International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:12, 2007

Pattern Classification of Back-Propagation Algorithm Using Exclusive Connecting Network

Insung Jung, and Gi-Nam Wang

***Abstract*—**The objective of this paper is to a design of pattern classification model based on the back-propagation (BP) algorithm for decision support system. Standard BP model has done full connection of each node in the layers from input to output layers. Therefore, it takes a lot of computing time and iteration computing for good performance and less accepted error rate when we are doing some pattern generation or training the network.

However, this model is using exclusive connection in between hidden layer nodes and output nodes. The advantage of this model is less number of iteration and better performance compare with standard back-propagation model. We simulated some cases of classification data and different setting of network factors (e.g. hidden layer number and nodes, number of classification and iteration). During our simulation, we found that most of simulations cases were satisfied by BP based using exclusive connection network model compared to standard BP. We expect that this algorithm can be available to identification of user face, analysis of data, mapping data in between environment data and information.

***Keywords*—**Neural network, Back-propagation, classification.

I. Introduction

MANY people and industries are interested in the decision support system, and prediction systems for the better choice and reduction of risk based on intelligence method. Especially, artificial neural network based decision making and prediction systems. These methods are seemed to be successful to solve difficult and diverse problems by supervised training methods such as back-propagation algorithm. This algorithm is the most popular neural network architecture for supervised learning, because it is based on the weight error correction rules. Although back-propagation algorithm could correct weights, it still got error and takes much of pattern generation computing time.

The model structure of BP (back-propagation) classification algorithm use full connection each layers and nodes from input layer to output layer. Consequently it needs much of calculation.

However, we are not still satisfied with standard neural network or back-propagation model based decision support system because we want to get better quality of decision performance and less computing iteration when we want to develop in the specific domain area.

In this paper, we proposed dividend exclusive connection in between hidden layer nodes and output nodes. This method is almost same as BP model. However, between last hidden layer nodes and output later nodes weights are not full connected. Before output node, it has got exclusive nodes for mapping hidden layer nodes. It could be prevention of given wrong information. We think sometimes full connection of weight could be given no effect or wrong calculation value and information. The advantage of this model is less number of iteration and better performance compare with standard back-propagation model.

To evaluate of this algorithm, we simulated some cases of classification data and different setting of network factors (e.g. hidden layer number and nodes, number of classification and iteration). We found that proposed dividend exclusive connection based BP model is fairly better performance than standard BP model.

Organization of paper is as follows. Section 2 is a review of the related work; Section 3 describes methodology; Section 4 is simulation results and finally it summarized the conclusion.

II. Related Work

In 1958, Frank Rosenblatt introduced a training algorithm that provided the first procedure for training a simple artificial neural network (ANN): a perceptron [1]. The perceptron is the simplest form of an ANN. It consists of a single neuron with adjustable synaptic weights and a hard limiter. The perceptron learning rule was first proposed by Rosenblatt in 1960 [2]. It is base model of the perceptron training algorithm for classification tasks.

Minsky and Papert (1969) showed that a two-layer feed-forward network can overcome many restrictions of single layer ANN [3]. But it did not present a solution to the problem of how to adjust the weights from input to hidden units. An answer to this question was presented by Rumelhart, Hinton, and Williams in 1986. Similar solutions appear to have been published earlier (Werbos, 1974; Parker, 1985; Cun, 1985) [4]-[7]. The central idea behind this solution is that the errors for the units of the hidden layer are determined by back-propagating the errors of the units of the output layer. For this reason the method is often called the back-propagation learning rule. Back-propagation can also be considered a generalization of the delta rule for nonlinear activation functions and multilayer networks.

Insung Jung, is with Department of Industrial Engineering Ajou University 442-749, Korea (e-mail: insung9908@gmail.com).

Gi-Nam Wang, is with Department of Industrial Engineering Ajou University 442-749, Korea (e-mail: gnwang@ajou.ac.kr).

However, when we use data pattern generation, it needs much of iteration number and computing time. Therefore, we proposed the Pattern classification of back-propagation algorithm using exclusive connecting network.

III. Methodology

A. Background Method Neural

1) Multilayer perceptron l & BP (Back-propagation) model

Standard multilayer perceptron (MLP) architecture consists more than 2 layers; A MLP can have any number of layers, units per layer, network inputs, and network outputs such as fig 1models. This network has 3 Layers; first layer is called input layer and last layer is called output layer; in between first and last layers which are called hidden layers. Finally, this network has three network inputs, one network output and hidden layer network.

