EE550: ARTIFICIAL NEURAL NETWORKS

PROJECT 3: MULTILAYER PERCEPTRON

PROJECT REPORT

INTRODUCTION

In this project, multilayer perceptron algorithm was implemented for three different cases. In the first case MLP model were generated to predict XOR function. Then, f(x) = sinx was approximated by a neural network. Finally, a real world dataset was taken from an online resource and ANN was trained to predict its type. For second and final case, 3 and 4 layer networks were employed. In all cases the cost function was demonstrated to converge. In this report, the performance of ANNs for each case will be discussed.

ANALYSIS

For each case a momentum term of 0.25 was added in the gradient descent to average out the cost function. Also, to prevent overfitting, a regularization term, which penalizes the high order terms in thetas, were superimposed on the cost function. In each case, the convergence of the cost function was guaranteed.

First Case: Binary XOR

A dataset was created for the binary XOR function. A neural network with 3 layers with 2, 9 and 1 nodes in the input, hidden and output layers respectively was created. After training the cost function was demonstrated to converge to a lower treshold value. The model was tested for each case and the ANN was able to 100% accurately predict the output.

Second Case: Function Approximation

In the second case, f(x) = sinx was approximated by the generated three and four layered ANNs. Individual digits of the samples were fed to the input layer of the ANN. For example, if a generated sample data point happens to be 3.1416, the inputs to the first layer will be 3, 1, 4, 1, 6. A layer of ten nodes were used for the hidden layer of the network. After training, the algorithm was demonstrated to approximate well the sine function. In fact, both in 3 and 4 layered networks the cost function was shown to converge. However, the 4 layer network, approximates the sine function very sharply. Generally, the ANN predicts 1 in the interval x = [0, pi] and changes to 0 after a zigzag in x = pi in x = (pi, 2pi]. Since there are two hidden layers with ten nodes, the network is observed to saturate.

Third Case: Iris Dataset

In this final case, the Iris dataset from archive.ics.uci.edu was obtained and the ANN was trained on this dataset for 1 and 2 hidden layers. After training the ANN with 125 samples, the cost function for each 1 and 2 hidden layered networks was demonstrated to converge to a lower treshold value. After the convergence achieved , both networks predict with around 70% accuracy on average. In fact, this is around 20% lower than what the models should have scored.

In spite of meticulously reviewing the algorithm, it seemed that it is correct. The output however states otherwise. Though the convergence of cost functions is a sign that the algorithm should work well, it didn’t in practice. A probable cause of the error is that saturation of the sigmoid function.