A Glossary

livebook.manning.com

This appendix defines a selection of key terms from throughout the book.

activation function—A function that transforms the output of a neuron in an *artificial neural network*, generally to render it capable of handling nonlinear transformations or to ensure its output value is clamped within some range (chapter 7).

acyclic—A graph with no *cycles* (chapter 4).

admissible heuristic—A *heuristic* for the A* search algorithm that never over-estimates the cost to reach the goal (chapter 2).

artificial neural network—A simulation of a biological *neural network* using computational tools to solve problems not easily reduced into forms amenable to traditional algorithmic approaches. Note that the operation of an artificial neural network generally strays significantly from its biological counterpart (chapter 7).

auto-memoization—A version of *memoization* implemented at the language level, in which the results of function calls without side effects are stored for lookup upon further identical calls (chapter 1).

backpropagation—A technique used for training neural network weights according to a set of inputs with known correct outputs. Partial derivatives are used to calculate each weight's "responsibility" for the error between actual results and expected results. These "deltas" are used to update the weights for future runs (chapter 7).

backtracking—Returning to an earlier decision point (to go a different direction than was last pursued) after hitting a wall in a search problem (chapter 3).

bit string—A data structure that stores a sequence of 1's and 0's represented using a single bit of memory for each. This is sometimes referred to as a "bit vector" or "bit array" (chapter 1).

centroid—The center point in a cluster. Typically, each dimension of this point is the mean of the rest of the points in that dimension (chapter 6).

chromosome—In a genetic algorithm, each individual in the *population* is referred to as a chromosome (chapter 5).

cluster—See *clustering* (chapter 6).

clustering—An *unsupervised learning* technique that divides a data set into groups of related points, known as *clusters* (chapter 6).

codon—A combination of three nucleotides that together form an amino acid (chapter 2).

compression—Encoding data (changing its form) to require less space (chapter 1).

connected—A graph property that indicates there is a path from any vertex to any other vertex (chapter 4).

constraint—A requirement that must be fulfilled in order for a constraint-satisfaction problem to be solved (chapter 3).

crossover—In a genetic algorithm, combining individuals from the population to create offspring that are a mixture of the parents, and that will be a part of the next generation (chapter 5).

CSV—A text interchange format in which rows of data sets have their values separated by commas, and the rows themselves are generally separated by newline characters. CSV stands for "comma-separated values." CSV is a common export format from spreadsheets and databases (chapter 7).

cycle—A path in a graph that visits the same vertex twice without backtracking (chapter 4).

decompression—Reversing the process of *compression*, returning the data to its original form (chapter 1).

deep learning—Something of a buzzword, deep learning can refer to any of several techniques that use advanced machine-learning algorithms to analyze "big data." Most commonly, deep learning refers to using multilayer artificial neural networks to solve problems using large data sets (chapter 7).

delta—A value that is representative of a gap between the expected value of a weight in a neural network and its actual value. The expected value is determined through the use of *training* data and *backpropagation* (chapter 7).

digraph—See *directed graph* (chapter 4).

directed graph—Also known as a *digraph*, a directed graph is a graph in which edges may only be traversed in one direction (chapter 4).

domain—The possible values of a variable in a constraint-satisfaction problem (chapter 3).

dynamic programming—Instead of solving a large problem outright using a brute-force approach, in dynamic programming the problem is broken up into smaller subproblems that are each more manageable (chapter 9).

edge—A connection between two *vertices* (nodes) in a graph (chapter 4).

exclusive or—See *XOR* (chapter 1).

feed-forward—A type of neural network in which signals propagate in one direction (chapter 7).

fitness function—A function that evaluates the effectiveness of a potential solution to a problem (chapter 5).

generation—One round in the evaluation of a genetic algorithm. Also used to refer to the population of individuals active in a round (chapter 5).

genetic programming—Programs that modify themselves using the *selection*, *crossover*, and *mutation* operators to find solutions to programming problems that are non-obvious (chapter 5).

gradient descent—The method of modifying an *artificial neural network's* weights using the *deltas* calculated during *backpropagation* and the *learning rate* (chapter 7).

graph—An abstract mathematical construct that is used for modeling a real-world problem by dividing the problem into a set of connected nodes. The nodes are known as *vertices*, and the connections are known as *edges* (chapter 4).

greedy algorithm—An algorithm that always selects the best immediate choice at any decision point, hopeful that it will lead to the globally optimal solution (chapter 4).

heuristic—An intuition about the way to solve a problem that points in the right direction (chapter 2).

