Notes on Machine Learning Nomenclature

Paul F. Roysdon, Ph.D.

I. Introduction

This is a collection of ML definitions found online.

A. bag of words

A representation of the words in a phrase or passage, irrespective of order. For example, bag of words represents the following three phrases identically:

- the dog jumps
- jumps the dog
- dog jumps the

Each word is mapped to an index in a sparse vector, where the vector has an index for every word in the vocabulary. For example, the phrase the dog jumps is mapped into a feature vector with non-zero values at the three indices corresponding to the words the, dog, and jumps. The non-zero value can be any of the following:

- A 1 to indicate the presence of a word.
- A count of the number of times a word appears in the bag. For example, if the phrase were the maroon dog is a dog with maroon fur, then both maroon and dog would be represented as 2, while the other words would be represented as 1.
- Some other value, such as the logarithm of the count of the number of times a word appears in the bag.

B. dropout regularization

A form of regularization useful in training neural networks. Dropout regularization works by removing a random selection of a fixed number of the units in a network layer for a single gradient step. The more units dropped out, the stronger the regularization. This is analogous to training the network to emulate an exponentially large ensemble of smaller networks. For full details, see Dropout: A Simple Way to Prevent Neural Networks from Overfitting.

C. embeddings

A categorical feature represented as a continuous-valued feature. Typically, an embedding is a translation of a high-dimensional vector into a low-dimensional space. For example, you can represent the words in an English sentence in either of the following two ways:

- As a million-element (high-dimensional) sparse vector in which all elements are integers. Each cell in the vector represents a separate English word; the value in a cell represents the number of times that word appears in a sentence. Since a single English sentence is unlikely to contain more than 50 words, nearly every cell in the vector will contain a 0. The few cells that aren't 0 will contain a low integer (usually 1) representing the number of times that word appeared in the sentence.
- As a several-hundred-element (low-dimensional) dense vector in which each element holds a floating-point value between 0 and 1. This is an embedding.

In TensorFlow, embeddings are trained by backpropagating loss just like any other parameter in a neural network.

D. feature extraction

Overloaded term having either of the following definitions:

- Retrieving intermediate feature representations calculated by an unsupervised or pretrained model (for example, hidden layer values in a neural network) for use in another model as input.
- Synonym for feature engineering.

E. feature set

The group of features your machine learning model trains on. For example, postal code, property size, and property condition might comprise a simple feature set for a model that predicts housing prices.

F. feature vector

The list of feature values representing an example passed into a model.

G. hyperparameter

The "knobs" that you tweak during successive runs of training a model. For example, learning rate is a hyperparameter.

Contrast with parameter.

H. L1 regularization

A type of regularization that penalizes weights in proportion to the sum of the absolute values of the weights. In models relying on sparse features, L1 regularization helps drive the weights of irrelevant or barely relevant features to exactly 0, which removes those features from the model. Contrast with L2 regularization.

I. L2 regularization

A type of regularization that penalizes weights in proportion to the sum of the squares of the weights. L2 regularization helps drive outlier weights (those with high positive or low negative values) closer to 0 but not quite to 0. (Contrast with L1 regularization.) L2 regularization always improves generalization in linear models.

J. *learning rate*

A scalar used to train a model via gradient descent. During each iteration, the gradient descent algorithm multiplies the learning rate by the gradient. The resulting product is called the gradient step.

Learning rate is a key hyperparameter.

K. *mini-batch*

A small, randomly selected subset of the entire batch of examples run together in a single iteration of training or inference. The batch size of a mini-batch is usually between 10 and 1,000. It is much more efficient to calculate the loss on a mini-batch than on the full training data.

L. *N-gram*

An ordered sequence of N words. For example, “truly madly” is a 2-gram. Because order is relevant, “madly truly” is a different 2-gram than truly madly.

Examples:

- bigram or 2-gram: to go, go to, eat lunch, eat dinner
- trigram or 3-gram: ate too much, three blind mice, the bell tolls
- 4-gram: walk in the park, dust in the wind, the boy ate lentils

Many natural language understanding models rely on N-grams to predict the next word that the user will type or say. For example, suppose a user typed three blind. An NLU model based on trigrams would likely predict that the user will next type mice.

Contrast N-grams with bag of words, which are unordered sets of words.

M. *one-hot encoding*

A sparse vector in which:

- One element is set to 1.
- All other elements are set to 0.

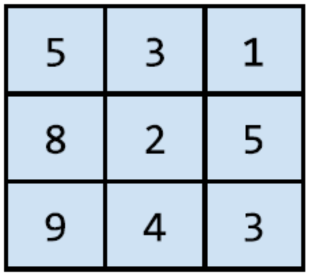
One-hot encoding is commonly used to represent strings or identifiers that have a finite set of possible values. For example, suppose a given botany dataset chronicles 15,000 different species, each denoted with a unique string identifier. As part of feature engineering, you’ll probably encode those string identifiers as one-hot vectors in which the vector has a size of 15,000.

N. *parameter*

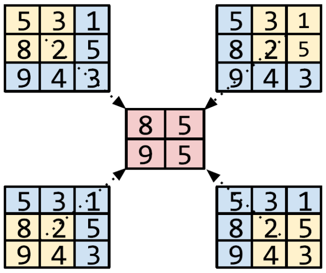
A variable of a model that the machine learning system trains on its own. For example, weights are parameters whose values the machine learning system gradually learns through successive training iterations. Contrast with hyperparameter.

O. *pooling*

Reducing a matrix (or matrices) created by an earlier convolutional layer to a smaller matrix. Pooling usually involves taking either the maximum or average value across the pooled area. For example, suppose we have the following 3x3 matrix: A pooling operation, just like a convolutional operation, divides that matrix into slices and then slides that convolutional operation by strides. For example, suppose the pooling operation divides the convolutional matrix into 2x2 slices with a 1x1 stride. As the following diagram illustrates, four pooling operations take place. Imagine that each pooling operation picks the maximum value of the four in that slice: Pooling helps enforce translational invariance in the input matrix.



5	3	1
8	2	5
9	4	3



Pooling for vision applications is known more formally as spatial pooling. Time-series applications usually refer to pooling as temporal pooling. Less formally, pooling is often called subsampling or downsampling.