A New Technique to solve local Minima problem with large number of hidden nodes on Feed Forward Neural Network

[1]Kuldip Vora, [2]Shruti Yagnik

[1] M. Tech. Scholar, [2]Lecturer
Department of Computer Engineering, Gujarat Technological University, Gujarat, India
[1]kuldipiscool@gmail.com ,[2]shruya@gmail.com

Abstract **-The Back-propagation (BP) algorithm is a well-known representative of all iterative gradient descent algorithms used for supervised learning in neural networks. It is its simplicity that attracts researchers and so that, many improvements and variations of the BP learning algorithm have been reported to beat its limitations such as slow convergence rate and convergence to the local minima. It is extensively used in many applications. In this paper we propose an algorithm that shows that by having structural changes we can still improve and can solve local minima problem with problems having large number of hidden nodes. Results have been shown on some classical problems such as parity problem and also on soil data classification problem.**

Keywords **- Artificial neural network (ANN); Backpropagation algorithm(BPA); Hidden nodes; Target values; Local minima**

I. INTRODUCTION

A multilayer feed-forward neural network (MLFFNN) consists of an input layer, hidden layer and an output layer of neurons. Every node in a layer is connected to every other node in the neighboring layer. A FFNN has no memory and the output is solely determined by the current input and weights values. The training of FNN is mainly undertaken using the back-propagation (BP) based learning. Back-Propagation Neural Network (BPNN) algorithm is the most popular and the oldest supervised learning multilayer feed-forward neural network algorithm proposed by Rumelhart, Hinton and Williams [1]. It is built on high mathematical foundation and has very good application potential such as to pattern recognition, dynamic modeling, sensitivity analysis, and the control of systems over time.

BPLAs use the gradient-decent search method to adjust the connection weights. The standard BP learning algorithm suffers the typical handicaps of all steepest descent approaches. Very *slow convergence rate* and the need for predetermined learning parameters limit the practical use of this algorithm. Many improved learning algorithms have been reported. These improvements achieve better convergence rates and for many purposes, they perform sufficiently. However, for applications which require larger scale MLPs, the known algorithms are still insufficient. Slow convergence and long training times are still the disadvantages often mentioned when neural networks are compared with other competing techniques [8].

At present, most research works on this problem are focused on the improvement of the learning algorithms. part from that , the slow convergence of BP-learning is caused not only by algorithms, but also by the structure of MLP. Therefore, it is difficult to solve this problem only by improving the learning algorithm [8]. The complex problem which has a large number of patterns needs as many hidden nodes as patterns in order not to cause a singular hidden output matrix [7]. As the number of hidden nodes increases the increase in local minima point's increases causes the algorithm to trap into and stuck. The exceeding number of hidden nodes can possibly make a lot of local minima.

The Main Aim of our work is to find some unique idea to solve this local minima problem for large number of hidden nodes. Here we have adopted divide and conquer strategy to train the network as shown in Fig. 1.

II. PROPOSED LEARNING ALGORITHM

The Proposed Learning Algorithm is based on one of the novel idea from the separate learning algorithm strategy [7]. Here the strategy for the proposed algorithm starts with a attention to only two layer network. Here as depicted in Fig.1 the proposed algorithm divides the network into two equal half one is INPUT-to-Hidden Layer and second is Hidden-to-Output Layer. Both the half will be trained differently. We will start by training the Hidden-to-Output Layer using simple Back propagation algorithm using modified cost function [4].

Firstly the cost function is :- mm=$\sum_{\mu j} \tau_j$, where $\tau_j = \frac{E_j^2}{2a_j(1-a_j{}^2)}$ .... (1)

with $E_j = t_j$-a_j and $a_j = activation\ function$=sigmoidal function =$g\left(\sum_i w_{ij} y_i\right)$=$O_{o_j}$

$$E_{HO}[w] = \sum_{\mu j} \left[\frac{(t_j - O_{o_j})^2}{2O_{o_j}\left(1-O_{o_j}{}^2\right)}\right] \quad (2)$$

Here , g is the sigmoid function, t_j is the target value for the output unit j for the pattern μ where O_{o_j} is the output value of the hidden layer unit j. The weight from hidden unit i to output unit j is w_{ij}.

Hidden to output layer calculation

To update the hidden-to-output connections the following rule is used and also we have used Three term back propagation equation as base[3].

$$\left[\Delta W_{ij}\right]^{t+1} = -\eta\frac{\partial E}{\partial W} + \alpha[\Delta W]^{t} + \gamma e\big(w(t+1)\big) = \eta\textstyle\sum_{\mu j}\left(t_j{}^{\mu} - O_{o_j}{}^{\mu}\right) < q > O_{o_j}{}' y_i + \alpha\left[\Delta W_{ij}\right]^{t} + \gamma e\big(w(t+1)\big) \text{.................(3)}$$

$$\text{Where } q = \frac{\left(O_{o_j}{}^{3} - 3t_j O_{o_j}{}^{2} + O_{o_j} + t_j\right)}{2O_{o_j}{}^{2}\left(O_{o_j} - 1\right)^{2}\left(O_{o_j} + 1\right)^{2}} \text{.................(4)}$$

Here, η is a learning rate ranges between 0 and 1 while α is momentum factor γ propotional factor .

