Comparison and Analysis of Multiplicative neuron and Multilayer Perceptrons using three different datasets

Pankaj Kumar Kandpal, Ashish Mehta
Department of Computer Science
Kumaun University, Nainital,UK
kandpalzee@gmail.com, ashishmehta19@gmail.com
Mobile No. +91-9412097979, +91-9411132201

Abstract-In this paper, Multiplicative Neuron Models is used for classification of nonlinear problems using three different datasets, namely; Iris, Mammographic dataset and Brest Cancer Original. The conventional neuron model Multi Layer Perceptrons (MLP) is used for comparative analysis with Multiplicative Neuron Model. It is found that Multiplicative neuron model with single neuron is sufficient for classification than the conventional neuron network which require number of neurons in different hidden layers. For comparative analysis of three models, various parameters of Artificial Neural Network like learning rate, execution time, number of iteration, time elapse in training, mean square error, etc., are considered. After comparing various performance evaluation parameters, it is found that execution time, number of iteration, time elapse in training is minimum in case of Multiplicative neuron model. On the basis of results from the study, it is observed that performance of Multiplicative neuron model better than the MLP for classification.

Keywords- Multiplicative Neuron, Multilayer Perceptron, Classification, Iris, Mammographic Mass, Brest Cancer Original, analysis

I. Introduction

Artificial Intelligence is the branch of the computer science concerned with the study and creation of computer systems that exhibit some form of intelligence: System that learns new concepts and tasks, system that can reason and draw useful conclusion about the world around us, System that can understand a natural language or perceive and comprehend a visual sense, and system that perform other types of feats that require human types of intelligence [1]. The Artificial Neural Networks is one stream of Artificial Intelligence.

Artificial Neural Networks is the mathematical model of biological neurons. Although all these models were primarily inspired from biological neuron. Every processing element of model bear a direct analogy to the actual constituents of biological neuron and hence is termed as artificial neuron[3]. After giving the so many contributions by plenty of researchers still a gap between philosophies used in neuron models for neuroscience studies and those used for artificial neural networks (ANN). Some of neural network models exhibit a close correspondence with their biological counterparts whiles other far away with their counterparts. It is being contributed by several scientists that gap between biology and mathematics can be minimized by investigating the learning capabilities of biological neuron models for use in the applications of classification, time-series prediction, function approximation, etc. In this paper, it is being taken a single Multiplicative Neuron (MNM), compared with Multilayer perceptron, after analyzing the results, it can be reached to conclusion that which one is the better model in context of various parameters of Artificial Neural Network like Learning Rate, Execution Time, Number of Iterations, Time Elapse in training etc.

In the initial study of Artificial Neural networks, the first artificial neuron model was proposed by McCulloch and Pitts [7] in 1943. They developed this neuron model based on the fact that the output of neuron is 1 if the weighted sum of its inputs is greater than a threshold value, and 0, otherwise. In 1949, Hebb [8] proposed a learning rule that became initiative for ANNs. He postulated that the brain learns by changing its connectivity patterns. Widrow and Hoff [9] in 1960 presented the most analyzed and most applied learning rule known as least mean square rule. Later in 1985,Widrow and Sterns [10] found that this rule converges in the mean square to the solution that corresponds to least mean square output error if all input patterns are of same length. A single neuron of the above and many other neuron types proposed by several scientists and researchers are capable of linear classification [11]. Further many scientists have used single neuron model for nonlinear classification. Multiplicative neural networks learning methodology has been provided by the authors in [12, 13]. It has been experienced that increasing number of terms in the high order expression, it is exceedingly difficult to train a network of such neurons. That is the main inspiration to choose simpler model for the high order expression with a well defined training procedure based on the back propagation. Section II, describes the models have been taken in the paper. Section III details about the datasets being taken in the paper, section IV discusses the results and section V provides concluding remarks of the paper.

II Material and Methods

A. Biological Neural Model

The elementary nerve cell, called a neuron, is the fundamental building block of the biological neural network. A typical cell has three major regions: the cell body, which is also called the soma, the axon, and the dendrites. Dendrites form a dendritic tree, which is a very fine bush of thin fibers around the neuron's body. Dendrites receive information from neurons through axons-long fibers that serve as transmission lines. An axon is a long cylindrical connection that carries impulses from the neuron. The end part of an axon splits into a fine arborization. Each branch of it terminates in a small end bulb almost touching the dendrites of neighboring neurons. The axon-dendrite contact organ is called a synapse. The synapse is where the neuron introduces its signal to the neighboring neuron. The signals reaching a synapse and received by dendrites are electrical impulses. The interneuronal transmission is sometimes electrical but is usually effected by the release of chemical transmitters at the synapse. Thus, terminal boutons generate the chemical that affects the receiving neuron. The receiving neuron either generates an impulse to its axon, or produces no response [2].

A. Multiplicative Neuron Model

Only single neuron of this model is used for the classification task. In this model, aggregation function is based upon the multiplicative activities (Ω) at the dendrites, given in the Fig. 2.

$$\mathbf{\Omega}(\mathbf{x}, \boldsymbol{\theta}) = \prod_{i=i}^{n} (w_i x_i + b_i) \qquad (1)$$

In above given generalized equation Eq. (1) of Multiplicative neuron model, Ω is a multiplicative operator with weights wi, inputs xi and biases bi. In the given equation ∏ (production) is being used instead of ∑ summation. It is investigated the complexity of computing and learning for multiplicative neuron. The author in [18] used single Multiplicative neuron for time series prediction. In particular, we derive upper and lower bounds on the Vapnik- Chervonenkis (VC) dimension and pseudo dimension for various types of networks with multiplicative units [23-25]. In the Internal architecture and computation methods are different but the procedure of training; testing and prediction are same as used in Multi-Layer Perceptron model. Unlike the higher-order neuron, this model is simpler in terms of its parameters and one does not need to determine the monomial structures prior to training of the neuron model. Multiplicative Neuron Model is used for problems with high nonlinearity and it can be trained easily.

