Machine Learning Lexicon

Alexandre P. Bazanté

January 2020

Machine learning is a subset of artificial intelligence; it's defined in a variety of ways, usually paraphrasing the definition of learning itself. Concretely, it's an algorithm aimed to perform (performance is user defined) with minimal loss (loss is used defined) after being optimized with some 'experience' (typically training data, also user defined) without specific tasks being programmed by the user. There are many different flavors of machine learning, they are commonly separated along 2 axes, forming 4 classes of machine learning problems.

	Supervised Learning	Unsupervised Learning
Discrete	Classification	Clustering
Continous	Regression	Dimensionality Reduction

Discrete v Continuous should be self explanatory; in supervised learning, the training data contains desired outputs, which is not the case for unsupervised learning.
This provides a non exhaustive list of terms commonly used in machine learning (especially with neural networks), which may have conflicting colloquial uses.

Feature: The number of features is the number of nodes in the input layer *i.e.*, the size of the vector for each data point.

Neuron: The building block of a neural network. It takes input, does some math with them (usually a weighted sum) and produces **one** output.

Layer: A collection of neurons connected to the same input and output, but not to one another

- *hidden* — Any layer between the input and output layers
- *dense* — Also called fully-connected layer, a linear operation in which every input to that layer is connected to every output by a weight.
- *convolutional* — A linear operation using only a subset of the weights of a dense layer; nearby inputs are connected to nearby outputs. The convolution weights are shared across the layer.
- *pooling* — A layer in which each pre-defined batch in the input is replaced with a single output (the maximum of the batch or its average).
- *normalization* — A layer in which the input is scaled to produce a near zero mean and unit standard deviation output.

Neural network: A collection of layers. A single layer network is sometimes called a perceptron. Computing systems vaguely inspired from animal brains; they learn to perform tasks by considering examples, without being programmed with task specific rules. Learning results in finding a set of optimal intrinsic parameters.

Parameters (weights, bias): In vanilla neural networks, each neuron computes a weighted sum of its own inputs, adds its own bias, which produces neuron specific output. These parameters are to be optimized by the network in the learning process.

Hyper-parameters: Parameter whose value is set (by the user) before the learning process begins. They can be optimized by trial and error, numerically on a grid, but rarely analytically.

Activation: A transformation of each neuron's output before being passed to the next layer as input. They serve several purposes, in particular bounding the output to a specific range, and adding non-linearity to the network so the latter is able to learn non-linear relationships between the data.

- *sigmoid, tanh* — Those simply refer to the functions used as activation.
- *reLU* — Stands for rectified Linear Unit: piecewise linear function that outputs zero for all negative inputs, and is identity otherwise.
- *leaky-reLU* — reLU has a tendency to fail at training if the parameters aren't initialized a specific way. This can be overcome by adding replacing the piecewise zero by "-0.1x"
- *softmax* — An activation function similar to sigmoid, but renormalized to output statistical probabilities normalized to 1 across one layer, enabling a specific kind of error metric known as cross entropy loss.

Feed forward: Refers to the information moving only in one direction from the input nodes to the output; the connections between the nodes do not form a cycle.

Loss: Term commonly used to describe the error between the predicted and expected/true data. There are several flavors of loss: MSE, MAE, ...

Back propagation: Refers to the error gradient being propagated backwards through all the nodes using the chain rule.

Learning rate: The gradient of loss vs. parameters gives the direction in which to modify the parameters in order to reduce loss, and a relative step size to take. These steps are modulated by a global parameter that prevents steps from being too large, commonly called learning rate.

Epoch: Refers to the number of times the training data is propagated through the entire network, and the gradients back propagated to update the network parameters.

Stochastic gradient descent: When the training set is too large, the set of parameters for the neural network often can't be optimized globally over the whole data set. Stochastic gradient descent refers to randomly selecting subsets of the training set of a given **batch size** and optimizing the network parameters for that subset. The global parameters can be chosen, for example as the average of the parameters obtained for each subsets, or the one set of parameters that leads to the smallest error on the validation set.

This list will be updated as needed.