Multi-Layer Perceptron

16BCE1259
Shushil Kumar Ravishankar

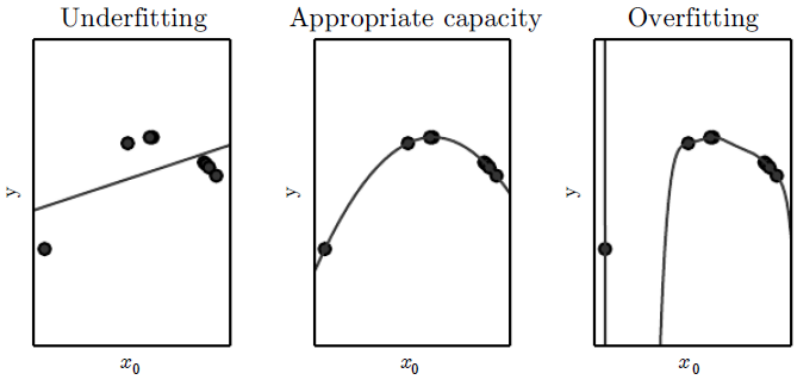
Report:

A multilayer perceptron (MLP) is a class of feedforward artificial neural network. An MLP consists of, at least, three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

Effect of increasing number of Hidden Layers:

Increasing the number of hidden layers might improve the accuracy or might not, it really depends on the complexity of the problem that you are trying to solve.

Increasing the number of hidden layers much more than the sufficient number of layers will cause accuracy in the test set to decrease, yes. It will cause your network to overfit to the training set, that is, it will learn the training data, but it won't be able to generalize to new unseen data. A picture taken from http://www.deeplearningbook.org/contents/ml.html gives a pretty good intuition for this concept.



Where in the left picture they try to fit a linear function to the data. This function is not complex enough to correctly represent the data, and it suffers from a bias(underfitting) problem. In the middle picture, the model has the appropriate complexity to accurately represent the data and to generalize, since it has learned the trend that this data follows (the data was synthetically created and has an inverted parabola shape). In the right picture, the model fits to the data, but it overfits to it, it hasn't learnt the trend and thus it is not able to generalize to new data.

Effect of increasing number of Epochs:

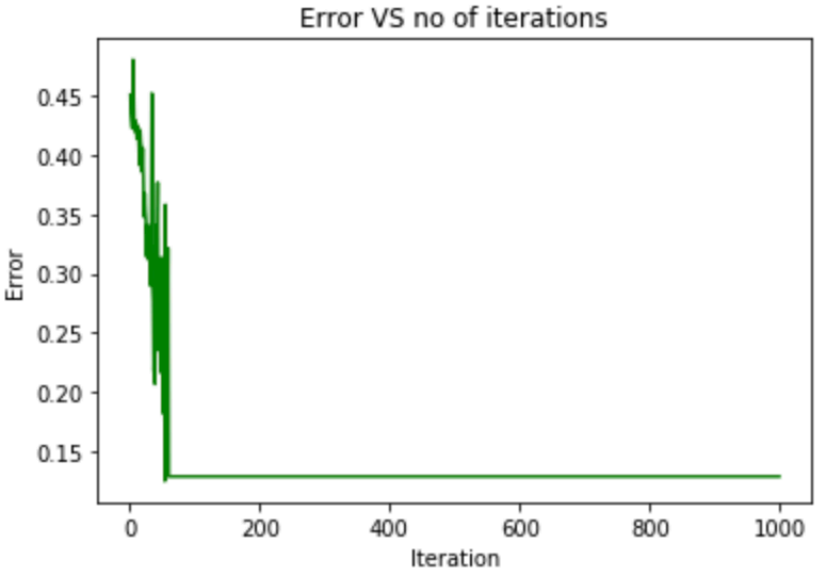
Increasing the number of epochs may lead to overfitting. There is an approach called “early stop”. In that approach, you plot the error rate on training and validation data. The horizontal axis is the error rate and the vertical axis is the error rate. You should stop training when the error rate of validation data is minimum. Consequently, if you increase the number of epochs, you will have an over-fitted model.

In general, too many epochs may cause your model to over-fit the training data. It means that your model does not learn the data, it memorizes the data. You have to find the accuracy of validation data for each epoch or maybe iteration to investigate whether it over-fits or not.

Observations on Big Mart Sales dataset:

Epochs:

In the Big Mart Sales dataset, after plotting the validation error vs the number of iterations of Multi-Layer Perceptron, it can be observed that the error is minimum at 54th iteration with an error of 12.5%. The error value stabilizes after a few more iterations.



Number of hidden layer and size of each layer:

After tabulating the error values for variable number of hidden layers (max 3 layers) and variable size of each layer (max 9 perceptron per layer), the minimum error was observed in a MLP with single hidden layer of size 5.

This means that, any size less than that is underfitting and size more than that is overfitting. The overfitting sizes may need more number of epochs to reduce their validation error.