smarter/>). *VentureBeat*. 27 January 2016. Retrieved 24 April 2016.
128. Solomonoff, R.J. The Time Scale of Artificial Intelligence; Reflections on Social Effects, Human Systems Management, Vol 5 1985, Pp 149–153
129. Moor, J., The Dartmouth College Artificial Intelligence Conference: The Next Fifty years, AI Magazine, Vol 27, No., 4, Pp. 87–9, 2006
130. Kline, Ronald R., Cybernetics, Automata Studies and the Dartmouth Conference on Artificial Intelligence, IEEE Annals of the History of Computing, October–December, 2011, IEEE Computer Society
131. Haghighat, Mohammad; Abdel-Mottaleb, Mohamed; Alhalabi, Wadee (2016). "Discriminant Correlation Analysis: Real-Time Feature Level Fusion for Multimodal Biometric Recognition" (<https://zenodo.org/record/889881>). *IEEE Transactions on Information Forensics and Security*. **11** (9): 1984–1996. doi:10.1109/TIFS.2016.2569061 (<https://doi.org/10.1109%2FTIFS.2016.2569061>). S2CID 15624506 (<https://api.semanticscholar.org/CorpusID:15624506>).
132. Lenzerini, Maurizio (2002). "Data Integration: A Theoretical Perspective" (<https://web.archive.org/web/20211027091431/https://www.dis.uniroma1.it/~lenzerin/homepagine/talks/TutorialPODS02.pdf>) (PDF). *PODS 2002*. pp. 233–246. Archived from the original (<https://dis.uniroma1.it/~lenzerin/homepagine/talks/TutorialPODS02.pdf>) (PDF) on 27 October 2021. Retrieved 5 July 2022.
133. Lane, Frederick (2006). "IDC: World Created 161 Billion Gigs of Data in 2006" ([https://toptechnews.com/story.xhtml?story\\_id=01300000E3D0&full\\_skip=1](https://toptechnews.com/story.xhtml?story_id=01300000E3D0&full_skip=1)).
134. Dhar, V. (2013). "Data science and prediction" (<https://cacm.acm.org/magazines/2013/12/169933-data-science-and-prediction/fulltext>). *Communications of the ACM*. **56** (12): 64–73. doi:10.1145/2500499 (<https://doi.org/10.1145%2F2500499>). S2CID 6107147 (<https://api.semanticscholar.org/CorpusID:6107147>).
135. Leek, Jeff (12 December 2013). "The key word in 'Data Science' is not Data, it is Science" (<https://web.archive.org/web/20140102194117/https://simplystatistics.org/2013/12/12/the-key-word-in-data-science-is-not-data-it-is-science/>). Simply Statistics. Archived from the original (<https://simplystatistics.org/2013/12/12/the-key-word-in-data-science-is-not-data-it-is-science/>) on 2 January 2014. Retrieved 11 November 2018.
136. Hayashi, Chikio (1 January 1998). "What is Data Science ? Fundamental Concepts and a Heuristic Example" (<https://springer.com/book/9784431702085>). In Hayashi, Chikio; Yajima, Keiji; Bock, Hans-Hermann; Ohsumi, Noboru; Tanaka, Yutaka; Baba, Yasumasa (eds.). *Data Science, Classification, and Related Methods*. Studies in Classification, Data Analysis, and Knowledge Organization. Springer Japan. pp. 40–51. doi:10.1007/978-4-431-65950-1\_3 ([https://doi.org/10.1007%2F978-4-431-65950-1\\_3](https://doi.org/10.1007%2F978-4-431-65950-1_3)). ISBN 9784431702085.

137. Dedić, Nedim; Stanier, Clare (2016). Hammoudi, Slimane; Maciaszek, Leszek; Missikoff, Michele M. Missikoff; Camp, Olivier; Cordeiro, José (eds.). *An Evaluation of the Challenges of Multilingualism in Data Warehouse Development* (<https://eprints.staffs.ac.uk/2770/>). International Conference on Enterprise Information Systems, 25–28 April 2016, Rome, Italy ([https://eprints.staffs.ac.uk/2770/1/ICEIS\\_2016\\_Volume\\_1.pdf](https://eprints.staffs.ac.uk/2770/1/ICEIS_2016_Volume_1.pdf)) (PDF). *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016)*. Vol. 1. SciTePress. pp. 196–206. doi:10.5220/0005858401960206 (<https://doi.org/10.5220%2F0005858401960206>). ISBN 978-989-758-187-8.
138. "9 Reasons Data Warehouse Projects Fail" (<https://blog.rjmetrics.com/2014/12/04/10-common-mistakes-when-building-a-data-warehouse/>). *blog.rjmetrics.com*. 4 December 2014. Retrieved 30 April 2017.
139. Huang; Green; Loo, "Datalog and Emerging applications", *SIGMOD 2011* (<https://web.archive.org/web/20220701172125/https://www.cs.ucdavis.edu/~green/papers/sigmod906t-huang.pdf>) (PDF), UC Davis, archived from the original (<https://cs.ucdavis.edu/~green/papers/sigmod906t-huang.pdf>) (PDF) on 1 July 2022, retrieved 5 July 2022.
140. Steele, Katie and Stefánsson, H. Orri, "Decision Theory", *The Stanford Encyclopedia of Philosophy* (Winter 2015 Edition), Edward N. Zalta (ed.), URL = [1] (<https://plato.stanford.edu/archives/win2015/entries/decision-theory>)
141. Lloyd, J.W., *Practical Advantages of Declarative Programming*
142. "About Us | DeepMind" (<https://deepmind.com/about/>). *DeepMind*. 17 December 2024.
143. "A return to Paris | DeepMind" (<https://deepmind.com/blog/a-return-to-paris/>). *DeepMind*. 29 March 2018.
144. "The Last AI Breakthrough DeepMind Made Before Google Bought It" (<https://medium.com/the-physics-arxiv-blog/the-last-ai-breakthrough-deepmind-made-before-google-bought-it-for-400m-7952031ee5e1>). *The Physics arXiv Blog*. 29 January 2014. Retrieved 12 October 2014.
145. Graves, Alex; Wayne, Greg; Danihelka, Ivo (2014). "Neural Turing Machines". arXiv:1410.5401 (<https://arxiv.org/abs/1410.5401>) [cs.NE (<https://arxiv.org/archive/cs/NE>)].
146. Best of 2014: Google's Secretive DeepMind Startup Unveils a "Neural Turing Machine" (<https://technologyreview.com/view/533741/best-of-2014-googles-secretive-deepmind-startup-unveils-a-neural-turing-machine/>) Archived (<https://web.archive.org/web/20151204081728/http://www.technologyreview.com/view/533741/best-of-2014-googles-secretive-deepmind-startup-unveils-a-neural-turing-machine/>) 4 December 2015 at the *Wayback Machine*, *MIT Technology Review*
147. Graves, Alex; Wayne, Greg; Reynolds, Malcolm; Harley, Tim; Danihelka, Ivo; Grabska-Barwińska, Agnieszka; Colmenarejo, Sergio Gómez; Grefenstette, Edward; Ramalho, Tiago (12 October 2016). "Hybrid computing using a neural network with dynamic external memory" (<https://ora.ox.ac.uk/objects/uuid:dd8473bd-2d70-424d-881b-86d9c9c66b51>). *Nature*. **538** (7626): 471–476. Bibcode:2016Natur.538..471G (<https://ui.adsabs.harvard.edu/abs/2016Natur.538..471G>). doi:10.1038/nature20101 (<https://doi.org/10.1038%2Fnature20101>). ISSN 1476-4687 (<https://search.worldcat.org/issn/1476-4687>). PMID 27732574 (<https://pubmed.ncbi.nlm.nih.gov/27732574>). S2CID 205251479 (<https://api.semanticscholar.org/CorpusID:205251479>).
148. Kohs, Greg (29 September 2017), *AlphaGo* (<https://imdb.com/title/tt6700846/>), Ioannis Antonoglou, Lucas Baker, Nick Bostrom, retrieved 9 January 2018
149. Silver, David; Hubert, Thomas; Schrittwieser, Julian; Antonoglou, Ioannis; Lai, Matthew; Guez, Arthur; Lanctot, Marc; Sifre, Laurent; Kumaran, Dharmashan; Graepel, Thore; Lillicrap, Timothy; Simonyan, Karen; Hassabis, Demis (5 December 2017). "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". arXiv:1712.01815 (<https://arxiv.org/abs/1712.01815>) [cs.AI (<https://arxiv.org/archive/cs/AI>)].

150. Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M. (eds.). *A density-based algorithm for discovering clusters in large spatial databases with noise* (<https://cdn.aaai.org/KDD/1996/KDD96-037.pdf>) (PDF). Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231. CiteSeerX 10.1.1.121.9220 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.121.9220>). ISBN 1-57735-004-9.
151. Sikos, Leslie F. (2017). *Description Logics in Multimedia Reasoning* (<https://springer.com/us/book/9783319540658>). Cham: Springer International Publishing. doi:10.1007/978-3-319-54066-5 (<https://doi.org/10.1007%2F978-3-319-54066-5>). ISBN 978-3-319-54066-5. S2CID 3180114 (<https://api.semanticscholar.org/CorpusID:3180114>).
152. Ho, Jonathan; Jain, Ajay; Abbeel, Pieter (19 June 2020). *Denoising Diffusion Probabilistic Models*. arXiv:2006.11239 (<https://arxiv.org/abs/2006.11239>).
153. Song, Yang; Sohl-Dickstein, Jascha; Kingma, Diederik P.; Kumar, Abhishek; Ermon, Stefano; Poole, Ben (10 February 2021). "Score-Based Generative Modeling through Stochastic Differential Equations". arXiv:2011.13456 (<https://arxiv.org/abs/2011.13456>) [cs.LG (<https://arxiv.org/archive/cs.LG>)].
154. Gu, Shuyang; Chen, Dong; Bao, Jianmin; Wen, Fang; Zhang, Bo; Chen, Dongdong; Yuan, Lu; Guo, Baining (2021). "Vector Quantized Diffusion Model for Text-to-Image Synthesis". arXiv:2111.14822 (<https://arxiv.org/abs/2111.14822>) [cs.CV (<https://arxiv.org/archive/cs.CV>)].
155. Chang, Ziyi; Koulteris, George Alex; Shum, Hubert P. H. (2023). "On the Design Fundamentals of Diffusion Models: A Survey". arXiv:2306.04542 (<https://arxiv.org/abs/2306.04542>) [cs.LG (<https://arxiv.org/archive/cs.LG>)].
156. Croitoru, Florinel-Alin; Hondru, Vlad; Ionescu, Radu Tudor; Shah, Mubarak (2023). "Diffusion Models in Vision: A Survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **45** (9): 10850–10869. arXiv:2209.04747 (<https://arxiv.org/abs/2209.04747>). doi:10.1109/TPAMI.2023.3261988 (<https://doi.org/10.1109%2FTPAMI.2023.3261988>). PMID 37030794 (<https://pubmed.ncbi.nlm.nih.gov/37030794>). S2CID 252199918 (<https://api.semanticscholar.org/CorpusID:252199918>).
157. Roweis, S. T.; Saul, L. K. (2000). "Nonlinear Dimensionality Reduction by Locally Linear Embedding". *Science*. **290** (5500): 2323–2326. Bibcode:2000Sci...290.2323R (<https://ui.adsabs.harvard.edu/abs/2000Sci...290.2323R>). CiteSeerX 10.1.1.111.3313 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.111.3313>). doi:10.1126/science.290.5500.2323 (<https://doi.org/10.1126%2Fscience.290.5500.2323>). PMID 11125150 (<https://pubmed.ncbi.nlm.nih.gov/11125150>). S2CID 5987139 (<https://api.semanticscholar.org/CorpusID:5987139>).
158. Pudil, P.; Novovičová, J. (1998). "Novel Methods for Feature Subset Selection with Respect to Problem Knowledge". In Liu, Huan; Motoda, Hiroshi (eds.). *Feature Extraction, Construction and Selection* (<https://archive.org/details/featureextractio00liuh>). pp. 101 (<https://archive.org/details/featureextractio00liuh/page/n120>). doi:10.1007/978-1-4615-5725-8\_7 ([https://doi.org/10.1007%2F978-1-4615-5725-8\\_7](https://doi.org/10.1007%2F978-1-4615-5725-8_7)). ISBN 978-1-4613-7622-4.
159. Demazeau, Yves, and J-P. Müller, eds. *Decentralized Ai*. Vol. 2. Elsevier, 1990.
160. "Deep Double Descent" (<https://openai.com/blog/deep-double-descent/>). *OpenAI*. 5 December 2019. Retrieved 12 August 2022.
161. Schaeffer, Rylan; Khona, Mikail; Robertson, Zachary; Boopathy, Akhilan; Pistunova, Kateryna; Rocks, Jason W.; Fiete, Ila Rani; Koyejo, Oluwasanmi (24 March 2023). "Double Descent Demystified: Identifying, Interpreting & Ablating the Sources of a Deep Learning Puzzle". arXiv:2303.14151v1 (<https://arxiv.org/abs/2303.14151v1>) [cs.LG (<https://arxiv.org/archive/cs.LG>)].
162. Hendrickx, Iris; Van den Bosch, Antal (October 2005). "Hybrid algorithms with Instance-Based Classification" (<https://books.google.com/books?id=GtcevX7n90wC&pg=PA158>). *Machine Learning: ECML2005*. Springer. pp. 158–169. ISBN 9783540292432.

163. Ostrow, Adam (5 March 2011). "Roger Ebert's Inspiring Digital Transformation" (<https://mashable.com/2011/03/05/roger-ebert-ted-talk/>). Mashable Entertainment. Retrieved 12 September 2011. "With the help of his wife, two colleagues and the Alex-equipped MacBook that he uses to generate his computerized voice, famed film critic Roger Ebert delivered the final talk at the TED conference on Friday in Long Beach, California...."
164. Lee, Jennifer (7 March 2011). "Roger Ebert Tests His Vocal Cords, and Comedic Delivery" (<https://bits.blogs.nytimes.com/2011/03/07/roger-ebert-tests-his-vocal-cords-and-comedic-delivery/?src=me>). *The New York Times*. Retrieved 12 September 2011. "Now perhaps, there is the Ebert Test, a way to see if a synthesized voice can deliver humor with the timing to make an audience laugh.... He proposed the Ebert Test as a way to gauge the humanness of a synthesized voice."
165. "Roger Ebert's Inspiring Digital Transformation" (<https://web.archive.org/web/20110325160035/https://tips-tricks.co.in/2011/03/roger-eberts-inspiring-digital.html>). Tech News. 5 March 2011. Archived from the original (<https://tips-tricks.co.in/2011/03/roger-eberts-inspiring-digital.html>) on 25 March 2011. Retrieved 12 September 2011. "Meanwhile, the technology that enables Ebert to "speak" continues to see improvements – for example, adding more realistic inflection for question marks and exclamation points. In a test of that, which Ebert called the "Ebert test" for computerized voices,"
166. Pasternack, Alex (18 April 2011). "A MacBook May Have Given Roger Ebert His Voice, But An iPod Saved His Life (Video)" (<https://web.archive.org/web/20110906063605/https://motherboard.tv/2011/4/18/a-macbook-may-have-given-roger-ebert-his-voice-but-an-ipod-saved-his-life-video>). Motherboard. Archived from the original (<https://motherboard.tv/2011/4/18/a-macbook-may-have-given-roger-ebert-his-voice-but-an-ipod-saved-his-life-video>) on 6 September 2011. Retrieved 12 September 2011. "He calls it the "Ebert Test," after Turing's AI standard..."
167. Jaeger, Herbert; Haas, Harald (2004). "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication" ([https://web.archive.org/web/20220901054532/http://www.columbia.edu/cu/biology/courses/w4070/Reading\\_List\\_Yuste/haas\\_04.pdf](https://web.archive.org/web/20220901054532/http://www.columbia.edu/cu/biology/courses/w4070/Reading_List_Yuste/haas_04.pdf)) (PDF). *Science*. **304** (5667): 78–80. Bibcode:2004Sci...304...78J (<https://ui.adsabs.harvard.edu/abs/2004Sci...304...78J>). doi:10.1126/science.1091277 (<https://doi.org/10.1126%2Fscience.1091277>). PMID 15064413 (<https://pubmed.ncbi.nlm.nih.gov/15064413>). S2CID 2184251 (<https://api.semanticscholar.org/CorpusID:2184251>). Archived from the original ([https://columbia.edu/cu/biology/courses/w4070/Reading\\_List\\_Yuste/haas\\_04.pdf](https://columbia.edu/cu/biology/courses/w4070/Reading_List_Yuste/haas_04.pdf)) (PDF) on 1 September 2022. Retrieved 5 July 2022.
168. Herbert Jaeger (2007) Echo State Network. ([https://scholarpedia.org/article/Echo\\_State\\_Network](https://scholarpedia.org/article/Echo_State_Network)) Archived ([https://web.archive.org/web/20220628153329/http://www.scholarpedia.org/article/Echo\\_state\\_network](https://web.archive.org/web/20220628153329/http://www.scholarpedia.org/article/Echo_state_network)) 28 June 2022 at the Wayback Machine Scholarpedia.
169. Serenko, Alexander; Bontis, Nick; Detlor, Brian (2007). "End-user adoption of animated interface agents in everyday work applications" ([https://aserenko.com/papers/Serenko\\_Bontis\\_Detlor\\_end\\_user\\_adoption\\_agent.pdf](https://aserenko.com/papers/Serenko_Bontis_Detlor_end_user_adoption_agent.pdf)) (PDF). *Behaviour and Information Technology*. **26** (2): 119–132. doi:10.1080/01449290500260538 (<https://doi.org/10.1080%2F01449290500260538>). S2CID 2175427 (<https://api.semanticscholar.org/CorpusID:2175427>).
170. Opitz, D.; Maclin, R. (1999). "Popular ensemble methods: An empirical study" (<https://doi.org/10.1613%2Fjair.614>). *Journal of Artificial Intelligence Research*. **11**: 169–198. arXiv:1106.0257 (<https://arxiv.org/abs/1106.0257>). doi:10.1613/jair.614 (<https://doi.org/10.1613%2Fjair.614>).
171. Polikar, R. (2006). "Ensemble based systems in decision making". *IEEE Circuits and Systems Magazine*. **6** (3): 21–45. doi:10.1109/MCAS.2006.1688199 (<https://doi.org/10.1109%2FMCAS.2006.1688199>). S2CID 18032543 (<https://api.semanticscholar.org/CorpusID:18032543>).