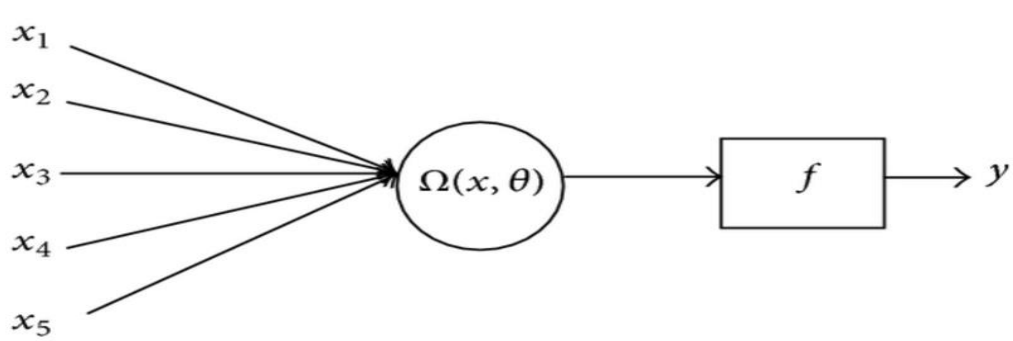

Fig.1 A Generalized Single Neuron.

B. Learning for Multiplicative Neuron Model

Learning algorithm for the artificial neural network is a optimization techniques. In this paper, authors used most popular back propagation learning algorithm[12]. The simplicity of back propagation methods make it convenient for the models to be used in different situation, unlike the high order neuron model, which is difficult to train and is susceptible to combinatorial explosion of terms. A simple gradient descent rule, using a norm-squared error function, is described by the following set of equations[13].

A. Forward Pass

$$S = \prod_{i=1}^{n} (w_i x_i + b_i)$$

$$y = \frac{1}{1 + e^{-neth}}$$

B. Backward Pass

$$E = \frac{1}{2N} \sum_{P=1}^{N} (y^P - t^P)^2 \tag{2}$$

$$\frac{de}{dy} = (y - t) \tag{3}$$

$$\frac{dy}{dnet} = y\,(1 - y) \tag{4}$$

$$\frac{dnet}{dw_i} = \frac{net}{(w_i x_i + b_i)}\, x_i \tag{5}$$

$$\frac{de}{dw_i} = \frac{de}{dy}\, \frac{dy}{dnet}\, \frac{dnet}{dw_i} \tag{6}$$

$$\frac{de}{dw_i} = (y - t) * y * (1 - y) * \frac{net}{(w_i x_i + b_i)}\, x_i \tag{7}$$

$$\frac{de}{db_i} = \frac{de}{dy}\, \frac{dy}{dnet}\, \frac{dnet}{db_i} \tag{8}$$

$$\frac{dnet}{db_i} = \frac{net}{(w_i x_i + b_i)} \tag{9}$$

$$\frac{de}{db_i} = (y - t) * y * (1 - y) * (w_i x_i + b_i) \tag{10}$$

$$w_{i(new)} = w_{i(old)} - \frac{de}{dw_i} * eta \quad (11)$$

$$b_{i(new)} = b_{i(old)} - \frac{de}{db_i} * eta \quad (12)$$

In the Eq. (2) simple steepest descent methods applied to reduce deviation between actual values (y) and target values (t). Where eta (η) is learning rate which can be assigned a value on the heuristics basis. Eq. (3-8) weights modifying process to minimize the error. On the other hand, Eq. (8-10) biases modifying.Weights and biases, parameters update rules are exhibited in the Eq. (11-12) after every epoch. Using the back propagation learning method, it being solved some most popular classification problem in next section.

C. Multilayer Perceptron Model

It is a very well known conventional model. The adapted perceptrons are arranged in layers and so the model is termed as multilayer perceptron. This model has three layers: an input layer, an output layer, and a layer in between, not connected directly to the input or output, and hence called the hidden layer. For the perceptrons in the input layer, linear transfer function is used, and for the perceptrons in the hidden layer and the output layer, sigmoidal or squashed-S functions is used. The input layer serves to distribute the values they receive to the next layer and so does not perform a weighted sum or

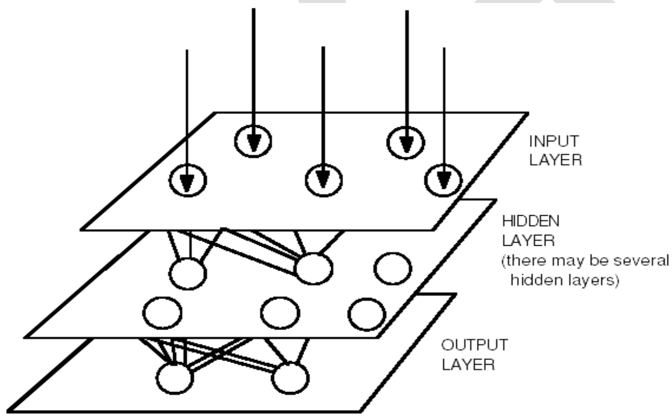

FIG.2 MULTILAYER NEURAL NETWORK.

shown in Figure1. Many capabilities of neural networks, such as nonlinear functional approximation, learning, generalization etc. are, in fact, due to nonlinear activation function of each neuron. Sigmoid Activation Function is given below:

A. Forward Pass

$$neth_j = \sum_{i=1}^{ni} (w_{ji} * x_i + b_i) \quad (13)$$

$$h_j = \frac{1}{1+e^{-neth_j}} \quad (14)$$

$$nety_k = \sum_{j=1}^{nh} (w_{kj} * h_j) \quad (15)$$

$$y_k = \frac{1}{1+e^{-nety_k}} \quad (16)$$

B. Backward Pass

$$dy_k = (y_k - t_k) \tag{17}$$

$$E = \frac{1}{2}\sum_{k=1}^{N} (d_{yk})^2 \tag{18}$$

$$h_{i(new)} = h_{i(old)} - \frac{de}{dh_i} * \eta \tag{19}$$

$$y_{k(new)} = y_{k(old)} - \frac{de}{dy_i} * \eta \tag{20}$$

The activity of neurons in the input layers represents the raw information fed into the network; the activity of neurons in the hidden layer is determined by the activities of the neuron in the input layer and connecting weights between input and hidden units. Similarly, the activity of the output units depends on the activity of neurons in the hidden layer and the weight between the hidden and output layers. This structure is interesting because neurons in the hidden layers are free to conduct their own representation of the input. [2]

D. Non linear Classification Techniques

One way to classify data is to first create models of the probability density functions for data generated from each class. Then, a new data point is classified by determining the probability density function whose value is larger than the others. Linear discriminant analysis (LDA) is an example of such an algorithm. Linear Discrimination Analysis is a technique for linear classification. For nonlinear classification there are two well known classification techniques given below:

A. Artificial Neural Networks (ANN)

Artificial neural networks are often used to develop nonlinear classification boundaries. Reasons for their common use include their ease of application, their robustness to choices of parameter values, and their similarity to other nonlinear regression methods.

B. Support Vector Machine (SVM)

Conventional neural networks can be difficult to build due to the need to select an appropriate number of hidden units. The network must contain enough hidden units to be able to approximate the function in question to the desired accuracy. A primary motivation behind SVMs is to directly deal with the objective of good generalization by simultaneously maximizing the performance of the machine while minimizing the complexity of the learned model.

III. Data set used

A. Iris dataset

This classic data of Anderson and Fisher pertains to a four-input, three class classification problem. Iris is a species of flowering plants with showing flowers. Iris data base prepared by the Fisher in 1936, and perhaps the best known database to be found in the pattern recognition literature. In Fisher's Iris database among the several species, three species of Iris plants setosa, versicolor, virginica are selected. In this dataset, Fisher taken four attributes of Iris flowers, are petals length, Petal width & sepals length, sepal width (in centimeter; cm). The data set consists of 50 samples from each of three species of Iris (Setosa, Virginica and Versicolor). Therefore there are 150 instances in the dataset, which are collected on large number of Iris flowers. The Neural network will be trained to determine specie of iris plant for given set of petal and sepal width and length. Fisher's Iris data base is available in Matlab (load fisheris) and in Internet [20].

B. Mammographic Mass dataset

Matthias Elterand and Dr. Rudiger Schulz Wendtland efforts have made easy approachable Mammographic Mass dataset to all researchers. Mammography is the most effective method for breast cancer screening available today. However, the low positive predictive value of breast biopsy resulting from mammogram interpretation leads to approximately 70% unnecessary biopsies with benign outcomes. To reduce the high number of unnecessary breast biopsies, several computer-aided diagnosis (CAD) systems have been proposed in the last years. These systems help physicians in their decision to perform a breast biopsy on a suspicious lesion seen in a mammogram or to perform a short term follow-up examination instead.

This data set can be used to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age [21]. The mammographic problem deal with the classification between benign (0) and malignant (1),

C. Brest Cancer Wisconsin dataset

This dataset contains cases from a study that was conducted at the University of Wisconsin Hospitals, Madison, about patients who had undergone surgery for breast cancer. The task is to determine if the detected tumor is benign (2, after normalization 0.1) otherwise malignant (4 after normalization 0.9). To assess the data to classification process, the first attribute of the original data set (the sample code number) has been removed in this version. The dataset have 9 attributes, (excluded first attribute "Sample code number") and one class attribute. Total number of 699 instances where 16 instances having missing values. For the sake of clearity, author has removed the missing value instances. There are total 458 instances of Benign(65.5%) and 241 instances of Malignant(34.5%)[22].

IV. Results and discussion

Two classification models have been selected and well known datasets has been taken. The experimental parameters show that multiplicative neuron and multi layer perceptron (MLP) has been trained by using three well known data sets Iris, Pima Indian Diabetes and Brest Cancer original. Whole data set is been used for training and a small subset is been used for testing. The training is continued until the network going on improving. When network trained, the training is stopped. Training can be stopped in another condition when training goal in term of MSE is met or given iteration (epoch) are completed. In both classification problems the dataset has been preprocessed. The dataset has been normalized between 0.1 and 0.9. For each simulation the minimum configrational requirement of the computer is Pentium 4 processor with 1.8 GHz and 512 MB RAM.

A. Iris Problem

The authors compare the performance of multiplicative neural networks (MNM) with that of multi layer perceptrons (MLP). For this objective, the MLP taken with three layers, with multiple hidden neurons. The Fig. 3 shows the mean square error (MSE) versus number of epochs (Iteration) curve for training with multiplicative neuron model (MNM) and MLP while dealing with the IRIS flower classification problem. It is cleared with the curve that Multiplicative neuron model with single neuron, learns easily and minimize the error early in comparison to multilayer perceptron. Table 2 exhibit the comparison between MLP and MNM in terms of deviation of actual outputs from corresponding targets. It can be seen with the help of Table 1 that the performance of MNM is better than that of MLP. From Table 1, it is observed that the training time required by MNM is much less than MLP. It means that a single neuron in MNM is capable to learn IRIS relationship almost four times faster than MLP with 18 hidden neurons. Table shows the comparison of training and testing performance with MLP and MNM, while solving the IRIS classification problem.

