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 [1], 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

In order to generate cluster of 1000 data points, we have used Mersenne Twister [2] generator. Here initial part of code generates cluster of 2000 training data sets and 1000 test data sets having different separation distant of 2,-4, -8 units.Here, Initial Data is obtained in the form of polar coordinates which then converted in rectangular coordinate. 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. Furthermore, to maintain randomness in data generation, each training and test data sets have different seeds[3]. Generated clusters for different Separation distances can be visualized in figure 1, figure 2, Figure 3.

B. Training Data

Initially, 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. As 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

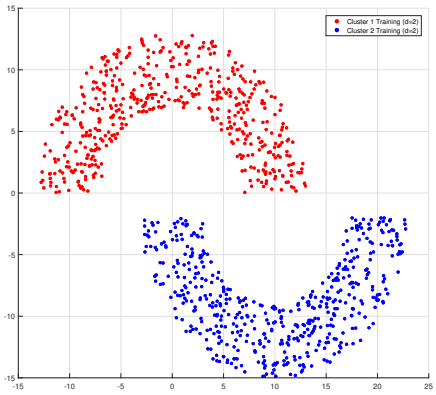

Fig. 1. Visualization of Clusters with separation of 2 units

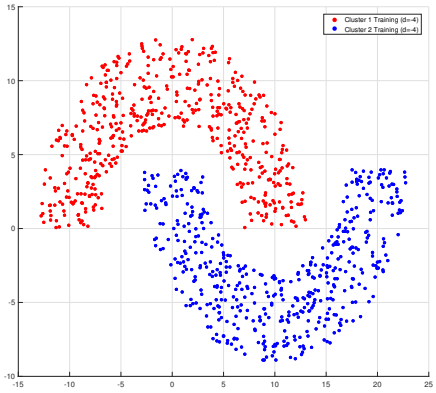

Fig. 2. Visualization of Clusters with separation of -4 units

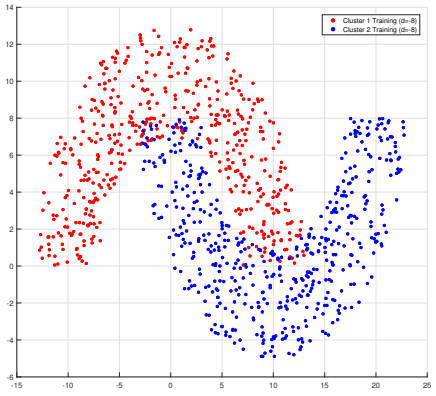

Fig. 3. Visualization of Clusters with separation of -8 units

neural net toolbox such as for basic back propagation we have used 'traingd' (Gradient Descend Back Propagation)[4] [5], for back propagation with momentum we have used 'trainrp' (Resilient Back Propagation Algorithm)[6] [7]and for Levenberg Marquardt we have used 'trainlm' (Levenberg Marquardt)[8] [9]. In this case, we have set the 75% of data is training data and 25% is used for cross validation. Furthermore, We have trained network with different learning rates having values viz. 1.6,0.9 and 0.2 for all three separation distances of 2, -4, -8.

C. Testing Data

Here, we have generated 1000 testing data points for separation distant between clusters as d=2, d=-4 and d=-8. Here, visualization of trained neural network for different settings and parameter values is done in terms of decision boundary and learning curves. Lets Consider each algorithm for network training and its response to the testing data.

D. Experiment I

1) Basic Back Propagation: Now consider simple back propagation algorithm, we are using 'traingd'(Gradient Descend Algorithm)[5] available in MATLAB Neural Net toolbox [10]. we have set the number of units in hidden layer to 3. Here, we can visualize the response of single hidden layer neural network trained with different learning rates in fig. 4:9.

2) Back Propagation with Momentum: Here, We are using resilient back propagation algorithm available in MATLAB Neural Net tool box. Important parameter that we have to focused on is momentum rate of back propagation algorithm[6] [7]. Due to addition of momentum rate, there is attenuation in oscillation of gradient descent [6] . thus, we have tested the output of neural network trained with back propagation algorithm having momentum rate of "0.9" and response can be visulaized in terms of decision boundary in fig. 4:9

3) Levenberg-Marquardt Back Propagation: Levenberg Marquardt Back Propagation is optimal algorithm for least square estimation when we deal with non linear decision boundaries[8]. as per the given problem, we have used 'trainlm' (Levenberg Marquardt) [9] parameter available in MATLAB neural net tool box. Response of neural net with Levenberg-Marquardt algorithm can be in fig. 4:9.