172. Rokach, L. (2010). "Ensemble-based classifiers". *Artificial Intelligence Review*. **33** (1–2): 1–39. doi:10.1007/s10462-009-9124-7 (<https://doi.org/10.1007/s10462-009-9124-7>). hdl:11323/1748 (<https://hdl.handle.net/11323/1748>). S2CID 11149239 (<https://api.semanticscholar.org/CorpusID:11149239>).
173. Vikhar, P. A. (2016). "Evolutionary algorithms: A critical review and its future prospects". *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*. Jalgaon, 2016, pp. 261–265. pp. 261–265. doi:10.1109/ICGTSPICC.2016.7955308 (<https://doi.org/10.1109/ICGTSPICC.2016.7955308>). ISBN 978-1-5090-0467-6. S2CID 22100336 (<https://api.semanticscholar.org/CorpusID:22100336>).
174. Russell, Stuart; Norvig, Peter (2009). "26.3: The Ethics and Risks of Developing Artificial Intelligence". *Artificial Intelligence: A Modern Approach*. Prentice Hall. ISBN 978-0-13-604259-4.
175. Bostrom, Nick (2002). "Existential risks". *Journal of Evolution and Technology*. **9** (1): 1–31.
176. "Your Artificial Intelligence Cheat Sheet" ([https://slate.com/articles/technology/future\\_tense/2016/04/killer\\_ai\\_101\\_a\\_cheat\\_sheet\\_to\\_the\\_terminology\\_the\\_ethical\\_debates\\_the.html](https://slate.com/articles/technology/future_tense/2016/04/killer_ai_101_a_cheat_sheet_to_the_terminology_the_ethical_debates_the.html)). *Slate*. 1 April 2016. Retrieved 16 May 2016.
177. Jackson, Peter (1998), *Introduction To Expert Systems* (3 ed.), Addison Wesley, p. 2, ISBN 978-0-201-87686-4
178. "Conventional programming" ([https://web.archive.org/web/20121014124656/https://pcmag.com/encyclopedia\\_term/0%2C2542%2Ct%3Dconventional+programming%26i%3D40325%2C00.asp](https://web.archive.org/web/20121014124656/https://pcmag.com/encyclopedia_term/0%2C2542%2Ct%3Dconventional+programming%26i%3D40325%2C00.asp)). *PC Magazine*. Archived from the original ([https://pcmag.com/encyclopedia\\_term/0,2542,t=conventional+programming&i=40325,00.asp](https://pcmag.com/encyclopedia_term/0,2542,t=conventional+programming&i=40325,00.asp)) on 14 October 2012. Retrieved 15 September 2013.
179. Martignon, Laura; Vitouch, Oliver; Takezawa, Masanori; Forster, Malcolm. "Naive and Yet Enlightened: From Natural Frequencies to Fast and Frugal Decision Trees" ([https://researchgate.net/publication/27278577\\_Naive\\_and\\_Yet\\_Enlightened\\_From\\_Natural\\_Frequencies\\_to\\_Fast\\_and\\_Frugal\\_Decision\\_Trees](https://researchgate.net/publication/27278577_Naive_and_Yet_Enlightened_From_Natural_Frequencies_to_Fast_and_Frugal_Decision_Trees)), published in *Thinking : Psychological perspectives on reasoning, judgement and decision making* (David Hardman and Laura Macchi; editors), Chichester: John Wiley & Sons, 2003.
180. Bishop, Christopher (2006). *Pattern recognition and machine learning*. Berlin: Springer. ISBN 0-387-31073-8.
181. Bengio, Y.; Courville, A.; Vincent, P. (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/tpami.2013.50>). PMID 23787338 (<https://pubmed.ncbi.nlm.nih.gov/23787338>). S2CID 393948 (<https://api.semanticscholar.org/CorpusID:393948>).
182. Hodgson, Dr. J. P. E., "First Order Logic" (<https://people.sju.edu/~jhodgson/ugai/1order.html>) Archived (<https://web.archive.org/web/20190921071136/http://people.sju.edu/~jhodgson/ugai/1order.html>) 21 September 2019 at the Wayback Machine, Saint Joseph's University, Philadelphia, 1995.
183. Hughes, G. E., & Cresswell, M. J., *A New Introduction to Modal Logic* (London: Routledge, 1996), p.161 ([https://books.google.com/books?id=\\_CB5wiBeaA4C&pg=PA161](https://books.google.com/books?id=_CB5wiBeaA4C&pg=PA161)).
184. Feigenbaum, Edward (1988). *The Rise of the Expert Company* (<https://archive.org/details/riseofexpertco00feig/page/318>). Times Books. p. 318 (<https://archive.org/details/riseofexpertco00feig/page/318>). ISBN 978-0-8129-1731-4.



185. Hayes, Patrick (1981). "The Frame Problem and Related Problems in Artificial Intelligence" (<https://web.archive.org/web/20131203002046/https://aitopics.org/sites/default/files/classic/Webber-Nilsson-Readings/Rdgs-NW-Hayes-FrameProblem.pdf>) (PDF). *Readings in Artificial Intelligence*. University of Edinburgh: 223–230. doi:10.1016/B978-0-934613-03-3.50020-9 (<https://doi.org/10.1016%2FB978-0-934613-03-3.50020-9>). ISBN 9780934613033. S2CID 141711662 (<https://api.semanticscholar.org/CorpusID:141711662>). Archived from the original (<https://aitopics.org/sites/default/files/classic/Webber-Nilsson-Readings/Rdgs-NW-Hayes-FrameProblem.pdf>) (PDF) on 3 December 2013. Retrieved 9 March 2019.
186. Sardar, Z (2010). "The Namesake: Futures; futures studies; futurology; futuristic; Foresight – What's in a name?". *Futures*. **42** (3): 177–184. doi:10.1016/j.futures.2009.11.001 (<https://doi.org/10.1016%2Fj.futures.2009.11.001>).
187. Pedrycz, Witold (1993). *Fuzzy control and fuzzy systems* (2 ed.). Research Studies Press Ltd.
188. Hájek, Petr (1998). *Metamathematics of fuzzy logic* (4 ed.). Springer Science & Business Media.
189. D. Dubois and H. Prade (1988) *Fuzzy Sets and Systems*. Academic Press, New York.
190. Liang, Lily R.; Lu, Shiyong; Wang, Xuena; Lu, Yi; Mandal, Vinay; Patacsil, Dorrelyn; Kumar, Deepak (2006). "FM-test: A fuzzy-set-theory-based approach to differential gene expression data analysis" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1780132>). *BMC Bioinformatics*. **7** (Suppl 4): S7. doi:10.1186/1471-2105-7-S4-S7 (<https://doi.org/10.1186%2F1471-2105-7-S4-S7>). PMC 1780132 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1780132>). PMID 17217525 (<https://pubmed.ncbi.nlm.nih.gov/17217525>).
191. Myerson, Roger B. (1991). *Game Theory: Analysis of Conflict*, Harvard University Press, p. 1 (<https://books.google.com/books?id=E8WQFRCsNr0C&pg=PA1>). Chapter-preview links, pp. vii–xi (<https://books.google.com/books?id=E8WQFRCsNr0C&pg=PR7>).
192. Pell, Barney (1992). H. van den Herik; L. Allis (eds.). "Metagame: a new challenge for games and learning" (<https://web.archive.org/web/20200217154408/https://svn.sable.mcgill.ca/sable/courses/COMP763/oldpapers/pell-92-metagame.pdf>) [Heuristic programming in artificial intelligence 3—the third computerolympiad] (PDF). Ellis-Horwood. Archived from the original (<https://svn.sable.mcgill.ca/sable/courses/COMP763/oldpapers/pell-92-metagame.pdf>) (PDF) on 17 February 2020. Retrieved 13 June 2020.
193. Pell, Barney (1996). "A Strategic Metagame Player for General Chess-Like Games". *Computational Intelligence*. **12** (1): 177–198. doi:10.1111/j.1467-8640.1996.tb00258.x (<https://doi.org/10.1111%2Fj.1467-8640.1996.tb00258.x>). ISSN 1467-8640 (<https://search.worldcat.org/issn/1467-8640>). S2CID 996006 (<https://api.semanticscholar.org/CorpusID:996006>).
194. Genesereth, Michael; Love, Nathaniel; Pell, Barney (15 June 2005). "General Game Playing: Overview of the AAAI Competition". *AI Magazine*. **26** (2): 62. doi:10.1609/aimag.v26i2.1813 (<https://doi.org/10.1609%2Faimag.v26i2.1813>). ISSN 2371-9621 (<https://search.worldcat.org/issn/2371-9621>).
195. Gluck, Mark A.; Mercado, Eduardo; Myers, Catherine E. (2011). *Learning and memory: from brain to behavior* (2nd ed.). New York: Worth Publishers. p. 209. ISBN 9781429240147.
196. Mohri, M., Rostamizadeh A., Talwaker A., (2018) *Foundations of Machine learning*, 2nd ed., Boston: MIT Press
197. Y S. Abu-Mostafa, M.Magdon-Ismael, and H.-T. Lin (2012) *Learning from Data*, AMLBook Press. ISBN 978-1600490064
198. Griffith, Erin; Metz, Cade (27 January 2023). "Anthropic Said to Be Closing In on \$300 Million in New A.I. Funding" (<https://www.nytimes.com/2023/01/27/technology/anthropic-ai-funding.html>). *The New York Times*. Retrieved 14 March 2023.

199. Lanxon, Nate; Bass, Dina; Davalos, Jackie (10 March 2023). "A Cheat Sheet to AI Buzzwords and Their Meanings" (<https://news.bloomberglaw.com/tech-and-telecom-law/a-cheat-sheet-to-ai-buzzwords-and-their-meanings-quicktake>). *Bloomberg News*. Retrieved 14 March 2023.
200. Pasick, Adam (27 March 2023). "Artificial Intelligence Glossary: Neural Networks and Other Terms Explained" (<https://www.nytimes.com/article/ai-artificial-intelligence-glossary.html>). *The New York Times*. ISSN 0362-4331 (<https://search.worldcat.org/issn/0362-4331>). Retrieved 22 April 2023.
201. Andrej Karpathy; Pieter Abbeel; Greg Brockman; Peter Chen; Vicki Cheung; Yan Duan; Ian Goodfellow; Durk Kingma; Jonathan Ho; Rein Houthooft; Tim Salimans; John Schulman; Ilya Sutskever; Wojciech Zaremba (16 June 2016). "Generative models" (<https://openai.com/research/generative-models>). *OpenAI*.
202. Smith, Craig S. (15 March 2023). "ChatGPT-4 Creator Ilya Sutskever on AI Hallucinations and AI Democracy" (<https://www.forbes.com/sites/craigsmith/2023/03/15/gpt-4-creator-ilya-sutskever-on-ai-hallucinations-and-ai-democracy/>). *Forbes*. Retrieved 25 December 2023.
203. Mitchell 1996, p. 2.
204. Trudeau, Richard J. (1993). *Introduction to Graph Theory* (<https://store.doverpublications.com/0486678709.html>) (Corrected, enlarged republication. ed.). New York: Dover Pub. p. 19. ISBN 978-0-486-67870-2. Retrieved 8 August 2012. "A graph is an object consisting of two sets called its *vertex set* and its *edge set*."
205. Yoon, Byoung-Ha; Kim, Seon-Kyu; Kim, Seon-Young (March 2017). "Use of Graph Database for the Integration of Heterogeneous Biological Data" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5389944>). *Genomics & Informatics*. **15** (1): 19–27. doi:10.5808/GI.2017.15.1.19 (<https://doi.org/10.5808%2FGI.2017.15.1.19>). ISSN 1598-866X (<https://search.worldcat.org/issn/1598-866X>). PMC 5389944 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5389944>). PMID 28416946 (<https://pubmed.ncbi.nlm.nih.gov/28416946>).
206. Bourbakis, Nikolaos G. (1998). *Artificial Intelligence and Automation* (<https://books.google.com/books?id=mV3wxKLHlnwC&q=%22gdb%22+%22graph+database%22&pg=PA381>). World Scientific. p. 381. ISBN 9789810226374. Retrieved 20 April 2018.
207. Pearl, Judea (1984). *Heuristics: intelligent search strategies for computer problem solving* (<https://archive.org/details/intelligentsearc00jude>). United States: Addison-Wesley Pub. Co., Inc., Reading, MA. p. 3 (<https://archive.org/details/intelligentsearc00jude/page/n21>). Bibcode:1985hiss.book.....P (<https://ui.adsabs.harvard.edu/abs/1985hiss.book.....P>). OSTI 5127296 (<https://www.osti.gov/biblio/5127296>).
208. E. K. Burke, E. Hart, G. Kendall, J. Newall, P. Ross, and S. Schulenburg, Hyper-heuristics: An emerging direction in modern search technology, *Handbook of Metaheuristics* (F. Glover and G. Kochenberger, eds.), Kluwer, 2003, pp. 457–474.
209. P. Ross, Hyper-heuristics, *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (E. K. Burke and G. Kendall, eds.), Springer, 2005, pp. 529–556.
210. Ozcan, E.; Bilgin, B.; Korkmaz, E. E. (2008). "A Comprehensive Analysis of Hyper-heuristics". *Intelligent Data Analysis*. **12** (1): 3–23. doi:10.3233/ida-2008-12102 (<https://doi.org/10.3233%2Fida-2008-12102>).
211. "IEEE CIS Scope" (<https://web.archive.org/web/20160604143046/https://cis.ieee.org/scope.html>). Archived from the original (<https://cis.ieee.org/scope.html>) on 4 June 2016. Retrieved 18 March 2019.
212. "Control of Machining Processes – Purdue ME Manufacturing Laboratories" ([https://engineering.purdue.edu/ManLab/control/intell\\_control.htm](https://engineering.purdue.edu/ManLab/control/intell_control.htm)). *engineering.purdue.edu*.