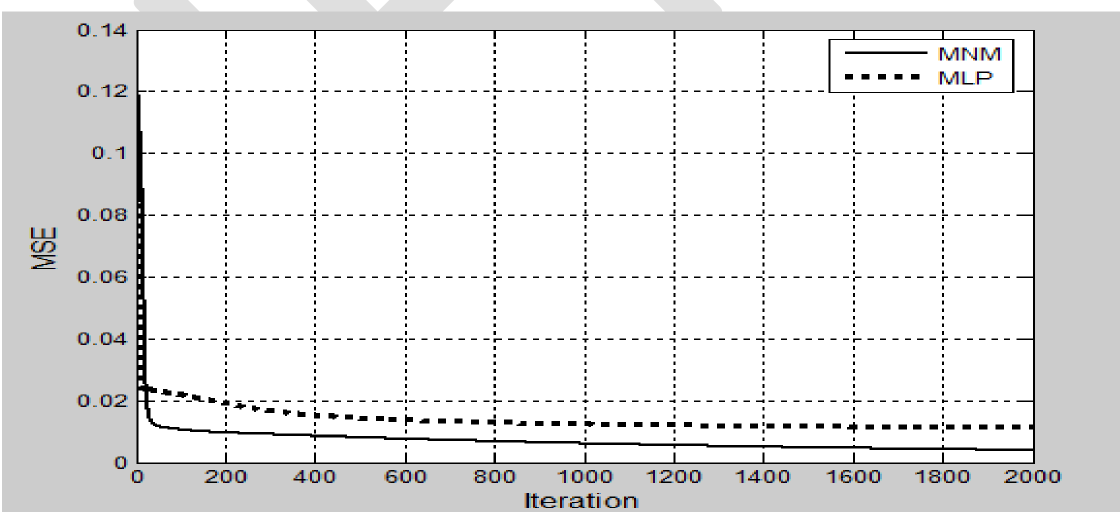

Fig.3 Mean square error vs. iteration for training for IRIS problem.

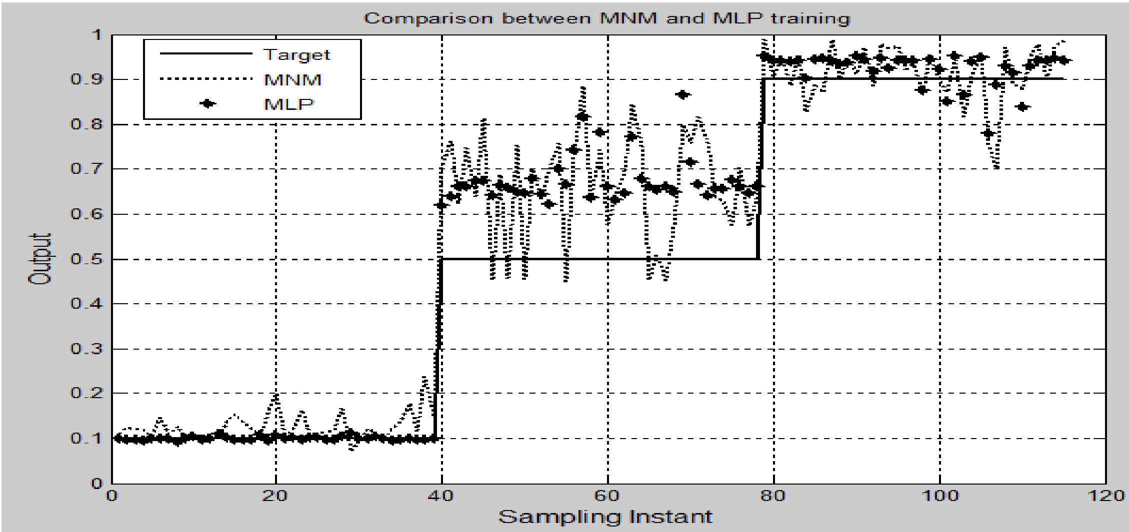

Fig.4 Comparison between MNM and MLP training.

Table.1

Comparison of training and testing performance for IRIS problem

S.No.	Parameter	MNM	MLP
1	Training goal, in term of MSE (error check)	0.0001	0.00001
2	Iteration needed	500	4000
3	Training time in seconds	19	92
4	testing time in seconds	0	0
5	MSE for training data	0.0074	0.0058
6	MSE for testing data	0.0054	0.0033
7	RSME for training	0.858	0.0763
8	RMSE for testing	0.755	0.0572
9	Correlation coefficient	0.9682	0.9699
10	percentage of miss classification	5%	5%
11	number of neurons	1	23
12	learning late (Ƞ)	1.8	2.1

Table 2, shows the input values and equivalent outputs values of both models. Figure 4 and figure 5 shows the training and testing results of Iris datasets. The figures show that some marginal overlapping all three classes are clearly separable with each others.

