Normalization of Data

quantra

Table of Contents

Why Normalization?

The pre-processed data may contain attributes with a mixture of scales for various quantities such as prices, time and volume traded. Many machine learning methods expect or are more effective if the data attributes have the same scale. Two popular data scaling methods are normalization and standardization, which will be discussed in this document.

Example of unscaled data

	RSI	Volume	SMA	Close Price
Day 1	36	10000	150	162
Day 2	52	15000	152	175
Day 3	44	11000	145	160

Example of scaled data

	RSI	Volume	SMA	Close Price
Day 1	-1	-0.7559	0.27735	-0.4502
Day 2	1	1.13389	0.83205	1.14596
Day 3	0	-0.378	-1.1094	-0.6958

What is Standardization?

Before we understand Batch Normalization, we need to understand what standardization is. Standardization of data is the process by which we transform the data to have a zero mean and a unit variance. For example, we can do this by using the equation below.

$$x_{new} = \frac{x - \mu}{\sigma}$$

Let us understand this with a simple example:

A = [1,2,3,4,5]

The mean(m) of A is 3

The Standard deviation(std) of A is 1.58113883

When we apply the above standardization process to the first element of A:

A[0] = (1-m)/std

= (1-3)/1.58113883

= -1.264911064

If we apply the same process to all the elements of A, then the list A would like this:

A=[-1.264911064,-0.632455532,0,0.632455532,1.264911064]

What is Normalization?

Normalization of data is the process by which we transform the data to a range of 0 to 1. For example, we can do this by using the equation below.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Continuing the same example of list, A from above, when we apply this above transformation to the list A, then the resulting data set would look like this.

A= [0,0.25,0.5,0.75,1]

What is a Batch?

Now, let us look at the definition of a batch. It is a small sample taken from the actual training data. Batch size is one of the input parameters while training a neural network. During every epoch, the network would be trained on different batches of data to optimize the loss function. If you have 'n' training examples and the batch size is specified as 'b', then the number batches per epoch would be 'n/b'.

What is Batch Normalization?

The idea of Batch Normalization is to normalize the inputs of a network such that they have a mean output activation of zero and a standard deviation of one. This is analogous to the definition of standardization.

Batch Normalization

In a network, the output of one layer is the input of other layer, so we normalize the out of one layer before feeding it to the activation function of the next layer. By doing this, we not only normalize the inputs of the network as a whole but also the inputs of every layer.

It's called "batch" normalization, because during training, we normalise the outputs of every layer for each batch, i.e. apply a transformation that maintains the mean of the output close to 0 and the standard deviation close to 1 for every batch.

Apart from the above-mentioned reasons, there are a few mathematical reasons as to why it helps the network learn better. It helps combat what the authors call internal covariate shift. This is discussed in the original paper and the Deep Learning book (Goodfellow et al), in section 8.7.1 of Chapter 8.

Each layer in a neural net tries to model the input from the previous layer, so each layer actually tries to adapt to its input, but for hidden layers things get a bit complicated. The input's statistical distribution changes after a few iterations, so if the input's statistical distribution keeps changing, called internal covariate shift, the hidden layers will keep trying to adapt to that new distribution, slowing down the convergence. It is like a goal that keeps changing for hidden layers.

So the batch normalization algorithm tries to normalize the inputs to each hidden layer so that their distribution is fairly constant as the training proceeds. This improves convergence of the neural net.

Benefits of Batch Normalization:

The intention behind batch normalisation is to optimise network training. It has been shown to have several benefits:

Networks train faster

Although each training iteration will be slower as the extra normalisation calculations during the forward propagation and the additional hyperparameters to train during back propagation increase, the final convergence will happen much more quickly, so the training will be faster overall.

Allows higher learning rates

Gradient descent usually requires small learning rates for the network to converge. As networks get deeper, gradients get smaller during back propagation, and so require even more iterations. Using batch normalisation allows much higher learning rates, increasing the speed at which networks train.

Makes weights easier to initialise

Weight initialisation can be difficult, especially when creating deeper networks. Batch normalisation helps reduce the sensitivity to the initial starting weights.

Makes more activation functions viable

Some activation functions don't work well in certain situations. Sigmoids can't be used in deep networks, and ReLUs often die out during training (stop learning completely), so we must be careful about the range of values fed into them. But as batch normalisation regulates the values going into each activation function, nonlinearities that don't work well in deep networks tend to become viable again.

Provides some regularisation

Batch normalisation adds a little noise to your network, and in some cases, (e.g. Inception modules) it has been shown to work as well as dropout. You can consider batch normalisation as a bit of extra regularization, allowing you to reduce some of the dropout you might add to a network.