213. Hoy, Matthew B. (2018). "Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants". *Medical Reference Services Quarterly*. **37** (1): 81–88. doi:10.1080/02763869.2018.1404391 (<https://doi.org/10.1080%2F02763869.2018.1404391>). PMID 29327988 (<https://pubmed.ncbi.nlm.nih.gov/29327988>). S2CID 30809087 (<https://api.semanticscholar.org/CorpusID:30809087>).
214. Oudeyer, Pierre-Yves; Kaplan, Frederic (2008). "How can we define intrinsic motivation?". *Proc. of the 8th Conf. on Epigenetic Robotics*. Vol. 5. pp. 29–31.
215. Chevallier, Arnaud (2016). "Strategic thinking in complex problem solving". *Oxford Scholarship Online*. Oxford; New York: Oxford University Press. doi:10.1093/acprof:oso/9780190463908.001.0001 (<https://doi.org/10.1093%2Facprof%3Aoso%2F9780190463908.001.0001>). ISBN 9780190463908. OCLC 940455195 (<https://search.worldcat.org/oclc/940455195>). S2CID 157255130 (<https://api.semanticscholar.org/CorpusID:157255130>).
216. "Strategy survival guide: Issue trees" ([https://web.archive.org/web/20120217163843/https://interactive.cabinetoffice.gov.uk/strategy/survivalguide/skills/s\\_issue.htm](https://web.archive.org/web/20120217163843/https://interactive.cabinetoffice.gov.uk/strategy/survivalguide/skills/s_issue.htm)). London: Government of the United Kingdom. July 2004. Archived from the original ([https://interactive.cabinetoffice.gov.uk/strategy/survivalguide/skills/s\\_issue.htm](https://interactive.cabinetoffice.gov.uk/strategy/survivalguide/skills/s_issue.htm)) on 17 February 2012. Retrieved 6 October 2018. Also available in PDF format ([http://webarchive.nationalarchives.gov.uk/20060213205515/https://strategy.gov.uk/downloads/survivalguide/downloads/ssg\\_v2.1.pdf](http://webarchive.nationalarchives.gov.uk/20060213205515/https://strategy.gov.uk/downloads/survivalguide/downloads/ssg_v2.1.pdf)).
217. Paskin, Mark. "A Short Course on Graphical Models" (<https://ai.stanford.edu/~paskin/gm-short-course/lec3.pdf>) (PDF). *Stanford*.
218. Woods, W. A.; Schmolze, J. G. (1992). "The KL-ONE family". *Computers & Mathematics with Applications*. **23** (2–5): 133. doi:10.1016/0898-1221(92)90139-9 (<https://doi.org/10.1016%2F0898-1221%2892%2990139-9>).
219. Brachman, R. J.; Schmolze, J. G. (1985). "An Overview of the KL-ONE Knowledge Representation System" (<https://dli.iit.ac.in/vdata/IJCAI/IJCAI-83-VOL-1/PDF/072.pdf>) (PDF). *Cognitive Science*. **9** (2): 171. doi:10.1207/s15516709cog0902\_1 ([https://doi.org/10.1207%2Fs15516709cog0902\\_1](https://doi.org/10.1207%2Fs15516709cog0902_1)).
220. Duce, D.A.; Ringland, G.A. (1988). *Approaches to Knowledge Representation, An Introduction* (<https://archive.org/details/approachestoknow0000unse>). Research Studies Press, Ltd. ISBN 978-0-86380-064-1.
221. Fix, Evelyn; Hodges, Joseph L. (1951). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties (<https://apps.dtic.mil/dtic/tr/fulltext/u2/a800276.pdf>) (PDF) (Report). USAF School of Aviation Medicine, Randolph Field, Texas. Archived (<https://web.archive.org/web/20200926212807/https://apps.dtic.mil/dtic/tr/fulltext/u2/a800276.pdf>) (PDF) from the original on 26 September 2020.
222. Schank, Roger; Robert Abelson (1977). *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures*. Lawrence Erlbaum Associates, Inc.
223. "Knowledge Representation in Neural Networks – deepMinds" (<https://web.archive.org/web/20180817023355/https://deepminds.science/knowledge-representation-neural-networks/>). *deepMinds*. 16 August 2018. Archived from the original (<https://deepminds.science/knowledge-representation-neural-networks/>) on 17 August 2018. Retrieved 16 August 2018.
224. Kerner, Sean Michael. "What are Large Language Models?" (<https://www.techtarget.com/whatis/definition/large-language-model-LLM>). *TechTarget*. Retrieved 28 January 2024.
225. Reilly, Edwin D. (2003). *Milestones in computer science and information technology* (<https://archive.org/details/milestonesincomp0000reil>). Greenwood Publishing Group. pp. 156 (<https://archive.org/details/milestonesincomp0000reil/page/156>)–157. ISBN 978-1-57356-521-9.
226. Hochreiter, Sepp; Schmidhuber, Jürgen (1997). "Long short-term memory". *Neural Computation*. **9** (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735 (<https://doi.org/10.1162%2Fneco.1997.9.8.1735>). PMID 9377276 (<https://pubmed.ncbi.nlm.nih.gov/9377276>). S2CID 1915014 (<https://api.semanticscholar.org/CorpusID:1915014>).

227. Siegelmann, Hava T.; Sontag, Eduardo D. (1992). "On the computational power of neural nets". *Proceedings of the fifth annual workshop on Computational learning theory*. Vol. COLT '92. pp. 440–449. doi:10.1145/130385.130432 (<https://doi.org/10.1145%2F130385.130432>). ISBN 978-0897914970. S2CID 207165680 (<https://api.semanticscholar.org/CorpusID:207165680>). {{cite book}}: |work= ignored (help)
228. "Markov chain | Definition of Markov chain in US English by Oxford Dictionaries" ([https://web.archive.org/web/20171215001435/https://en.oxforddictionaries.com/definition/us/markov\\_chain](https://web.archive.org/web/20171215001435/https://en.oxforddictionaries.com/definition/us/markov_chain)). *Oxford Dictionaries | English*. Archived from the original ([https://en.oxforddictionaries.com/definition/us/markov\\_chain](https://en.oxforddictionaries.com/definition/us/markov_chain)) on 15 December 2017. Retrieved 14 December 2017.
229. Definition at Brilliant.org "Brilliant Math and Science Wiki" (<https://brilliant.org/wiki/markov-chains/>). Retrieved 12 May 2019
230. "The Nature of Mathematical Programming (<https://glossary.computing.society.informs.org/index.php?page=nature.html>) Archived (<https://web.archive.org/web/20140305080324/https://glossary.computing.society.informs.org/index.php?page=nature.html>) 2014-03-05 at the Wayback Machine," *Mathematical Programming Glossary*, INFORMS Computing Society.
231. Wang, Wenwu (1 July 2010). *Machine Audition: Principles, Algorithms and Systems* (<https://igi-global.com/book/machine-audition-principles-algorithms-systems/40288>). IGI Global. ISBN 9781615209194 – via igi-global.com.
232. "Machine Audition: Principles, Algorithms and Systems" ([https://epubs.surrey.ac.uk/596085/1/Wang\\_Preface\\_MA\\_2010.pdf](https://epubs.surrey.ac.uk/596085/1/Wang_Preface_MA_2010.pdf)) (PDF).
233. Malcolm Tatum (October 3, 2012). "What is Machine Perception".
234. Alexander Serov (January 29, 2013). "Subjective Reality and Strong Artificial Intelligence" (PDF).
235. "Machine Perception & Cognitive Robotics Laboratory" (<https://ccs.fau.edu/~hahn/mpcr/>). *ccs.fau.edu*. Retrieved 18 June 2016.
236. "What is Mechatronics Engineering?" (<https://web.archive.org/web/20111006100431/https://mme.uwaterloo.ca/undergrad/mechatronics/prospective/prospective.html>). *Prospective Student Information*. University of Waterloo. Archived from the original (<https://mme.uwaterloo.ca/undergrad/mechatronics/prospective/prospective.html>) on 6 October 2011. Retrieved 30 May 2011.
237. "Mechatronics (Bc., Ing., PhD.)" (<https://web.archive.org/web/20160815193349/https://mechatronics.tul.cz/>). Archived from the original (<https://mechatronics.tul.cz/>) on 15 August 2016. Retrieved 15 April 2011.
238. Franke; Siezen, Teusink (2005). "Reconstructing the metabolic network of a bacterium from its genome". *Trends in Microbiology*. **13** (11): 550–558. doi:10.1016/j.tim.2005.09.001 (<https://doi.org/10.1016%2Fj.tim.2005.09.001>). PMID 16169729 (<https://pubmed.ncbi.nlm.nih.gov/16169729/>).
239. Balamurugan, R.; Natarajan, A.M.; Premalatha, K. (2015). "Stellar-Mass Black Hole Optimization for Biclustering Microarray Gene Expression Data" (<https://doi.org/10.1080%2F08839514.2015.1016391>). *Applied Artificial Intelligence*. **29** (4): 353–381. doi:10.1080/08839514.2015.1016391 (<https://doi.org/10.1080%2F08839514.2015.1016391>). S2CID 44624424 (<https://api.semanticscholar.org/CorpusID:44624424>).
240. Bianchi, Leonora; Dorigo, Marco; Maria Gambardella, Luca; Gutjahr, Walter J. (2009). "A survey on metaheuristics for stochastic combinatorial optimization" ([https://doc.rero.ch/record/319945/files/11047\\_2008\\_Article\\_9098.pdf](https://doc.rero.ch/record/319945/files/11047_2008_Article_9098.pdf)) (PDF). *Natural Computing*. **8** (2): 239–287. doi:10.1007/s11047-008-9098-4 (<https://doi.org/10.1007%2Fs11047-008-9098-4>). S2CID 9141490 (<https://api.semanticscholar.org/CorpusID:9141490>).
241. Herbert B. Enderton, 2001, A Mathematical Introduction to Logic Second Edition Enderton:110, Harcourt Academic Press, Burlington MA, ISBN 978-0-12-238452-3.
242. Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function *Mathematics of Control, Signals, and Systems*, 2(4), 303–314.