Table.2

Comparison of Output of MNM and MLP for IRIS Problem

Input	Target	Actual Output with MNM	Actual Output with MLP
0.678, 0.467, 0.656,0.833	0.9	0.8818	0.92848
0.544, 0.567, 0.724,0.867	0.9	0.95001	0.95041
0.167, 0.567, 0.154, 0.167	0.1	0.097149	0.092123
0.411, 0.7, 0.195, 0.167	0.1	0.15513	0.10947
0.256, 0.2, 0.412, 0.4	0.5	0.66848	0.64186
0.389, 0.267, 0.493, 0.433	0.5	0.56815	0.6241

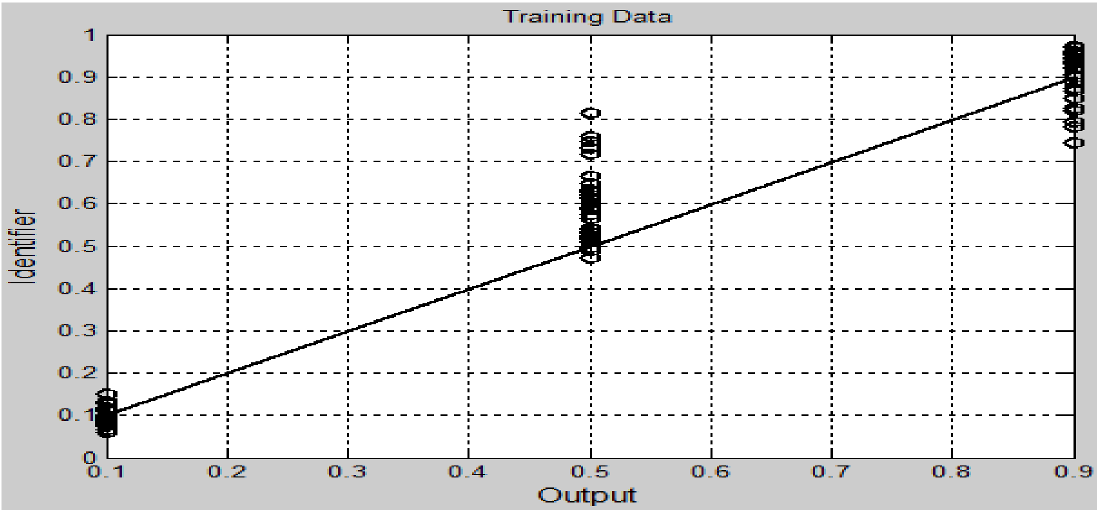

Fig.5 Training results for IRIS Problem.

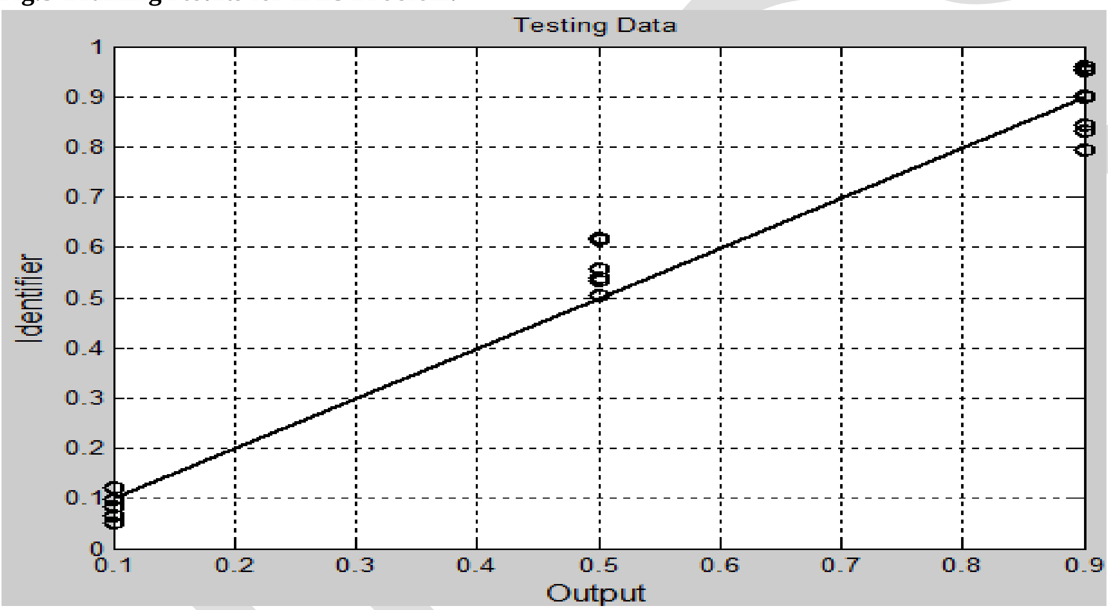

Fig.6 Testing results for Iris Problem.

B. Mammographic Mass Dataset

The authors compared the performance of multiplicative neural networks (MNM) with that of multi layer perceptrons (MLP). Depicted in the Fig. 7 that MSE versus epochs curves for training with MNM and MLP while dealing the problem. Where MLP takes 4000 epochs to learn the pattern, on the other hand, MNM takes only 1000 epochs. From the Table 3, it is observed that the training time required by MLP is much more than MNM. It means that a single neuron of MNM is capable to learn mammographic mass pattern, where MLP model required 31 neurons.