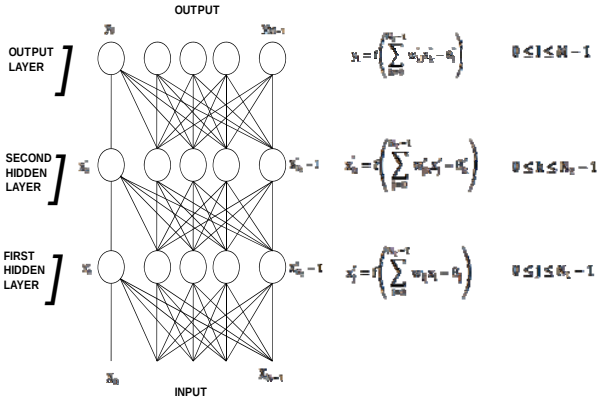

Fig. 1 Standard Multi layer perceptron architecture

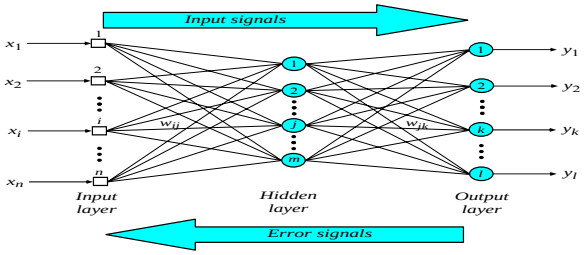

Fig. 2 Back-propagation neural network

However, this research is compared with Back-propagation (BP) model. This model is the most popular in the supervised learning architecture because of the weight error correct rules. It is considered a generalization of the delta rule for nonlinear activation functions and multilayer networks.

In a back-propagation neural network, the learning algorithm has two phases. First, a training input pattern is presented to the network input layer. The network propagates the input pattern from layer to layer until the output pattern is generated by the output layer. If this pattern is different from the desired output, an error is calculated and then propagated backward through the network from the output layer to the input layer. The weights are modified as the error is propagated. According to the Richard P. Lippmann [8], he represents step of the back-propagation training algorithm and explanation. The back-propagation training algorithm is an iterative gradient designed to minimize the mean square error between the actual output of multi-layer feed forward perceptron and the desired output. It requires continuous differentiable non-linearity. The following assumes a sigmoid logistic nonlinearity.

Step1: Initialize weights and offsets

Set all weights and node offsets to small random values.

Step2: Present input and desired outputs

Present a continuous valued input vector X_0, X_1…..X_{N-1} and specify the desired output d_0,d_1,….d_{M-1}. If the net is used as a classifier them all desired outputs are typically set to zero except for that corresponding to the class the input is from. That desired output is 1. The input could be new on each trial or samples from a training set could be presented cyclically until stabilize.

Step 3: Calculate Actual Output

Use the sigmoid non linearity from above and formulas as in fig 3 to calculate output y0,y1….yM-1.

Step 4: Adapt weights

Use a recursive algorithm starting at the output nodes and working back to the first hidden layer. Adjust weights by

$$w_{ij}(t+1) = w_{ij}(t) + n\delta_j x_i' \quad (3)$$

In this equation $w_{ij}(t)$ is the weight from hidden node i or from an input to node j at time t, w_j', is either the output of node i or is an input, η is a gain term, and δ_j, is an error term for node j, if node j is an output node, then

$$\delta_j = y_j(1-y_j)(d_j - y_j) \quad (4)$$

where d_j is the desired output of node j and y_j is the actual output.

If node j is an internal hidden node, then

$$\delta_j = x_j'(1-x_j')\sum_k \delta_j^m w_{jk} \quad (5)$$

where k is over all nodes in the layers above node j.

Internal node thresholds are adapted in a similar manner by assuming they are connection weights on links from auxiliary constant-valued inputs. Convergence is sometimes faster if a momentum term is added and weight change are smoothed by

$$w_{ij}(t+1) = w_{ij}(t) + n\delta_j x_i' + \alpha(w_{ij}(t) - w_{ij}(t-1)) \quad (6)$$

,where0< α <1.

Step 5: Repeat by going to step 2

B. Dividend Exclusive Connection based Back-Propagation

In this paper, we proposed dividend exclusive connection in between hidden layer nodes and output nodes. This method is almost same as BP model. However, between last hidden layer nodes and output later nodes weights are not full connected.