243. **"Naive Semantics to Support Automated Database Design (<https://portal.acm.org/citation.cfm?id=628188>)"**, *IEEE Transactions on Knowledge and Data Engineering*, Volume 14, issue 1 (January 2002) by V. C. Storey, R. C. Goldstein and H. Ullrich
244. *Using early binding and late binding in Automation* (<https://support.microsoft.com/kb/245115>), Microsoft, 11 May 2007, retrieved 11 May 2009
245. strictly speaking a URIRef
246. <https://w3.org/TR/PR-rdf-syntax/> "Resource Description Framework (RDF) Model and Syntax Specification"
247. Miller, Lance A. "Natural language programming: Styles, strategies, and contrasts." *IBM Systems Journal* 20.2 (1981): 184–215.
248. Hopfield, J. J. (1982). "Neural networks and physical systems with emergent collective computational abilities" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC346238>). *Proc. Natl. Acad. Sci. U.S.A.* **79** (8): 2554–2558. Bibcode:1982PNAS...79.2554H (<https://ui.adsabs.harvard.edu/abs/1982PNAS...79.2554H>). doi:10.1073/pnas.79.8.2554 (<https://doi.org/10.1073%2Fpnas.79.8.2554>). PMC 346238 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC346238>). PMID 6953413 (<https://pubmed.ncbi.nlm.nih.gov/6953413>).
249. "Deep Minds: An Interview with Google's Alex Graves & Koray Kavukcuoglu" (<https://linkedin.com/pulse/deep-minds-interview-googles-alex-graves-koray-sophie-curtis>). Retrieved 17 May 2016.
250. Graves, Alex; Wayne, Greg; Danihelka, Ivo (2014). "Neural Turing Machines". arXiv:1410.5401 (<https://arxiv.org/abs/1410.5401>) [cs.NE (<https://arxiv.org/archive/cs/NE>)].
251. Krucoff, Max O.; Rahimpour, Shervin; Slutzky, Marc W.; Edgerton, V. Reggie; Turner, Dennis A. (1 January 2016). "Enhancing Nervous System Recovery through Neurobiologics, Neural Interface Training, and Neurorehabilitation" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5186786>). *Frontiers in Neuroscience*. **10**: 584. doi:10.3389/fnins.2016.00584 (<https://doi.org/10.3389%2Ffnins.2016.00584>). PMC 5186786 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5186786>). PMID 28082858 (<https://pubmed.ncbi.nlm.nih.gov/28082858>).
252. Mead, Carver (1990). "Neuromorphic electronic systems" (<https://authors.library.caltech.edu/53090/1/00058356.pdf>) (PDF). *Proceedings of the IEEE*. **78** (10): 1629–1636. CiteSeerX 10.1.1.161.9762 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.161.9762>). doi:10.1109/5.58356 (<https://doi.org/10.1109%2F5.58356>). S2CID 1169506 (<https://api.semanticscholar.org/CorpusID:1169506>).
253. Maan, A. K.; Jayadevi, D. A.; James, A. P. (1 January 2016). "A Survey of Memristive Threshold Logic Circuits". *IEEE Transactions on Neural Networks and Learning Systems*. **PP** (99): 1734–1746. arXiv:1604.07121 (<https://arxiv.org/abs/1604.07121>). doi:10.1109/TNNLS.2016.2547842 (<https://doi.org/10.1109%2FTNNLS.2016.2547842>). ISSN 2162-237X (<https://search.worldcat.org/issn/2162-237X>). PMID 27164608 (<https://pubmed.ncbi.nlm.nih.gov/27164608>). S2CID 1798273 (<https://api.semanticscholar.org/CorpusID:1798273>).
254. "A Survey of Spintronic Architectures for Processing-in-Memory and Neural Networks ([http://academia.edu/37832670/A\\_Survey\\_of\\_Spintronic\\_Architectures\\_for\\_Processing-in-Memory\\_and\\_Neural\\_Networks](http://academia.edu/37832670/A_Survey_of_Spintronic_Architectures_for_Processing-in-Memory_and_Neural_Networks))", JSA, 2018
255. Zhou, You; Ramanathan, S. (1 August 2015). "Mott Memory and Neuromorphic Devices" (<https://zenodo.org/record/895565>). *Proceedings of the IEEE*. **103** (8): 1289–1310. doi:10.1109/JPROC.2015.2431914 (<https://doi.org/10.1109%2FJPROC.2015.2431914>). ISSN 0018-9219 (<https://search.worldcat.org/issn/0018-9219>). S2CID 11347598 (<https://api.semanticscholar.org/CorpusID:11347598>).
256. Monroe, D. (2014). "Neuromorphic computing gets ready for the (really) big time". *Communications of the ACM*. **57** (6): 13–15. doi:10.1145/2601069 (<https://doi.org/10.1145%2F2601069>). S2CID 20051102 (<https://api.semanticscholar.org/CorpusID:20051102>).

257. Zhao, W. S.; Agnus, G.; Derycke, V.; Filoramo, A.; Bourgoin, J. -P.; Gamrat, C. (2010). "Nanotube devices based crossbar architecture: Toward neuromorphic computing" (<https://web.archive.org/web/20210410124706/https://zenodo.org/record/3428659>). *Nanotechnology*. **21** (17): 175202. Bibcode:2010Nanot..21q5202Z (<https://ui.adsabs.harvard.edu/abs/2010Nanot..21q5202Z>). doi:10.1088/0957-4484/21/17/175202 (<https://doi.org/10.1088%2F0957-4484%2F21%2F17%2F175202>). PMID 20368686 (<https://pubmed.ncbi.nlm.nih.gov/20368686>). S2CID 16253700 (<https://api.semanticscholar.org/CorpusID:16253700>). Archived from the original (<https://zenodo.org/record/3428659>) on 10 April 2021. Retrieved 2 December 2019.
258. The Human Brain Project SP 9: Neuromorphic Computing Platform (<https://www.youtube.com/watch?v=6RoiZ90mGfw>) on YouTube
259. Copeland, Jack (May 2000). "What is Artificial Intelligence?" ([https://web.archive.org/web/20151109070037/http://www.alanturing.net/turing\\_archive/pages/reference%20articles/What\\_is\\_AI/What%20is%20AI11.html](https://web.archive.org/web/20151109070037/http://www.alanturing.net/turing_archive/pages/reference%20articles/What_is_AI/What%20is%20AI11.html)). *Alan Turing.net*. Archived from the original ([https://alanturing.net/turing\\_archive/pages/reference%20articles/what\\_is\\_AI/What%20is%20AI11.html](https://alanturing.net/turing_archive/pages/reference%20articles/what_is_AI/What%20is%20AI11.html)) on 9 November 2015. Retrieved 7 November 2015.
260. Kleinberg, Jon; Tardos, Éva (2006). *Algorithm Design* (<https://archive.org/details/algorithmdesign0000klei>) (2nd ed.). Addison-Wesley. p. 464 (<https://archive.org/details/algorithmdesign0000klei/page/464>). ISBN 0-321-37291-3.
261. Cobham, Alan (1965). "The intrinsic computational difficulty of functions". *Proc. Logic, Methodology, and Philosophy of Science II*. North Holland.
262. "What is Occam's Razor?" (<https://math.ucr.edu/home/baez/physics/General/occam.html>). *math.ucr.edu*. Retrieved 1 June 2019.
263. "OpenAI shifts from nonprofit to 'capped-profit' to attract capital". TechCrunch. Retrieved 2019-05-10.
264. "OpenCog: Open-Source Artificial General Intelligence for Virtual Worlds" (<https://web.archive.org/web/20090306053354/https://cybertechnews.org/?p=915>). *CyberTech News*. 6 March 2009. Archived from the original (<https://cybertechnews.org/?p=915>) on 6 March 2009. Retrieved 1 October 2016.
265. St. Laurent, Andrew M. (2008). *Understanding Open Source and Free Software Licensing* (<https://books.google.com/books?id=04jG7TTLujoC&pg=PA4>). O'Reilly Media. p. 4. ISBN 9780596553951.
266. Levine, Sheen S.; Prietula, Michael J. (30 December 2013). "Open Collaboration for Innovation: Principles and Performance". *Organization Science*. **25** (5): 1414–1433. arXiv:1406.7541 (<https://arxiv.org/abs/1406.7541>). doi:10.1287/orsc.2013.0872 (<https://doi.org/10.1287%2Forsc.2013.0872>). ISSN 1047-7039 (<https://search.worldcat.org/issn/1047-7039>). S2CID 6583883 (<https://api.semanticscholar.org/CorpusID:6583883>).
267. Definition of "overfitting" (<https://web.archive.org/web/20171107014257/https://en.oxforddictionaries.com/definition/overfitting>) at OxfordDictionaries.com: this definition is specifically for statistics.
268. Bishop, Christopher M. (2006). Pattern Recognition and Machine Learning (PDF). Springer. p. vii. Pattern recognition has its origins in engineering, whereas machine learning grew out of computer science. However, these activities can be viewed as two facets of the same field, and together they have undergone substantial development over the past ten years.
269. Hughes, G. E., & Cresswell, M. J., *A New Introduction to Modal Logic* (<https://books.google.com/books?id=Dsn1xWNB4MEC&q=%22first-order+logic%22>) (London: Routledge, 1996), p.161 ([https://books.google.com/books?id=\\_CB5wiBeaA4C&pg=PA161](https://books.google.com/books?id=_CB5wiBeaA4C&pg=PA161)).
270. Nyce, Charles (2007), *Predictive Analytics White Paper* (<https://the-digital-insurer.com/wp-content/uploads/2013/12/78-Predictive-Modeling-White-Paper.pdf>) (PDF), American Institute for Chartered Property Casualty Underwriters/Insurance Institute of America, p. 1