E. Experiment II

Initially we trained neural network with single hidden layer with three different back propagation algorithm and learning rates. however, in this part of experiment, we are restricting data set to cluster with separation distance of -8.Untill now, we have not changed the number of units (neurons) in hidden layers which indirectly was restriction on performance of neural network to complext data set (d=-8). But by changing number of units in hidden layer to 2, 5 and 11 we can visualize the change and improvement in performance using learning curves and decision boundaries shown in fig. 10 and fig. 11

III. Results

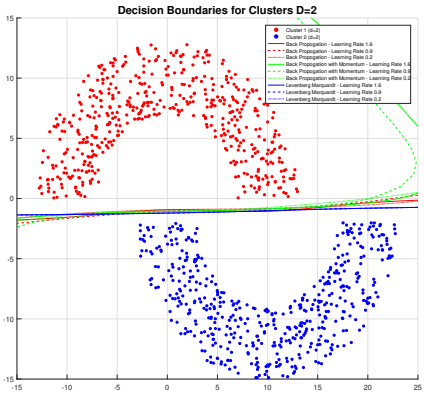

Fig. 4. Response of Neural Network in terms of decision boundary for cluster with separation distance of 2 units

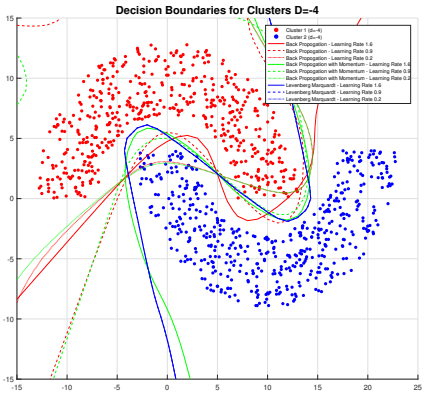

Fig. 5. Response of Neural Network in terms of decision boundary for cluster with separation distance of -4 units

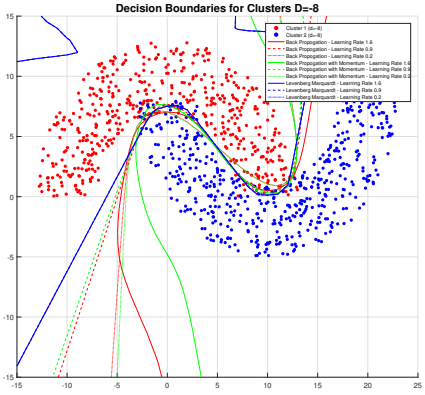

Fig. 6. Response of Neural Network in terms of decision boundary for cluster with separation distance of -8 units

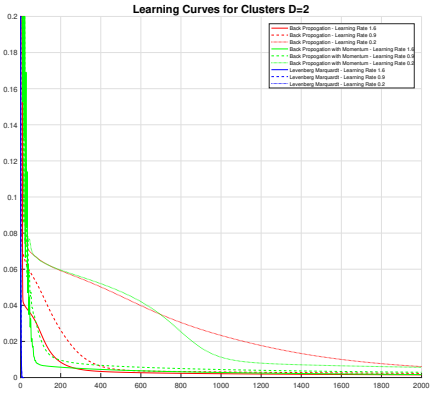

Fig. 7. Performance of three different algorithms for three different learning rates in terms of Learning Curve for data set with separation distance of 2 units

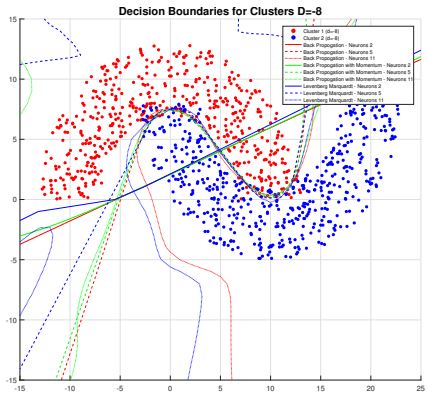

Fig. 10. Response of Neural Network in terms of decision boundary and different numbers of neurons for cluster with separation distance of -8 units

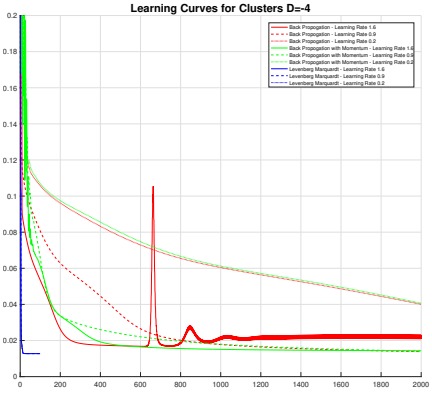

Fig. 8. Performance of three different algorithms for three different learning rates in terms of Learning Curve for data set with separation distance of 2 units

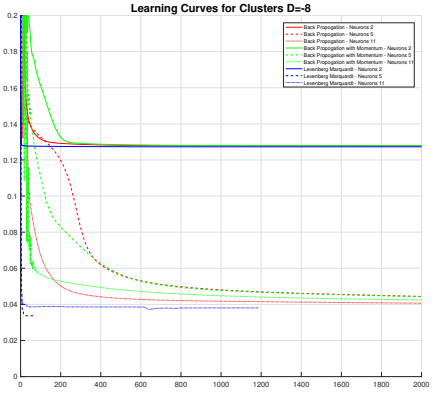

Fig. 11. Performance of neural network with threedifferent algorithms and three different number of hiddenneurons in terms of Learning Curve for data set withseparation distance of -8 units