Before output node, it has got exclusive nodes for mapping hidden layer nodes. It could be prevention of given wrong information. We think sometimes full connection of weight could be given no effect or wrong calculation value and information.

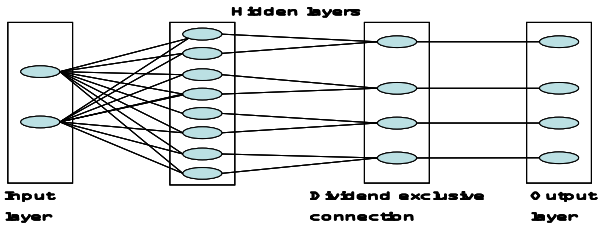

Fig. 3 Dividend exclusive connection based BP

Step1. Modeling of dividend exclusive connection based BP

To Select each nodes from input to output (input layer nodes, first hidden layer nodes, second hidden layer nodes and output nodes). It has to be same second hidden layer nodes number with output layer nodes. First nodes of hidden layer connected with second hidden layer nodes in the couple way (Dividend exclusive connection); it has to be more than double nodes than divided exclusive nodes.

N : the number of input node.
M : the number of output node.
Z : the number of first hidden layer node.

$w_{ij}^{(I,H)}$ is the weight between input i th node x_i and first hidden layer j th node y_j^H where $i = 0,1,\cdots,N-1$ and $j = 0,1,\cdots,Z-1$.

First hidden layer nodes must be grouped into the number of dividend exclusive layer nodes, M.

$w_{i_j,j}^{(H,DX)}$ is the weight between first hidden i_j th node $y_{i_j}^H$ and dividend exclusive layer j th node y_j^{DX} where $j = 0,1,\cdots,M-1$. i_j is the hidden layer i th node in group j.

$w_i^{(DX,O)}$ is the weight between dividend exclusive layer i th node y_i^{DX} and output i th node y_i^o where $i = 0,1,\cdots,M-1$.

Step2: Initialize weights and offsets

Set all weights and node offsets to small random values.

Step3: Present input and desired outputs

Present a continuous valued input vector $x_0, x_1, \cdots, x_{N-1}$ and specify the desired output $d_0, d_1, \cdots, d_{M-1}$. If the net is used as a classifier them all desired outputs are typically set to zero except for that corresponding to the class the input is from. That desired output is 1. The input could be new on each trial or samples from a training set could be presented cyclically until stabilize.

Step 4: Proceeding Forward.

Calculate hidden layer vector $Y^H(y_0^H, y_1^H, \cdots, y_{Z-1}^H)$ with input vector $X(x_0, x_1, \cdots, x_{N-1})$ and weight vector $W^{(I,H)}$

$$Y^H = F(XW^{(I,H)})$$

and dividend exclusive layer node y_j^{DX}

$$y_j^{DX} = f\left(\sum_{j=0}^{M-1} y_{i_j}^H w_{i_j,j}^{(H,DX)}\right)$$

where $f(x)$ is the sigmoid function.

Step 5: Calculate Actual Output

Output layer i th node,

$$y_i^O = f\left(\sum_{j=0}^{M-1} y_i^{(DX)} w_{ij}^{(DX,O)}\right)$$

Step 6: Adapt weights

Use a recursive algorithm starting at the output nodes and working back to the first hidden layer. Adjust weights by

$$w_{ij}(t+1) = w_{ij}(t) + n\delta_j x_i' \quad (3)$$

In this equation $w_{ij}(t)$ is the weight from hidden node i or from an input to node j at time t, w_j', is either the output of node i or is an input, η is a gain term, and δ_j, is an error term for node j, if node j is an output node, then

$$\delta_j = y_j(1-y_j)(d_j - y_j) \quad (4)$$

where d_j is the desired output of node j and y_j is the actual output.

If node j is an internal hidden node, then

$$\delta_j = x_j'(1-x_j')\sum_k \delta_j^m w_{jk} \quad (5)$$

where k is over all nodes in the layers above node j.

Internal node thresholds are adapted in a similar manner by assuming they are connection weights on links from auxiliary constant-valued inputs. Convergence is sometimes faster if a momentum term is added and weight change are smoothed by

$$w_{ij}(t+1) = w_{ij}(t) + n\delta_j x_i' + \alpha(w_{ij}(t) - w_{ij}(t-1)) \quad (6)$$

,where $0 < \alpha < 1$.