271. Eckerson, Wayne (10 May 2007), *Extending the Value of Your Data Warehousing Investment* ([https://tdwi.org/articles/2007/05/10/predictive-analytics.aspx?sc\\_lang=en](https://tdwi.org/articles/2007/05/10/predictive-analytics.aspx?sc_lang=en)), The Data Warehouse Institute
272. Karl R. Popper, *The Myth of Framework*, London (Routledge) 1994, chap. 8.
273. Karl R. Popper, *The Poverty of Historicism*, London (Routledge) 1960, chap. iv, sect. 31.
274. "Probabilistic programming does in 50 lines of code what used to take thousands" (<https://phys.org/news/2015-04-probabilistic-lines-code-thousands.html>). *phys.org*. 13 April 2015. Retrieved 13 April 2015.
275. "Probabilistic Programming" (<https://web.archive.org/web/20160110035042/https://probabilistic-programming.org/wiki/Home>). *probabilistic-programming.org*. Archived from the original (<https://probabilistic-programming.org/wiki/Home>) on 10 January 2016. Retrieved 31 July 2019.
276. Pfeffer, Avrom (2014), *Practical Probabilistic Programming*, Manning Publications. p.28. ISBN 978-1 6172-9233-0
277. Clocksin, William F.; Mellish, Christopher S. (2003). *Programming in Prolog*. Berlin; New York: Springer-Verlag. ISBN 978-3-540-00678-7.
278. Bratko, Ivan (2012). *Prolog programming for artificial intelligence* (4th ed.). Harlow, England; New York: Addison Wesley. ISBN 978-0-321-41746-6.
279. Covington, Michael A. (1994). *Natural language processing for Prolog programmers*. Englewood Cliffs, N.J.: Prentice Hall. ISBN 978-0-13-629213-5.
280. Lloyd, J. W. (1984). *Foundations of logic programming*. Berlin: Springer-Verlag. ISBN 978-3-540-13299-8.
281. Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012.
282. Yegulalp, Serdar (19 January 2017). "Facebook brings GPU-powered machine learning to Python" (<https://www.infoworld.com/article/3159120/artificial-intelligence/facebook-brings-gpu-powered-machine-learning-to-python.html>). *InfoWorld*. Retrieved 11 December 2017.
283. Lorica, Ben (3 August 2017). "Why AI and machine learning researchers are beginning to embrace PyTorch" (<https://www.oreilly.com/ideas/why-ai-and-machine-learning-researchers-are-beginning-to-embrace-pytorch>). O'Reilly Media. Retrieved 11 December 2017.
284. Ketkar, Nikhil (2017). "Introduction to PyTorch". *Deep Learning with Python*. Apress, Berkeley, CA. pp. 195–208. doi:10.1007/978-1-4842-2766-4\_12 ([https://doi.org/10.1007%2F978-1-4842-2766-4\\_12](https://doi.org/10.1007%2F978-1-4842-2766-4_12)). ISBN 9781484227657.
285. Moez Ali (June 2023). "NLP with PyTorch: A Comprehensive Guide" (<https://www.datacamp.com/tutorial/nlp-with-pytorch-a-comprehensive-guide>). *datacamp.com*. Retrieved 1 April 2024.
286. Patel, Mo (7 December 2017). "When two trends fuse: PyTorch and recommender systems" (<https://www.oreilly.com/ideas/when-two-trends-fuse-pytorch-and-recommender-systems>). O'Reilly Media. Retrieved 18 December 2017.
287. Mannes, John. "Facebook and Microsoft collaborate to simplify conversions from PyTorch to Caffe2" (<https://techcrunch.com/2017/09/07/facebook-and-microsoft-collaborate-to-simplify-conversions-from-pytorch-to-caffe2/>). *TechCrunch*. Retrieved 18 December 2017. "FAIR is accustomed to working with PyTorch – a deep learning framework optimized for achieving state of the art results in research, regardless of resource constraints. Unfortunately in the real world, most of us are limited by the computational capabilities of our smartphones and computers."
288. Arakelyan, Sophia (29 November 2017). "Tech giants are using open source frameworks to dominate the AI community" (<https://venturebeat.com/2017/11/29/tech-giants-are-using-open-source-frameworks-to-dominate-the-ai-community/>). *VentureBeat*. Retrieved 18 December 2017.

289. "PyTorch strengthens its governance by joining the Linux Foundation" (<https://pytorch.org/blog/PyTorchfoundation/>). *pytorch.org*. Retrieved 13 September 2022.
290. Reiter, Raymond (2001). *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems* ([https://archive.org/details/knowledgeactionl00reit\\_022](https://archive.org/details/knowledgeactionl00reit_022)). Cambridge, Massachusetts: The MIT Press. pp. 20 ([https://archive.org/details/knowledgeactionl00reit\\_022/page/n40](https://archive.org/details/knowledgeactionl00reit_022/page/n40))–22. ISBN 9780262527002.
291. Thielscher, Michael (September 2001). "The Qualification Problem: A solution to the problem of anomalous models". *Artificial Intelligence*. **131** (1–2): 1–37. doi:10.1016/S0004-3702(01)00131-X (<https://doi.org/10.1016%2FS0004-3702%2801%2900131-X>).
292. Grumbling, Emily; Horowitz, Mark, eds. (2019). *Quantum Computing : Progress and Prospects (2018)*. Washington, DC: National Academies Press. p. I-5. doi:10.17226/25196 (<https://doi.org/10.17226%2F25196>). ISBN 978-0-309-47969-1. OCLC 1081001288 (<https://search.worldcat.org/oclc/1081001288>). S2CID 125635007 (<https://api.semanticscholar.org/CorpusID:125635007>).
293. R language and environment Hornik, Kurt (4 October 2017). "R FAQ" ([https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R\\_003f](https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R_003f)). *The Comprehensive R Archive Network*. 2.1 What is R?. Retrieved 6 August 2018. R Foundation Hornik, Kurt (4 October 2017). "R FAQ" ([https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-the-R-Foundation\\_003f](https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-the-R-Foundation_003f)). *The Comprehensive R Archive Network*. 2.13 What is the R Foundation?. Retrieved 6 August 2018. The R Core Team asks authors who use R in their data analysis to cite the software using: R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://R-project.org/>.
294. widely used Fox, John & Andersen, Robert (January 2005). "Using the R Statistical Computing Environment to Teach Social Statistics Courses" (<https://socialsciences.mcmaster.ca/jfox/Teaching-with-R.pdf>) (PDF). Department of Sociology, McMaster University. Retrieved 6 August 2018. Vance, Ashlee (6 January 2009). "Data Analysts Captivated by R's Power" (<https://nytimes.com/2009/01/07/technology/business-computing/07program.html>). *The New York Times*. Retrieved 6 August 2018. "R is also the name of a popular programming language used by a growing number of data analysts inside corporations and academia. It is becoming their lingua franca..."
295. Vance, Ashlee (6 January 2009). "Data Analysts Captivated by R's Power" (<https://nytimes.com/2009/01/07/technology/business-computing/07program.html>). *The New York Times*. Retrieved 6 August 2018. "R is also the name of a popular programming language used by a growing number of data analysts inside corporations and academia. It is becoming their lingua franca..."
296. Broomhead, D. S.; Lowe, David (1988). *Radial basis functions, multi-variable functional interpolation and adaptive networks* (<https://apps.dtic.mil/sti/pdfs/ADA196234.pdf>) (PDF) (Technical report). RSRE. 4148. Archived (<https://web.archive.org/web/20130409223044/https://dtic.mil/cgi-bin/GetTRDoc?AD=ADA196234>) from the original on 9 April 2013.
297. Broomhead, D. S.; Lowe, David (1988). "Multivariable functional interpolation and adaptive networks" (<https://sci2s.ugr.es/keel/pdf/algorithm/articulo/1988-Broomhead-CS.pdf>) (PDF). *Complex Systems*. **2**: 321–355.
298. Schwenker, Friedhelm; Kestler, Hans A.; Palm, Günther (2001). "Three learning phases for radial-basis-function networks". *Neural Networks*. **14** (4–5): 439–458. doi:10.1016/s0893-6080(01)00027-2 (<https://doi.org/10.1016%2FS0893-6080%2801%2900027-2>). PMID 11411631 (<https://pubmed.ncbi.nlm.nih.gov/11411631>).
299. Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. Archived from the original (PDF) on 17 April 2016. Retrieved 5 June 2016.
300. Ho, TK (1998). "The Random Subspace Method for Constructing Decision Forests" (<https://repository.unal.edu.co/handle/unal/81834>). *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **20** (8): 832–844. doi:10.1109/34.709601 (<https://doi.org/10.1109%2F34.709601>). S2CID 206420153 (<https://api.semanticscholar.org/CorpusID:206420153>).

301. Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome(2008). *The Elements of Statistical Learning* (2nd ed.). Springer. ISBN 0-387-95284-5.
302. Graves, A.; Liwicki, M.; Fernandez, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. (2009). "A Novel Connectionist System for Improved Unconstrained Handwriting Recognition" ([https://dsia.ch/~juergen/tpami\\_2008.pdf](https://dsia.ch/~juergen/tpami_2008.pdf)) (PDF). *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **31** (5): 855–868. CiteSeerX 10.1.1.139.4502 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.139.4502>). doi:10.1109/tpami.2008.137 (<https://doi.org/10.1109%2Ftpami.2008.137>). PMID 19299860 (<https://pubmed.ncbi.nlm.nih.gov/19299860/>). S2CID 14635907 (<https://api.semanticscholar.org/CorpusID:14635907>).
303. Sak, Hasim; Senior, Andrew; Beaufays, Francoise (2014). "Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling" (<https://web.archive.org/web/20180424203806/https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf>) (PDF). Archived from the original (<https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf>) (PDF) on 24 April 2018. Retrieved 6 August 2019.
304. Li, Xiangang; Wu, Xihong (15 October 2014). "Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition". arXiv:1410.4281 (<https://arxiv.org/abs/1410.4281>) [cs.CL (<https://arxiv.org/archive/cs/CL>)].
305. Kaelbling, Leslie P.; Littman, Michael L.; Moore, Andrew W. (1996). "Reinforcement Learning: A Survey" (<http://web.archive.loc.gov/all/20011120234539/https://cs.washington.edu/research/jair/abstracts/kaelbling96a.html>). *Journal of Artificial Intelligence Research*. **4**: 237–285. arXiv:cs/9605103 (<https://arxiv.org/abs/cs/9605103>). doi:10.1613/jair.301 (<https://doi.org/10.1613%2Fjair.301>). S2CID 1708582 (<https://api.semanticscholar.org/CorpusID:1708582>). Archived from the original (<https://cs.washington.edu/research/jair/abstracts/kaelbling96a.html>) on 20 November 2001. Retrieved 5 July 2022.
306. Patrizio, Andy. "What is reinforcement learning from human feedback (RLHF)?" (<https://www.techtarget.com/whatis/definition/reinforcement-learning-from-human-feedback-RLHF>). *TechTarget*. Retrieved 28 January 2024.
307. Schrauwen, Benjamin, David Verstraeten, and Jan Van Campenhout. "An overview of reservoir computing: theory, applications, and implementations." *Proceedings of the European Symposium on Artificial Neural Networks ESANN 2007*, pp. 471–482.
308. Mass, Wolfgang; Nachtschlaeger, T.; Markram, H. (2002). "Real-time computing without stable states: A new framework for neural computation based on perturbations". *Neural Computation*. **14** (11): 2531–2560. doi:10.1162/089976602760407955 (<https://doi.org/10.1162%2F089976602760407955>). PMID 12433288 (<https://pubmed.ncbi.nlm.nih.gov/12433288/>). S2CID 1045112 (<https://api.semanticscholar.org/CorpusID:1045112>).
309. Jaeger, Herbert, "The echo state approach to analyzing and training recurrent neural networks." Technical Report 154 (2001), German National Research Center for Information Technology.
310. Jaeger, Herbert (2007). "Echo state network" (<https://doi.org/10.4249%2Fscholarpedia.2330>). *Scholarpedia*. **2** (9): 2330. Bibcode:2007SchpJ...2.2330J (<https://ui.adsabs.harvard.edu/abs/2007SchpJ...2.2330J>). doi:10.4249/scholarpedia.2330 (<https://doi.org/10.4249%2Fscholarpedia.2330>).
311. "XML and Semantic Web W3C Standards Timeline" (<https://web.archive.org/web/20220706210815/http://www.dblab.ntua.gr/~bikakis/XMLSemanticWebW3CTimeline.pdf>) (PDF). 4 February 2012. Archived from the original (<https://dblab.ntua.gr/~bikakis/XMLSemanticWebW3CTimeline.pdf>) (PDF) on 6 July 2022. Retrieved 5 July 2022.
312. See, for example, Boolos and Jeffrey, 1974, chapter 11.
313. Sowa, John F. (1987). "Semantic Networks" (<https://jfsowa.com/pubs/semnet.htm>). In Shapiro, Stuart C (ed.). *Encyclopedia of Artificial Intelligence*. Retrieved 29 April 2008.

314. O'Hearn, P. W.; Pym, D. J. (June 1999). "The Logic of Bunched Implications". *Bulletin of Symbolic Logic*. **5** (2): 215–244. CiteSeerX 10.1.1.27.4742 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.4742>). doi:10.2307/421090 (<https://doi.org/10.2307%2F421090>). JSTOR 421090 (<https://www.jstor.org/stable/421090>). S2CID 2948552 (<https://api.semanticscholar.org/CorpusID:2948552>).
315. Abran et al. 2004, pp. 1–1
316. "Computing Degrees & Careers" ([https://web.archive.org/web/20110617053818/https://computingcareers.acm.org/?page\\_id=12](https://web.archive.org/web/20110617053818/https://computingcareers.acm.org/?page_id=12)). ACM. 2007. Archived from the original ([https://computingcareers.acm.org/?page\\_id=12](https://computingcareers.acm.org/?page_id=12)) on 17 June 2011. Retrieved 23 November 2010.
317. Laplante, Phillip (2007). *What Every Engineer Should Know about Software Engineering* (<https://books.google.com/books?id=pFHYk0KWAEgC&q=What%20Every%20Engineer%20Should%20Know%20about%20Software%20Engineering.&pg=PA1>). Boca Raton: CRC. ISBN 978-0-8493-7228-5. Retrieved 21 January 2011.
318. Rapoza, Jim (2 May 2006). "SPARQL Will Make the Web Shine" (<https://eweek.com/development/sparql-will-make-the-web-shine>). *eWeek*. Retrieved 17 January 2007.
319. Segaran, Toby; Evans, Colin; Taylor, Jamie (2009). *Programming the Semantic Web* ([https://archive.org/details/programmingseman00sega\\_683](https://archive.org/details/programmingseman00sega_683)). O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. p. 84 ([https://archive.org/details/programmingseman00sega\\_683/page/n101](https://archive.org/details/programmingseman00sega_683/page/n101)). ISBN 978-0-596-15381-6.
320. Maass, Wolfgang (1997). "Networks of spiking neurons: The third generation of neural network models". *Neural Networks*. **10** (9): 1659–1671. doi:10.1016/S0893-6080(97)00011-7 (<https://doi.org/10.1016%2FS0893-6080%2897%2900011-7>). ISSN 0893-6080 (<https://search.worldcat.org/issn/0893-6080>).
321. "What is stateless? - Definition from WhatIs.com" (<https://whatis.techtarget.com/definition/stateless>). *techtarget.com*.
322. Lise Getoor and Ben Taskar: *Introduction to statistical relational learning* (<https://books.google.com/books?id=ISklewOw2WoC>), MIT Press, 2007
323. Ryan A. Rossi, Luke K. McDowell, David W. Aha, and Jennifer Neville, "Transforming Graph Data for Statistical Relational Learning. (<https://jair.org/media/3659/live-3659-6589-jair.pdf>) Archived (<https://web.archive.org/web/20180106202217/http://www.jair.org/media/3659/live-3659-6589-jair.pdf>) 6 January 2018 at the Wayback Machine" *Journal of Artificial Intelligence Research (JAIR)*, **Volume 45** (2012), pp. 363–441.
324. Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization* (<https://jhupl.edu/ISO>). Wiley. ISBN 978-0-471-33052-3.
325. *Language Understanding Using Two-Level Stochastic Models* by F. Pla, et al, 2001, Springer Lecture Notes in Computer Science ISBN 978-3-540-42557-1
326. Stuart J. Russell, Peter Norvig (2010) *Artificial Intelligence: A Modern Approach, Third Edition*, Prentice Hall ISBN 9780136042594.
327. Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar (2012) *Foundations of Machine Learning*, The MIT Press ISBN 9780262018258.
328. Cortes, Corinna; Vapnik, Vladimir N (1995). "Support vector networks" (<https://doi.org/10.1007%2FBF00994018>). *Machine Learning*. **20** (3): 273–297. doi:10.1007/BF00994018 (<https://doi.org/10.1007%2FBF00994018>).
329. Beni, G.; Wang, J. (1993). "Swarm Intelligence in Cellular Robotic Systems". *Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30 (1989)*. pp. 703–712. doi:10.1007/978-3-642-58069-7\_38 ([https://doi.org/10.1007%2F978-3-642-58069-7\\_38](https://doi.org/10.1007%2F978-3-642-58069-7_38)). ISBN 978-3-642-63461-1.
330. Haugeland 1985, p. 255.
331. Poole, Mackworth & Goebel 1998, p. 1.
332. "Collection of sources defining "singularity" " (<https://singularitysymposium.com/definition-of-singularity.html>). *singularitysymposium.com*. Retrieved 17 April 2019.