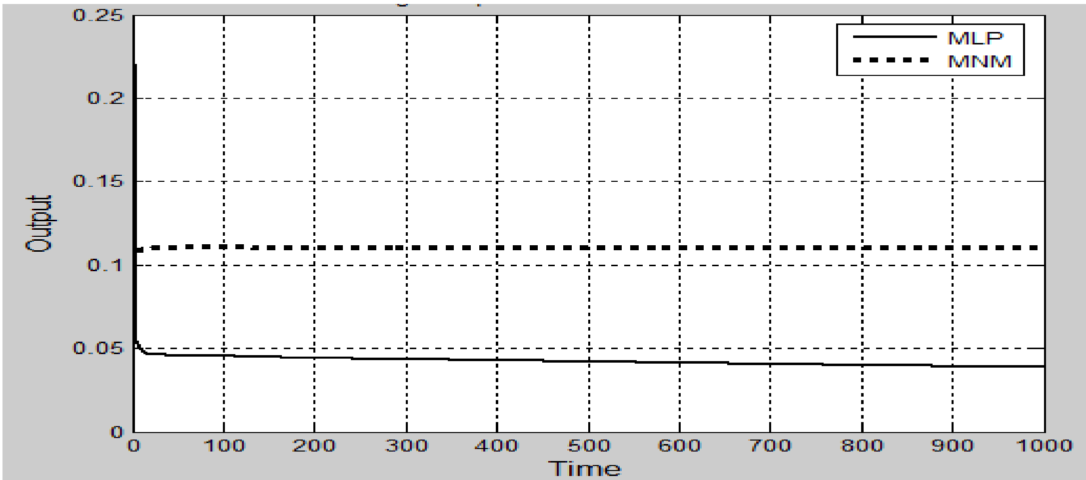

Fig.7 MSE vs. iteration for training for Mammographic Mass problem.

Table.3

Comparison of training and testing performance for Mammographic Mass problem

S.No.	Parameter	MNM	MLP
1	Training goel, in term of MSE (error check)	0.0001	0.00001
2	Iteration needed	1000	4000
3	Training time in seconds	106	214
4	testing time in seconds	0.18	0.02
5	MSE for training data	0.0606	0.0363
6	MSE for testing data	0.0663	0.0442
7	RSME for training	0.2462	0.1904
8	RMSE for testing	0.2575	0.2103
9	Correlation coefficient	0.5711	0.7509
10	percentage of miss classification	23%	13%
11	number of neurons	1	31
12	learning late (Ŋ)	0.77	0.85

Table 4 exhibits the comparison between MNM and MLP in terms of deviation of actual output from coresponding targets. In context of Mammographic Mass Problem results are not very good as IRIS problem or Brest Cancer original but shows a clear classification and reveal deference between the MLP and MNM models.

Table.4

Comparison of Output of MNM and MLP for Mammographic Mass Problem

Input	Target	Actual Output with MNM	Actual Output with MLP
0.172, 0.5,0.1,0.9,0.633	0.9	0.83568	0.94885
0.172,0.694,0.1,0.7,0.633	0.9	0.63788	0.87049
0.158,0.284,0.633, 0.1,0.633	0.1	0.14166	0.1139
0.158, 0.580, 0.366, 0.1, 0.366	0.1	0.15231	0.10555

Fig. 8 Training results for Mammographic Mass Problem.

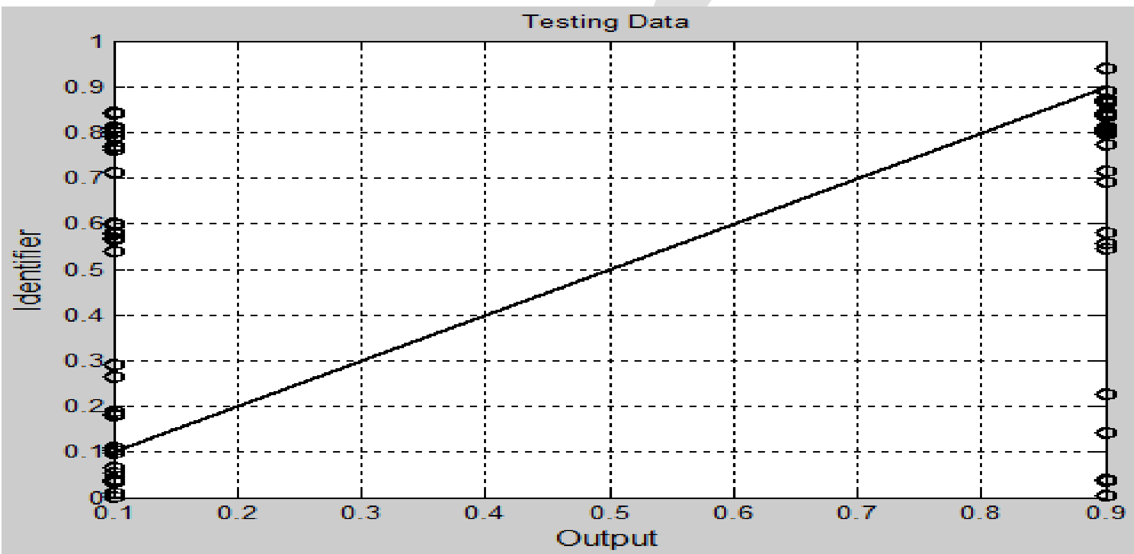

Fig.9 Testing results for Mammographic Mass Problem.

C. Brest Cancer winconsin Original Problem

Nine inputs Brest cancer winconsin dataset donate by Dr. William H. Wolberg to UCI. Table 5 shows the comparison of training and testing performance with MLP and MSN while solving the Brest cancer winconsin problem . It is observed from the table that the performance of the MSN model is Significantly better on this data set. It can be compared with the other tables (Table 1&3) and easily found that Brest cancer problems results are far better than Iris and Pima Indian dataset. The classification results depicted in the Fig.11 and Fig. 12, from the figures it can be seen that the proposed model performance in better than that of MLP. The single neuron model capable to classify the pattern in only 400 epochs where the MLP model with total 26 neurons learnt in 1500 iterations. It is become clear from the MSE vs. epoch curve in Fig.10. the proposed model resolve the problem with 93% of success.