hidden layer—Any layers between the input layer and the output layer in a *feed-forward artificial neural network* (chapter 7).

infinite loop—A loop that does not terminate (chapter 1).

infinite recursion—A set of recursive calls that does not terminate, but instead continues to make additional recursive calls. Analogous to an *infinite loop*. Usually caused by the lack of a base case (chapter 1).

input layer—The first layer of a *feed-forward artificial neural network* that receives its input from some kind of external entity (chapter 7).

learning rate—A value, usually a constant, used to adjust the rate at which weights are modified in an *artificial neural network*, based on calculated *deltas* (chapter 7).

memoization—A technique in which the results of computational tasks are stored for later retrieval from memory, saving additional computation time to recreate the same results (chapter 1).

minimum spanning tree—A *spanning tree* that connects all vertices using the minimum total weight of edges (chapter 4).

mutate—In a genetic algorithm, randomly changing some property of an individual before it is included in the next generation (chapter 5).

natural selection—The evolutionary process by which well adapted organisms succeed and poorly adapted organisms fail. Given a limited set of resources in the environment, the organisms best suited to leverage those resources will survive and propagate. Over several generations, this leads to helpful traits being propagated amongst a population, hence being naturally selected by the constraints of the environment (chapter 5).

neural network—A network of multiple neurons that act in concert to process information. The neurons are often thought about as being organized in layers (chapter 7).

neuron—An individual nerve cell, such as those in the human brain (chapter 7).

normalization—The process of making different types of data comparable (chapter 6).

NP-hard—A problem that belongs to a class of problems for which there is no known polynomial time algorithm to solve (chapter 9).

nucleotide—One instance of one of the four bases of DNA: adenine (A), cytosine (C), guanine (G), and thymine (T) (chapter 2).

output layer—The last layer in a *feed-forward artificial neural network* that is used for determining the result of the network for a given input and problem (chapter 7).

path—A set of edges that connects two vertices in a graph (chapter 4).

ply—A turn (often thought of as a move) in a two-player game (chapter 8).

population—In a genetic algorithm, the population is the collection of individuals (each representing a potential solution to the problem) competing to solve the problem (chapter 5).

priority queue—A data structure that pops items based on a "priority" ordering. For instance, a priority queue may be used with a collection of emergency calls in order to respond to the highest priority calls first (chapter 2).

queue—An abstract data structure that enforces the ordering FIFO (First-In-First-Out). A queue implementation provides at least the operations "push" and "pop" for adding and removing elements, respectively (chapter 2).

recursive function—A function that calls itself (chapter 1).

selection—The process of selecting individuals in a generation of a genetic algorithm for reproduction, to create individuals for the next generation (chapter 5).

sigmoid function—One of a set of popular *activation functions* used in *artificial neural networks*. The eponymous sigmoid function always returns a value between 0 and 1. It is also useful for ensuring results beyond just linear transformations can be represented by the network (chapter 7).

SIMD instructions—Microprocessor instructions optimized for doing calculations using vectors, also sometimes known as "vector instructions." SIMD stands for "single instruction, multiple data" (chapter 7).

spanning tree—A tree that connects every vertex in a graph (chapter 4).

stack—An abstract data structure that enforces the Last-In-First-Out (LIFO) ordering. A stack implementation provides at least the operations "push" and "pop" for adding and removing elements, respectively (chapter 2).

synapses—Gaps between *neurons* in which neurotransmitters are released to allow for the conduction of electrical current. In layman's terms, these are the connections between *neurons* (chapter 7).

supervised learning—Any machine-learning technique in which the algorithm is somehow guided toward correct results using outside resources (chapter 7).

training—A phase in which an *artificial neural network* has its weights adjusted by using *backpropagation* with known correct outputs for some given inputs (chapter 7).

tree—A graph that has only one path between any two vertices. A tree is *acyclic* (chapter 4).

unsupervised learning—Any machine-learning technique that does not use foreknowledge to reach its conclusions. In other words, a technique that is not guided, but instead runs "on its own" (chapter 6).

variable—In the context of a constraint-satisfaction problem, a variable is some parameter that must be solved for as part of the problem's solution. The possible values of the variable are its *domain*. The requirements for a solution are one or more *constraints* (chapter 3).

vertex—A single node in a *graph* (chapter 4).

XOR—A logical bitwise operation that returns `true` when either of its operands is true, but not when both are true or neither is true. The abbreviation stands for *exclusive or*. In Swift, the `^` operator is used for XOR (chapter 1).

z-score—The number of standard deviations a data point is away from the mean of a data set (chapter 6).