Step 7: Repeat by going to step 3

The advantage of this model is less number of iteration and better performance compare with standard back-propagation model.

To evaluate this algorithm, we simulated some cases of classification data and different setting of network factors (e.g. hidden layer number and nodes, number of classification and iteration). We found that proposed dividend exclusive connection based BP model has fairly better performance than standard BP model.

IV. Simulation Result

In this simulation, with random generation data between 0 and 1 using neural network algorithms, standard BP and dividend exclusive connection are based on BP model. The evolutional condition was exactly same at each models. However, the comparing of each models is a little bit difficult situation because standard BP needs much of iteration number. Tables I and II present the performance of compared models. First test of our evaluation case was small size of output nodes type classification. In this time both models have very successful results. The performance values are more than 95% (standard model: 97.8%, proposed model: 98.4%). Our proposed model was a little bit better than standard model. Other simulation case is 16 output nodes classification.

Some case of simulation by standard model of BP was not satisfied because of being lack of nodes and iteration number. But when we setup enough iteration number, it is also quite reasonable. On the other hand, our model of dividend exclusive connection based BP model was proved to be better performance than standard BP results.

TABLE I
Back-Propagation Model Performance

Number of cluster	Data number	Hidden layer / node number	Iteration	Performance
4	500	2 (8, 4)	500	97.8% (489/500)
16	1000	2 (8, 4)	500	75.8% (379 / 500)
16	1000	2 (8, 4)	1000	82% (410/500)
16	500	2 (16, 4)	1000	83.8% (419/500)
16	1000	2 (16, 16)	1000	90.4% (452/500)
16	1000	2 (32, 16)	1000	91.4% (457/500)
16	1000	2 (256, 16)	1000	94.8% (474/500)
16	1000	2 (256, 16)	1500	96% (480/500)

TABLE II
Dividend Exclusive Connection based BP Performance

Number of cluster	Data number	Hidden layer / node number	Iteration	Performance
4	500	2 (8, 4)	500	98.4% (492/500)
16	500	2 (32, 16)	500	91%(455/500)
16	500	2 (32, 16)	1000	93.2% (466/500)
16	500	2 (256, 16)	300	91% (455/500)
16	500	2 (256, 16)	500	94% (470/500)
16	500	2 (256, 16)	900	95.2% (476/500)
16	500	2 (256, 16)	1500	96.6% (483/500)

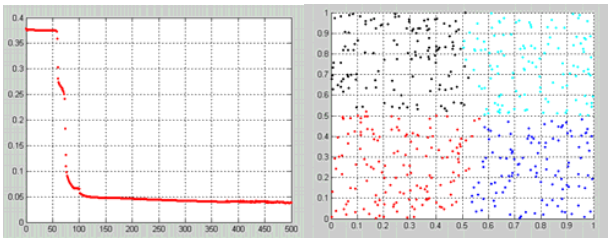

Fig. 4 BP model result (4output, 500 iteration)

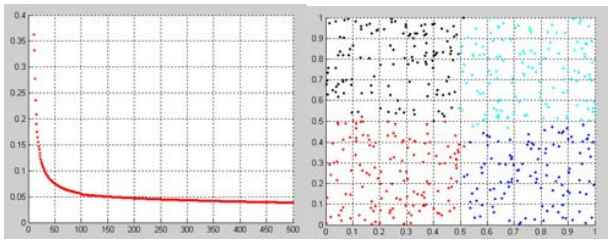

Fig. 5 Dividend exclusive connection based BP model (4output, 500 iteration)

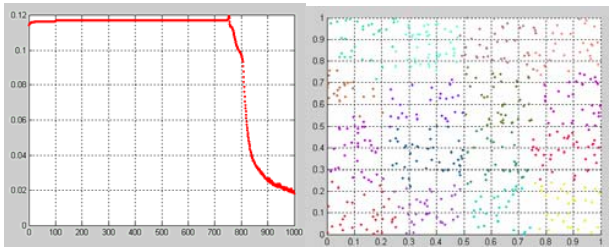

Fig. 6 BP model result (16output, 1000 iteration)