Table.5

Comparison of training and testing performance for Brest Cancer winconsin

S.No.	Parameter	MNM	MLP
1	Training goal, in term of MSE (error check)	0.0001	0.00001

2	Iteration needed	400	1500
3	Training time in seconds	77.72	89.23
4	testing time in seconds	0.02	0.01
5	MSE for training data	0.185	0.0012
6	MSE for testing data	0.0128	0.0004
7	RSME for training	0.1363	0.034
8	RMSE for testing	0.1132	0.0202
9	Correlation coefficient	0.8699	0.9924
10	percentage of miss classification	7%	0%
11	number of neurons	1	26
12	learning late (Ƞ)	2.1	2.1

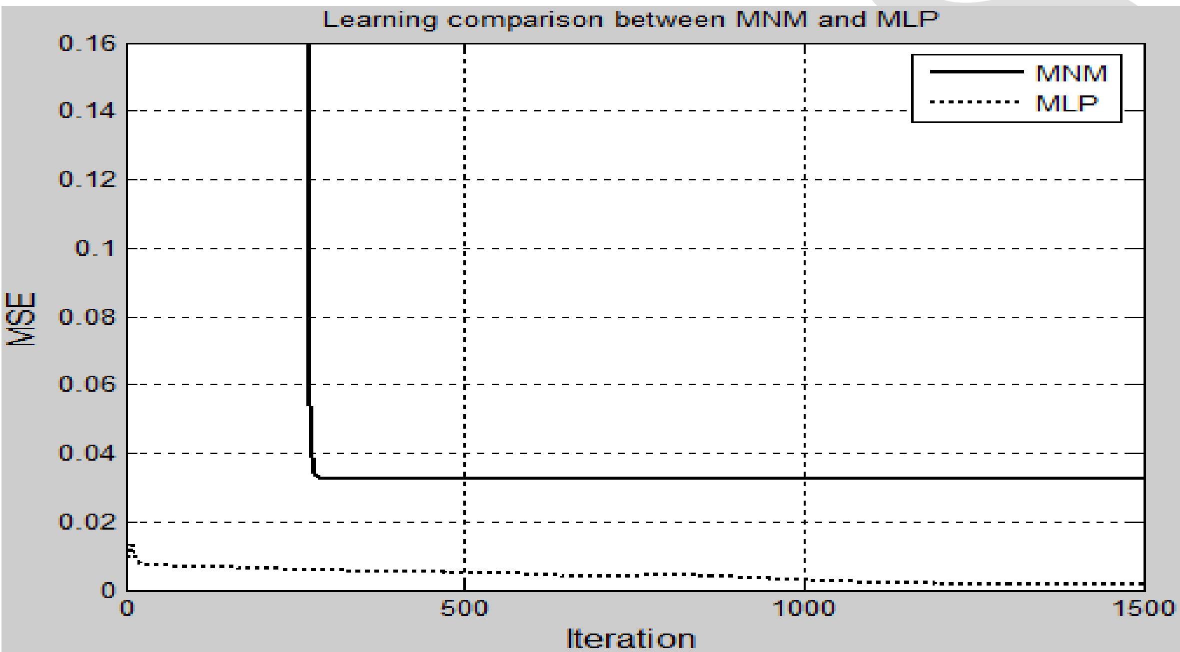

Fig.10 MSE vs. iteration for training of Brest Cancer Winconsin problem.

Table 6, shows the input values and equivalent outputs values of both models. The table show the deviation of output values from the actual values. For the same input values the corresponding output values of both models. It can be seen from the table that there is a little bit difference between both models despite of huge difference in the participating neurons.

Table.6

Comparison of Output of MNM and MLP for Brest Cancer Winconsin

Input	**Target**	**Actual Output with MNM**	**Actual Output with MLP**
0.72222,0.9,0.9,0.72222,0.63333,0.9,0.81111,0.63333,0.1	0.9	0.808	0.8703
0.45556,0.27778,0.27778,0.27778,0.18889,0.27778,0.36667,0.36667,0.1	0.9	0.78438	0.90227
0.36667,0.1,0.1,0.27778,0.18889,0.1,0.27778,0.1,0.1	0.1	0.072989	0.096094
0.1,0.1,0.1,0.1,0.18889,0.9,0.27778,0.1,0.1	0.1	0.09421	0.14736

Fig.11 Training results (MLP) for Brest Cancer Winconsin Problem.

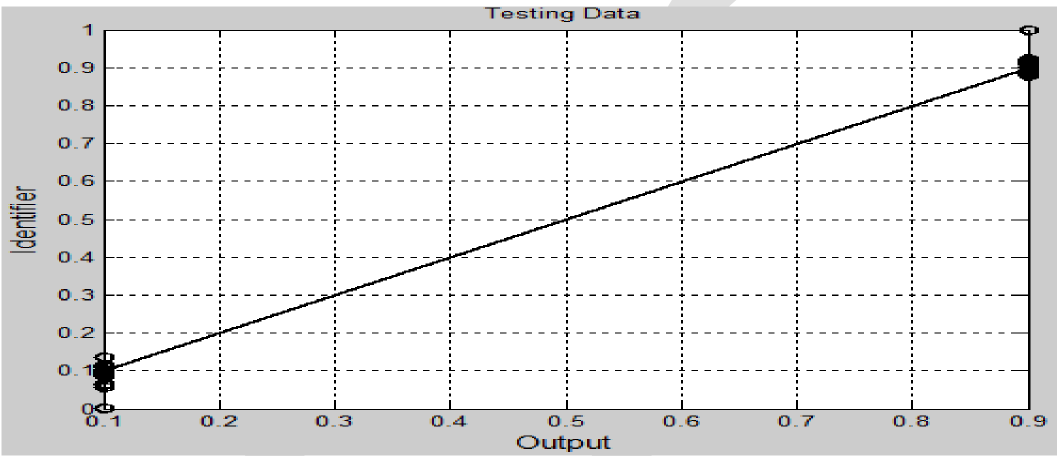

Fig.12 Testing results for(MLP) Brest Cancer Winconsin Problem.