Ideal value calculation

Now to train the Hidden-to-Output layer , ideal values of hidden units are described as the correct value to get rid of the error at the output layer . This ideal values are derived by Newton's method approximation.

The error function is approximated by Newton's method using first derivates only,

$$\mathrm{E(w)} \approx \mathrm{E(Y)} + (\Gamma - \mathrm{Y})\nabla \mathrm{E(Y)} \text{.................(5)}$$

where ,Y is a current vector in hidden layer for a pattern μ *and* Γ is ideal vector of Y such that E(w)=0.
Γ is the most desirable when E(w)=0. Thus,

$$\Gamma - \mathrm{Y} \approx \frac{-\mathrm{E(Y)}\nabla \mathrm{E(Y)}}{|\nabla \mathrm{E(Y)}|^{2}} \text{.................(6)}$$

The ideal value of hidden unit i , b_i is the ith component of the vector Γ and y_i is the ith component of Y.
So, from eq.(11) we can derive that,

$$b_i = \frac{E(Y)\sum_{\mu j}\left(t_j{}^{\mu} - O_{o_j}{}^{\mu}\right) < q > O_{o_j}{}' w_{ij}}{\left|\sum_{\mu j}\left(t_j{}^{\mu} - O_{o_j}{}^{\mu}\right) < q > O_{o_j}{}' w_{ij}\right|^{2}} + y_i \text{.................(7)}$$

Where b_i is the ith component of the vector Γ and y_i is the ith component of Y.

Input-to-hidden layer calculation

To calculate input-to-hidden layer calculation the updating rule and error calculating function is as follows,

$$E_{IH}[w] = \sum_{\mu j}\left[\frac{(b_i - O_{H_i})^{2}}{2O_{H_i}\left(1 - O_{H_i}{}^{2}\right)}\right] \text{.................(8)}$$

$$[\Delta W_{ki}]^{t+1} = \eta\left(b_i{}^{\mu} - O_{H_i}{}^{\mu}\right) < q *> O_{H_i}{}'\rho_k + \alpha[\Delta W_{ki}]^{t} + \gamma e\big(W_{ki}(t+1)\big) \text{.................(9)}$$

$$\text{Where } q^* = \frac{\left(O_{H_i}{}^{3} - 3b_i O_{H_i}{}^{2} + O_{H_i} + b_i\right)}{2O_{H_i}{}^{2}\left(O_{H_i} - 1\right)^{2}\left(O_{H_i} + 1\right)^{2}} \text{.................(10)}$$

Here, ΔW_{ki} is the weight vector for input to hidden layer connections. We have adopted divide and conquer strategy in which a whole network is divided into two separate parts , Hidden-to-Output Layer and Input-to-Hidden Layer.

Here we have trained both the parts differently. The Hidden-to-Output Layer is trained similar to BP algorithm as equation (3), while the other part is trained with using equations as equation (9).

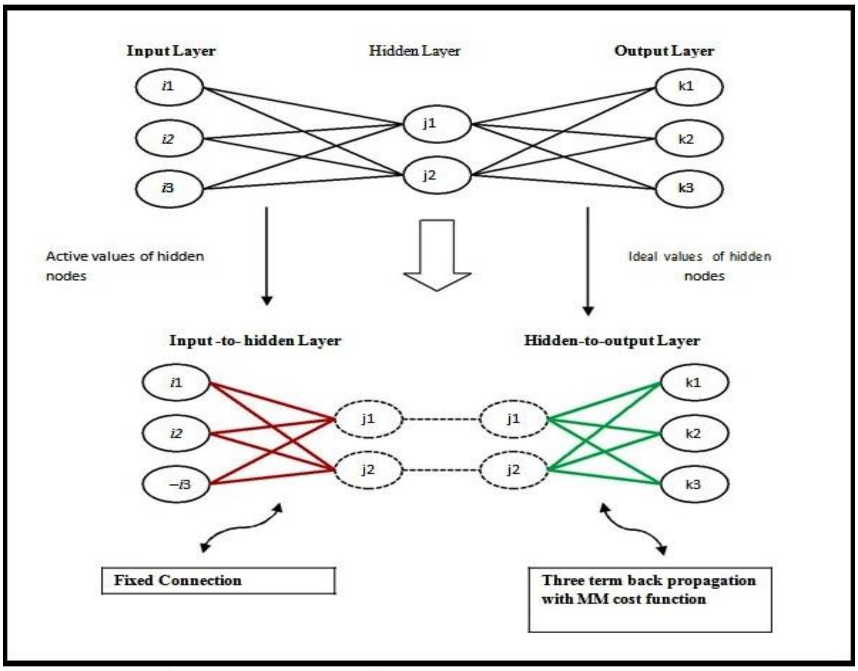

Fig.1 the Structure of the Proposed Algorithm

The Proposed Algorithm is as follows

Step 1. For each pattern at the first epoch , *Compute the output values* of hidden and output layers using given parameters and input values.