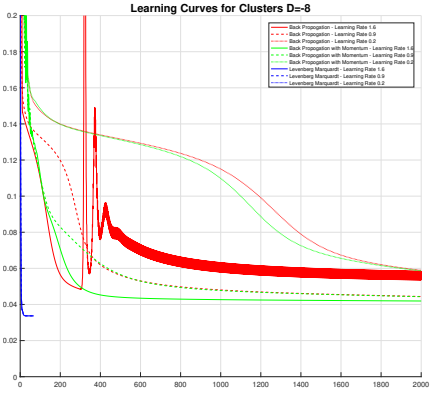

Fig. 9. Performance of three different algorithms for three different learning rates in terms of Learning Curve for data set with separation distance of 2 units

		D= 2 Units		
		Percentage of Correct Classification		
	Learning Rate	Cluster 1	Cluster 2	Total
GD	1.6	50.00	50.00	100.00
	0.9	50.00	50.00	100.00
	0.2	50.00	50.00	100.00
GDM	1.6	50.00	50.00	100.00
	0.9	50.00	50.00	100.00
	0.2	50.00	50.00	100.00
LM	1.6	50.00	50.00	100.00
	0.9	50.00	50.00	100.00
	0.2	50.00	50.00	100.00

Fig. 12. Confusion Matrix Summary of Clusters with Separation distance d= 2 for different conditions

		D= -4 Units		
		Percentage of Correct Classification		
	Learning Rate	Cluster 1	Cluster 2	Total
GD	1.6	49.7	49.9	99.6
	0.9	49.9	49.9	99.8
	0.2	48.2	48.8	97
GDM	1.6	50.00	49.9	99.9
	0.9	50.00	49.9	99.9
	0.2	47.8	48.8	96.6
LM	1.6	49.9	49.9	99.8
	0.9	49.9	49.9	99.8
	0.2	49.9	49.9	99.8

Fig. 13. Confusion Matrix Summary of Clusters with Separation distance d= -4 for different conditions

		D= -8 Units		
		Percentage of Correct Classification		
	Learning Rate	Cluster 1	Cluster 2	Total
GD	1.6	46.5	48.7	95.2
	0.9	46.9	47.6	94.5
	0.2	47.5	47.1	94.6
GDM	1.6	46.9	47.9	94.8
	0.9	46.2	48.0	94.2
	0.2	47.5	47.1	94.6
LM	1.6	46.7	47.8	94.5
	0.9	46.7	47.8	94.5
	0.2	46.7	47.8	94.5

Fig. 14. Confusion Matrix Summary of Clusters with Separation distance d= -8 for different conditions

IV. Conclusion

From this project, we understood how to implement back propagation algorithm and the merits and demerits of different algorithms such as back propagation with momentum and levenberg marquardt algorithm. A three layer neural network with enough neurons are enough to solve this classification problem but for even more complicated data sets which are not separable by a linear classifier, a more different sophisticated approach is required.

References

[1] H. Larochelle, "Back propagation," date last accessed 15-oct-2017. [Online]. Available: https://www.youtube.com/watch?v=_KoWTD8T45Q&list=PL6Xpj9I5qXYEcOhn7TqghAJ6NAPrNmUBH&index=13

[2] A. Jagannatam, "Mersenne twister a pseudo random number generator and its variants," 2017.

[3] MATLAB, "Control random number generation," date last accessed 15-oct-2017. [Online]. Available: https://www.mathworks.com/help/matlab/ref/rng.html?searchHighlight=rng&s_tid=doc_srchtitle

[4] P. Baldi, "Gradient descent learning algorithm overview: a general dynamical systems perspective," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 182–195, Jan 1995.

[5] MATLAB, "Gradient descend back propagation algorithm," date last accessed 15-oct-2017. [Online]. Available: https://www.mathworks.com/help/nnet/ref/traingd.html

[6] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the rprop algorithm," in *IEEE International Conference on Neural Networks*, 1993, pp. 586–591 vol.1.

[7] MATLAB, "Resilient back propagation algorithm," date last accessed 15-oct-2017. [Online]. Available: https://www.mathworks.com/help/nnet/ref/trainrp.html

[8] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963. [Online]. Available: https://doi.org/10.1137/0111030

[9] MATLAB, "Levenberg-marquardt algorithm," date last accessed 15-oct-2017. [Online]. Available: https://www.mathworks.com/help/nnet/ref/trainlm.html

[10] Matlab, "Multilayer neural networks and back propagation," (Date last accessed 15-oct-2017). [Online]. Available: https://www.mathworks.com/help/nnet/ug/multilayer-neural-networks-and-backpropagation-training.html