333. Eden, Amnon H.; Moor, James H. (2012). *Singularity hypotheses: A Scientific and Philosophical Assessment* (<https://archive.org/details/singularityhypot00mueh>). Dordrecht: Springer. pp. 1 (<https://archive.org/details/singularityhypot00mueh/page/n9>)–2. ISBN 9783642325601.
334. Cadwalladr, Carole (2014). "Are the robots about to rise? Google's new director of engineering thinks so... (<https://theguardian.com/technology/2014/feb/22/robots-google-ray-kurzweil-terminator-singularity-artificial-intelligence>)" *The Guardian*. Guardian News and Media Limited.
335. Sutton, Richard & Andrew Barto (1998). *Reinforcement Learning* (<https://web.archive.org/web/20170330005640/https://incompleteideas.net/sutton/book/the-book.html>). MIT Press. ISBN 978-0-585-02445-5. Archived from the original (<https://incompleteideas.net/sutton/book/the-book.html>) on 30 March 2017.
336. Pellionisz, A.; Llinás, R. (1980). "Tensorial Approach To The Geometry Of Brain Function: Cerebellar Coordination Via A Metric Tensor" ([https://academia.edu/download/31409354/pellionisz\\_1980\\_cerebellar\\_coordination\\_via\\_a\\_metric\\_tensor\\_fullpaper.pdf](https://academia.edu/download/31409354/pellionisz_1980_cerebellar_coordination_via_a_metric_tensor_fullpaper.pdf)) (PDF). *Neuroscience*. **5** (7): 1125—1136. doi:10.1016/0306-4522(80)90191-8 ([https://doi.org/10.1016/0306-4522\(80\)90191-8](https://doi.org/10.1016/0306-4522(80)90191-8)). PMID 6967569 (<https://pubmed.ncbi.nlm.nih.gov/6967569>). S2CID 17303132 (<https://api.semanticscholar.org/CorpusID:17303132>).
337. Pellionisz, A.; Llinás, R. (1985). "Tensor Network Theory Of The Metaorganization Of Functional Geometries In The Central Nervous System". *Neuroscience*. **16** (2): 245–273. doi:10.1016/0306-4522(85)90001-6 ([https://doi.org/10.1016/0306-4522\(85\)90001-6](https://doi.org/10.1016/0306-4522(85)90001-6)). PMID 4080158 (<https://pubmed.ncbi.nlm.nih.gov/4080158>). S2CID 10747593 (<https://api.semanticscholar.org/CorpusID:10747593>).
338. "TensorFlow: Open source machine learning" ([https://youtube.com/watch?v=oZikw5k\\_2FM](https://youtube.com/watch?v=oZikw5k_2FM)) "It is machine learning software being used for various kinds of perceptual and language understanding tasks" — Jeffrey Dean, minute 0:47 / 2:17 from YouTube clip
339. Sipser, Michael (2013). *Introduction to the Theory of Computation 3rd*. Cengage Learning. ISBN 978-1-133-18779-0. "central areas of the theory of computation: automata, computability, and complexity. (Page 1)"
340. Thompson, William R (1933). "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". *Biometrika*. **25** (3–4): 285–294. doi:10.1093/biomet/25.3-4.285 (<https://doi.org/10.1093/biomet/25.3-4.285>).
341. Russo, Daniel J.; Van Roy, Benjamin; Kazerouni, Abbas; Osband, Ian; Wen, Zheng (2018). "A Tutorial on Thompson Sampling". *Foundations and Trends in Machine Learning*. **11** (1): 1–96. arXiv:1707.02038 (<https://arxiv.org/abs/1707.02038>). doi:10.1561/22000000070 (<https://doi.org/10.1561/22000000070>). S2CID 3929917 (<https://api.semanticscholar.org/CorpusID:3929917>).
342. West, Jeremy; Ventura, Dan; Warnick, Sean (2007). "Spring Research Presentation: A Theoretical Foundation for Inductive Transfer" (<https://web.archive.org/web/20070801120743/http://cpms.byu.edu/springresearch/abstract-entry?id=861>). Brigham Young University, College of Physical and Mathematical Sciences. Archived from the original (<http://cpms.byu.edu/springresearch/abstract-entry?id=861>) on 1 August 2007. Retrieved 5 August 2007.
343. Dickson, Ben (2 May 2022). "Machine learning: What is the transformer architecture?" (<https://bdtechtalks.com/2022/05/02/what-is-the-transformer/>). *TechTarget*. Retrieved 2 May 2022.
344. Mercer, Calvin. *Religion and Transhumanism: The Unknown Future of Human Enhancement*. Praeger.
345. Bostrom, Nick (2005). "A history of transhumanist thought" (<https://nickbostrom.com/papers/history.pdf>) (PDF). *Journal of Evolution and Technology*. Retrieved 21 February 2006.

346. Minsky 1967:107 "In his 1936 paper, A. M. Turing defined the class of abstract machines that now bear his name. A Turing machine is a finite-state machine associated with a special kind of environment – its tape – in which it can store (and later recover) sequences of symbols," also Stone 1972:8 where the word "machine" is in quotation marks.
347. Stone 1972:8 states "This "machine" is an abstract mathematical model", also cf. Sipser 2006:137ff that describes the "Turing machine model". Rogers 1987 (1967):13 refers to "Turing's characterization", Boolos Burgess and Jeffrey 2002:25 refers to a "specific kind of idealized machine".
348. Sipser 2006:137 "A Turing machine can do everything that a real computer can do".
349. Turing originally suggested a teleprinter, one of the few text-only communication systems available in 1950. (Turing 1950, p. 433)
350. Pierce 2002, p. 1: "A type system is a tractable syntactic method for proving the absence of certain program behaviors by classifying phrases according to the kinds of values they compute."
351. Cardelli 2004, p. 1: "The fundamental purpose of a type system is to prevent the occurrence of execution errors during the running of a program."
352. Hinton, Jeffrey; Sejnowski, Terrence (1999). *Unsupervised Learning: Foundations of Neural Computation*. MIT Press. ISBN 978-0262581684.
353. Colaner, Seth; Humrick, Matthew (3 January 2016). "A third type of processor for AR/VR: Movidius' Myriad 2 VPU" (<https://tomshardware.com/news/movidius-myriad2-vpu-vision-processing-vr,30850.html>). *Tom's Hardware*.
354. Banerje, Prasad (28 March 2016). "The rise of VPUs: Giving Eyes to Machines" (<https://web.archive.org/web/20181211221429/https://www.digit.in/general/the-rise-of-vpus-giving-eyes-to-machines-29561.html>). *Digit.in*. Archived from the original (<https://digit.in/general/the-rise-of-vpus-giving-eyes-to-machines-29561.html>) on 11 December 2018. Retrieved 5 July 2022.
355. "DeepQA Project: FAQ" (<https://web.archive.org/web/20151105125722/https://www.research.ibm.com/deepqa/faq.shtml>). *IBM*. Archived from the original (<https://research.ibm.com/deepqa/faq.shtml>) on 5 November 2015. Retrieved 11 February 2011.
356. Ferrucci, David; Levas, Anthony; Bagchi, Sugato; Gondek, David; Mueller, Erik T. (1 June 2013). "Watson: Beyond Jeopardy!" (<https://doi.org/10.1016%2Fj.artint.2012.06.009>). *Artificial Intelligence*. **199**: 93–105. doi:10.1016/j.artint.2012.06.009 (<https://doi.org/10.1016%2Fj.artint.2012.06.009>).
357. Hale, Mike (8 February 2011). "Actors and Their Roles for \$300, HAL? HAL!" (<https://nytimes.com/2011/02/09/arts/television/09nova.html>). *The New York Times*. Retrieved 11 February 2011.
358. "The DeepQA Project" (<https://web.archive.org/web/20130121103239/http://www.research.ibm.com/deepqa/deepqa.shtml>). *IBM Research*. Archived from the original (<https://research.ibm.com/deepqa/deepqa.shtml>) on 21 January 2013. Retrieved 18 February 2011.
359. io9.com mentions narrow AI. Published 1 April 2013. Retrieved 16 February 2014: <https://io9.com/how-much-longer-before-our-first-ai-catastrophe-464043243>
360. AI researcher Ben Goertzel explains why he became interested in AGI instead of narrow AI. Published 18 Oct 2013. Retrieved 16 February 2014. <https://intelligence.org/2013/10/18/ben-goertzel/>
361. TechCrunch discusses AI App building regarding Narrow AI. Published 16 Oct 2015. Retrieved 17 Oct 2015. <https://techcrunch.com/2015/10/15/machine-learning-its-the-hard-problems-that-are-valuable/>
362. Jurafsky, Daniel; H. James, Martin (2000). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition* (<https://web.stanford.edu/~jurafsky/slp3/>). Upper Saddle River, N.J.: Prentice Hall. ISBN 978-0-13-095069-7.



363. "GitHub project webpage" (<https://github.com/dmlc/xgboost>). *GitHub*. June 2022. Archived (<https://web.archive.org/web/20210401110045/https://github.com/dmlc/xgboost>) from the original on 1 April 2021. Retrieved 5 April 2016.

## Works cited

- Abran, Alain; Moore, James W.; Bourque, Pierre; Dupuis, Robert; Tripp, Leonard L. (2004). *Guide to the Software Engineering Body of Knowledge*. IEEE. ISBN 978-0-7695-2330-9.
- Cardelli, Luca (2004). "Type systems" (<http://lucacardelli.name/Papers/TypeSystems.pdf>) (PDF). In Allen B. Tucker (ed.). *CRC Handbook of Computer Science and Engineering* (2nd ed.). CRC Press. ISBN 978-1584883609.
- Haugeland, John (1985). *Artificial Intelligence: The Very Idea*. Cambridge, Mass.: MIT Press. ISBN 978-0-262-08153-5.
- Legg, Shane; Hutter, Marcus (15 June 2007). "A Collection of Definitions of Intelligence". arXiv:0706.3639 (<https://arxiv.org/abs/0706.3639>) [cs.AI (<https://arxiv.org/archive/cs.AI>)].
- Mitchell, Melanie (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press. ISBN 9780585030944.
- 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.
- Pierce, Benjamin C. (2002). *Types and Programming Languages*. MIT Press. ISBN 978-0-262-16209-8.
- 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.
- 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
- Turing, Alan (October 1950). "Computing Machinery and Intelligence" (<https://academic.oup.com/mind/article/LIX/236/433/986238>). *Mind*. **59** (236): 433–460. doi:10.1093/mind/LIX.236.433 (<https://doi.org/10.1093/mind/LIX.236.433>). ISSN 1460-2113 (<https://search.worldcat.org/issn/1460-2113>). JSTOR 2251299 (<https://www.jstor.org/stable/2251299>). S2CID 14636783 (<https://api.semanticscholar.org/CorpusID:14636783>).

## Notes

---

1. *polynomial time* refers to how quickly the number of operations needed by an algorithm, relative to the size of the problem, grows. It is therefore a measure of efficiency of an algorithm.

---

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Glossary\\_of\\_artificial\\_intelligence&oldid=1271292959](https://en.wikipedia.org/w/index.php?title=Glossary_of_artificial_intelligence&oldid=1271292959)"