Project 1

Geoffrey Clark
Ira A. Fulton Schools of Engineering
Arizona State University
Tempe, AZ
Email: gmclark1@asu.edu

Venkatavaradhan Lakshminarayanan
Ira A. Fulton Schools of Engineering
Arizona State University
Tempe, AZ
Email: vvlakshm@asu.edu

Shubham Sonawani
Ira A. Fulton Schools of Engineering
Arizona State University
Tempe, AZ
Email: sdsonawa@asu.edu

***Abstract*—In this project, we have shown implementation of different back propagation algorithms to train neural net using training data of 2000 examples (fig. 1) as stated in problem. we have training sets with two region having separation distant between region equal to 2, -4, -8. Here, Separation between two training region goes on decreasing which makes it complex non linear classification problem. Using Neural Net tool box of MATLAB, we have achieved to train multilayer perceptron as non linear classifier for different separation distance such as 2, -4, -8.**

I. Introduction

In data analysis, Key Property in most of the data sets that it has Nonlinear distribution and separation. To solve problem of nonlinear classification, we need to design neural net with multiple neural units in single layer. As given in problem statement, visualization of training data shows that we need neural net trained for non linear binary classification. In this case we can not use single perceptron for training as it provide only linear separations boundary. Thus, we have shown the implementation of state of the art back propagation and levenberg Marquardt algorithm.

II. Approach

A. Generating Data

We have created function that generates cluster of 2000 training data sets and 1000 test data sets having different separation distant of 2,-4, -8 units. Generated data cluster have width of 6 units and radius of 10 units. here, we have taken care of data seeds for all training data sets. to maintain randomness in data generation, each training and test data sets have different seeds. We need For details about our approach to generate data look at code GenerateClusters.m training data sets for different separation units can be visualize in fig.1 fig.2 and fig.3

B. Training Data

Here, we have three layer neural net with first layer as input layer , second layer as hidden layer and third as output layer. Basically, we have single hidden layer with multiple neural units. we are planning on designing the non linear classifier with data distribution as shown in fig. 1:3. We have trained network using different function available in MATLAB neural net toolbox such as for basic back propagation we have used 'traingd' (Gradient Descend Back Propagation), for back propagation with momentum we have used 'trainrp' (Resilient Back Propagation Algorithm) and for Levenberg Marquardt we have used 'trainlm' (Levenberg Marquardt). In this case, we have set the 75% of data is training data and 25% is used for cross validation. Initially we have set the network property to specific values which are constant for all three speration distances of 2, -4, -8.

C. Testing Data

Here, we have 1000 samples of testing data generated from GenerateClusters.m file.

1) Basic Back Propagation: Now consider simple back propagation algorithm, we are using 'traingd'(Gradient Descend Algorithm) available in MATLAB Neural Net toolbox. we have set the hidden units to 3. here learning rate is set to 0.5 with ephochs equal to 16000 value. We have shown decision boundary for test data, which can be visualized in fig.4

2) Back Propagation with Momentum: Here, We are using resilient back propagation algorithm available in MATLAB Neural Net tool box. Basically, main parameter here is momentum rate for back propagation algorithm. we have set the momentum rate to 0.7 and learning rate to 0.5. Result of Network for test data can be visualized in fig. 5

3) Levenberg Marquardt Back Propagation: In this case, use of algorithms very much similar to basic back propogation algorithm. here only parameter that is changing is function parameter which is 'trainlm' in network training.

Visualization of decision boundary can be seen in fig.6

III. Conclusion