V. Conclusion

After the finding the training and testing results of MNM and MLP using both popular IRIS, Mammographic Mass and Brest Cancer winconsin classification problem, it can be seen in simulation results that that single multiplicative neuron capable of performing classification task as efficiently as a multilayer perceptron with many neurons. By seeing the misclassification rate, Iris and Brest Cancer problem cases, learning of single Multiplicative neuron is good and in case of Mammographic Mass problem, it is considerable than that of multilayer percedptron. Eventually it is observed that training and testing time in case of MNM are significantly less as compared with MLP, in both problems. R. N. Mishra and et. al's finding on time series prediction, supports our study[18]. Therefore it is justified that the proposed model is better than MLP. Future scopes of this work, incorporation of the Multiplicative neuron in different networks and analysis the learning capabilities for classification, regression, function approximation, and time series prediction.

Acknowledgment

We would like to thank to Dr. William H. Wolberg, Wisconsin Hospitals, Madison, for providing Brest Cancer dataset. We would like to thanks to Matthias Elterand Dr. Rudiger Schulz- Wendtland for making easy approachable Mammographic Mass dataset to all. And again we would like to pay out thanks to R A Fisher and University of California database Archive.

REFERENCES:

[1]Dan W Patterson "Inroduction to Artificial Intelligence and Expert System" Prentice Hall of India Private ltd 2005.
[2] Jacek M. Zurada "Artificial Neural System" West Publishing Company.
[3] S. Rajasekaran, G.A. Vijayalakshmi Pai, "Neural Networks, Fuzzy Logic, and Genetic Algorithms", PHI Learning Pvt. Ltd.
[4] M. Balasubramanian, M. Fellows and V. Raman, unpublished
[5] Abhishek Yadav "Dynamical aspects and learning in Biological Neuron Models" department of electrical engineering IIT Kanpur June 2005
[6] W. J. Freeman, "Why neural networks dont yet fly: inquiry into the neurodynamics of biological intelligence", IEEE International Conference on Neural Networks, 24-27 July 1988, pp.1-7, vol.2, 1988.
[7] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", Bulletin of Mathematical Biophysics, vol.5, pp. 115-133, 1943.
[8] D. Hebb, "Organization of behavior", John Weiley and Sons, New York, 1949.
[9] B. Widrow and M. E. Hoff, "Adaptive switching circuits", IREWESCON Connection Recors, IRS, New York, 1960.
[10] B. Widrow and S. Steams, "Adaptive signal processing", Prentice-Hall, Englewood Cliffs, NJ., 1985.
[11] M. Sinha, D.K. Chaturvedi and P.K. Kalra, "Development of flexible neural network", Journal of IE(I), vol.83, 2002.
[12]D. E. Rumelhart, G.E. Hinton, R.J. Williams, learning internal representations by error back propagations, in: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press,1986.
[13] D. Li, Hirasawa, J. Hu, J. Murata, Multiplicative unit in feed forward neural networks and its training, in proceeding of Ninth International conference proceeding ICONP'02, Singapore, 2002.
[14] Deepak Mishra, Abhishek Yadav, & Prem K. Kalra, A Novel Neural Network Architecture Motivated by Integrate-And-Fire Neuron Model Department of Electrical Engineering Indian Institute of Technology Kanpur, India
[15] Deepak Mishra, Abhishek Yadav, & Prem K. Kalra, A Novel Multiplicative Neural Network Architecture Motivated by Spiking Neuron Model Department of Electrical Engineering Indian Institute of Technology Kanpur, India
[16] R. N. Yadav, V. Singh and P. K. Kalra, "Classification using single neuron",Proceedings of IEEE International Conference on Industrial Informatics, 2003, pp.124-129, 21-24 Aug. 2003, Canada.
[17] A. Yadav *, D. Mishra, R.N. Yadav, S. Ray, P.K. Kalra, Time-series prediction with single integrate-and-fire neuron, Science Direct, Applied Soft Computing 7 (2007) 739–745
[18] R N Yadav, P K Kalra, S John, Time series prediction with Single Multiplicative Neuron Model, ,Science Direct, Applied Soft Computing 7 (2007) 1157–1163
[19] Deepak Mishra, Abhishek Yadav, Sudipta Ray,Levenberg-Marquardt Learning Algorithm for Integrate-and-Fire Neuron Model, IIT Kanpur.
[20] http://archive.ics.uci.edu/ml/datasets/Iris.
[21] https://archive.ics.uci.edu/ml/datasets/Mammographic+Mass
[22]https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/
[23] Peter Dayan and L F Abbott, Theoretical Neuroscience "Computational and Mathematical modeling of Neural System, MIT Press Cambridge, London.
[24] Christof Koch, Biophysics of Computation " information processing in single neuron", Oxford University Press.
[25] N. sauer,on the density of family of sets, J. Comb. theory(A) 13 (1972) 145-147.
[26] D Haussler, P M Long, A generalization of Sauer's Lemma, J. Comb. theory 71 (1995) 219-240.
[27] P Auer, P M Long, W. Maass, G J Woeginger, On the complexcity of function learning, Mac.Learn. 18(1995) 189-236.