Methods for Feature Learning and Extraction

Feature learning is widely regarded as a fundamental basis for many modern machine learning algorithms, specifically those classified under unsupervised learning. Its applications span a wide variety of fields, ranging from determining the connectedness of airports to classifying an apartment’s location based solely on a few properties. In particular, researchers have focused on image recognition; they are intrigued that the human brain can easily process and interpret images, yet even extremely powerful computers struggle to do the same. It is only logical then that researchers have turned to the human brain for inspiration and have attempted to mimic the visual cortex’s operations in code. Several methods and algorithms have developed over the years with some remarkable success, despite their drawbacks

In general, these algorithms follow a similar structure in terms of their derivation and operation. The basic equation that they try to solve is

where is the code matrix that represents a decomposed version of the input set and is the weight matrix, which varies based on the output we are trying to derive. From here, the models diverge by placing different constraints on the input, the code matrix, or the weight matrix. In some cases, the constraint is enforced by transforming the input to satisfy it. A reconstruction error is then defined and minimized to determine the best code matrix for the model. Some methods also outline a method for preprocessing the input, which helps increase the accuracy of the trained model.

In their paper on sparse coding, Bruno Olshausen and David Field utilize the above steps to recreate the visual cortex’s functionality to improve image recognition. From their perspective, the brain removes redundancy from the information it receives from the photoreceptors and represents the image as “a collection of independent events” (Olshausen 3311). Previous research in sparse coding focused on 2-dimensional redundancy reduction, which completely omits curves and other higher dimensional collections seen in natural images. To increase the accuracy of existing sparse coding algorithms, Olshausen and Field explore the creation of a linear strategy that can reduce higher order redundancy.

They use the following equation to create such a strategy, which is a slight modification of the feature extraction equation described earlier:

where represents the basis vectors that determine the image code and is the collection of amplitudes for each basis vector, which is computed for each image. The goal now is to find a set of basis vectors that can accurately represent the basic image structure of the input. An added constraint that Olshausen and Field make is that the basis functions must be overcomplete, which is when the number of basis vectors exceeds the dimensionality of the input. This will make the model more robust in the face of noise and help create greater flexibility in matching the model to the input since there can be multiple amplitudes that output the same image.

Finding the best set of basis vectors is analogous to creating a model with a probability distribution of each image () that matches the probability distribution of images experienced in nature () (3313). The Kullback-Leibler divergence is used as their reconstruction error, which is defined as:

where it is equal to 0 if and only if the two distributions are equivalent. Since the probability distribution of nature is fixed, Olshausen and Field focus on the log likelihood instead. The optimization function thus becomes:

and the learning rule for the model is:

where is defined as the residual image, or the difference between the output of the machine and the intended output. While their algorithm is quite accurate compared to other similar models, it is also slow and restricted. It assumes a linear image model, which is not always the case in nature. It also only consists of a single layer, preventing it from being as accurate as possible.

Aapo Hyvärinen and Erkki Oja describe another feature extraction algorithm known as ICA, or Independent Component Analysis. One of the many goals of ICA is to solve the cocktail party problem, which is to separate independent signals from a mixed signal input. They define their model as:

where is the mixing matrix and is the signal matrix. This is very similar to both the general model described earlier as well as the model derived for sparse coding. Once is estimated, the signal matrix can be derived with:

where is the inverse of .

Hyvärinen and Oja developed several constraints on their model that serves to differentiate it from other feature extraction algorithms. First of all, each component in is constrained to be statistically independent and have a non-Gaussian distribution. Non-Gaussianity is necessary for this model since a Gaussian distribution is symmetric, which does not give any indication of the direction of the columns in the mixing matrix, thus making it impossible to derive. Additionally, since both and are totally unknown, then the variance cannot be determined. Thus, the variance is arbitrarily fixed at 1 to simplify the model.

As seen earlier, the given derivation of the signal matrix requires that the mixing matrix be known. Since nothing can be assumed about the mixing matrix, Hyvärinen and Oja instead derive an estimator that can give a good estimation. They base their estimator off the equation:

where . Thus, they determined that their problem is to maximize the non-Gaussianity of . The researchers outline two main measures of non-Gaussianity, each of which has many advantages and disadvantages. They eventually determine that both estimators are impractical and instead focus on minimizing mutual information, which is defined as:

where is the differential entropy and is the density of . This equation is always nonnegative and equals 0 if and only if every variable is statistically independent. Hyvärinen and Oja continue to describe an additional measure of non-Gaussianity, but conclude that it is very much similar to minimizing the mutual information.

An area that particularly differentiates their approach from others is the process of preprocessing data. They describe two main ways in which this is done. The first is to center the input, which merely simplifies the computation of ICA. The second is to whiten the input, or to linearly transform it such that its components are uncorrelated and of equal variance. This makes the mixing matrix orthogonal and reduces the number of parameters that need to be estimated.

Non-negative Matrix Factorization (NMF) is an algorithm that has been around for several years and was the focus of a paper by Daniel Lee and H. Sebastian Seung. Much like the models previously explored, Lee and Seung use the following equation as the basis for NMF:

What makes this particular algorithm standout is its constraint. In fact, the constraint is in its very name – that the matrices in the equation are non-negative. The overall simplicity of this algorithm makes it very simple to implement, despite its increased computation time.

In determining a cost function, Lee and Seung discussed two distinct possibilities. The first is to minimize the Euclidean distance, which is defined as:

The second is to minimize the divergence, which is defined as:

Both functions converge, as proven by the researchers, making them excellent candidates for a cost function, though neither is definitively better than the other.

for the divergence cost function. These multiplicative equations are simply a rescaled version of the additive update rule, potentially increasing the speed at which the model is updated.