Step 2. By having input-to-hidden connections are fixed, train the Hidden-to-Output Layer. The inputs will be the output of the Hidden layer. The ideal value for each output Unit is similar to the original network.
The Error is evaluated by eq.(2) and the weights of the hidden to output connections are updated using eq.(3).

Step 3. Now, the whole training process will be said to successful if the training process of Step 2 converges within the given time limit specified (30sec). It said to be fail if the process goes over time or go slow(when the error cannot be reduced more than one percent of the previous error) then go to Step 4.

Step 4. Now choose one unit from hidden layer in order, and calculate the ideal value using Newton approximation according to equation (7).

Step 5. By using equation(8), train the Input-to-hidden layer while all the other connections are set to fixed. Now same as before if it converges or the training speed goes slow as described in Step 3 , then move to Step 2.

III. EXPERIMENTAL RESULTS & PERFORMANCE EVALUATION

The proposed algorithm in this paper shown is implemented in MATLAB environment. Each experiment was performed using Back propagation algorithm and the Proposed algorithm. By having the results, we analyze and evaluated the convergence rate (success rate , SE), average learning time (time , in seconds) in case of success. A MSE is set to 0.01 and the limit to the training time for each experiment are set to 30 seconds.

One of the experiment was on 4-bit parity problem which is given in Table.1. While the second problem was the Soil Classification problem and the dataset for the problem is taken from reference [6]. Each experiment was performed 15 times with random bipolar weight vectors ranging from -1 to 1, learning rate is set to 0.1, and momentum is set to 0.1. Each experiment was performed with different hidden node values which is one of the key factors to generate local minima. We have used different hidden node values so that we can achieve better results.

Table 1:- Results from 4-bit parity problem

No of Hidden Nodes	Backpropagation Algorithm (SE %)	Proposed Algorithm (SE %)
2	0	0
15	100	63
20	46	92
50	0	100

Table 2:- Results from Soil Classification Problem

No of Hidden Nodes	BP algorithm (SE %)	Proposed Algorithm(SE%)
2	93.3	0
5	100	21.4

10	93.3	56
20	33.	92.6
30	26.7	100
50	0	100

We can clearly see from the results that Back propagation shows poor performance when faced with large number of hidden nodes. Here our proposed algorithm which separates the network in two parts and shows improving results over back propagation algorithm for SE. The Table 1 and Table 2 depicts that for both the problems, as the no of hidden nodes increases the SE rate is decreases. On the other hand in proposed algorithm shows that as faced to large number of hidden nodes the algorithm gives better results.

IV. CONCLUSION

BP Algorithm is known for its mathematical simplicity and accuracy. The main aim of this research work is to proposed an algorithm that solves local minima problem for large number of hidden nodes with possessing the same mathematical simplicity. Here , the proposed algorithm cannot guarantee to reach to a global minima but it shows that it is faster than BP algorithm when faced to complex problems with having large number of hidden nodes. By separating the network we can simplify the complexity of large networks. Even though the proposed algorithm is tested over the classical problems but there is still a need to test the algorithm with some hard core real complex problems which will further can assist in the reliability of the algorithm.

REFERENCES

[1] Rumelhart, D.E., Hinton, G. E., and Williams, R, J. Learning Internal Representations by Error Propagation.1986
[2] Yu, Chien-Cheng, and Bin-Da Liu. "A backpropagation algorithm with adaptive learning rate and momentum coefficient." Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on. Vol. 2. IEEE, 2002.
[3] Shamsuddin, Siti Mariyam, Maslina Darus, and Fadhlina Izzah Saman. "Three term backpropagation algorithm for classification problem." Neural Network World 17.4 (2007): 363.
[4] Shamsuddin, Siti Mariyam, et al. "Study of cost functions in Three Term Backpropagation for classification problems." *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. IEEE, 2009.
[5] Zweiri, Yahya H., James F. Whidborne, and Lakmal D. Seneviratne. "A three-term backpropagation algorithm." Neurocomputing 50 (2003): 305-318.
[6] Rajasekaran, Sundaramoorthy, and GA Vijayalakshmi Pai. *NEURAL NETWORKS, FUZZY LOGIC AND GENETIC ALGORITHM: SYNTHESIS AND APPLICATIONS (WITH CD)*. PHI Learning Pvt. Ltd., 2003. pp.64.
[7] Bumghi Choi,Ju-Hong Lee,Deok-Hwan Kim, Solving local minima problem with large number of hidden nodes on two-layer feed forward artificial neural networks Neurocomputing 71 (2008).
[8] Sun, Jianye. "Local coupled feedforward neural network." *Neural networks* 23.1 (2010): 108-113.