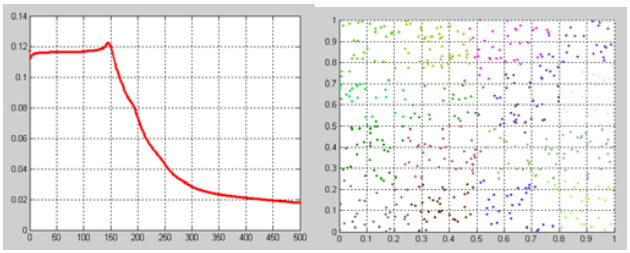

Fig. 7 Dividend exclusive connection based BP model (16output, 500 iteration)

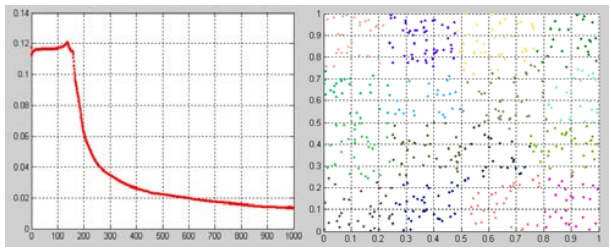

Fig. 8 Dividend exclusive connection based BP model (16output, 1000 iteration)

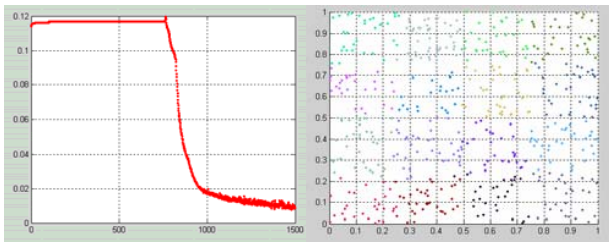

Fig. 9 BP model result (16output, 1500 iteration)

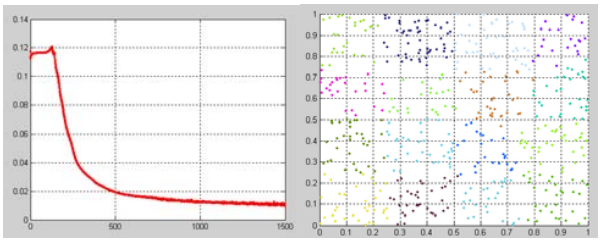

Fig. 10 Dividend exclusive connection based BP model (16output, 1500 iteration)

V. Conclusion

The objective of this paper is to design of pattern classification model based on the back-propagation (BP) algorithm for decision support system. Standard BP model has done full connection of each node in the layers from input to output layers. Therefore, it takes a lot of computing time and iteration computing for good performance and less accepted error rate when we are doing some pattern generation or training the network. However, our proposed model is using exclusive connection in between hidden layer nodes and output nodes. The advantage of this model is less number of iteration and better performance compare with standard back-propagation model.

To evaluate of this algorithm, we simulated some cases of classification data and different setting of network factors (e.g. hidden layer number and nodes, number of classification and iteration) with same conditions. Small size of output nodes type classification performance results are more than 95% (standard model: 97.8%, proposed model: 98.4%). However, standard model is a little bit hard to use complicated case because of being lack of nodes and iteration number. On the other hand, our model of dividend exclusive connection based BP model was proved to be better performance (96%) than standard BP results. We found that proposed dividend exclusive connection based BP model is fairly better performance than standard BP model.

The proposed model could be useful for identification of user face, analysis of data, mapping data in between environment data and information. The limitation of this research is not use real field data.

Acknowledgment

This research is supported by the ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

References

[1] F. Rosenblatt, "A Probabilistic Model for Information Storage and Organization in the Brain," Cornell Aeronautical Laboratory, vol. 65, pp. 386-108, 1958.
[2] R. H. Rosenblatt, "The Atlantic species of the blennioid fish genus", 1960.
[3] M. Minsky, & Papert, S., "Perceptrons: An Introduction to Computational Geometry," The MIT Press, pp. 3, 26, 31, 33, (1969).
[4] D. E. Rumelhart, Hinton, G. E., & Williams, R. J. , " Learning representations by backpropagating errors," Nature, vol. 323, pp. 533-536, 1986.
[5] P. J. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Unpublished doctoral dissertation Harvard University., 1976.
[6] D. B. Parker, "Learning-Logic (Tech. Rep. Nos. TR{47)." 1985.
[7] Y. L. Cun, "Une procedure d'apprentissage pour reseau a seuil assymetrique," Proceedings of Cognitiva, vol. 85, pp. 599-604.
[8] Richard P. Lippmann,:" An introduction to computing with neural network", IEEE ASSP magazine, 1987, pp. 4-22.