ISSN 2229-5518

Improved Pattern Classification Model Using Agents Technology Based Cooperative Neural Networks with Pruning

Hanaa Mohammed Mashjel Dr. Loay E. George Dr. Ban N. Dhannoon

Abstract—This paper attempts to improve the efficiency of pattern classification model which proposed by "Hanaa M.M 2014" in [7], where an efficient classification model built using Cooperative Neural Network (CNN). It decomposes the large complexity classification task into several sub-tasks; each one is handled by a simple, fast and efficient classifier (agent). Although the previous system tested on various simulation cases and was able to obtain a 100% correct classification ratio upon the tested dataset with optimal execution time; but here in this paper, we improve its ability to choose the best useful sub-set of discriminating features given to the system through the input layer. The selection of best features is a complex process because it requires extensive experience and a deep understanding of the problem domain in case of very complex data set or when the number of distinctive features are few, so neural networks were able to detect the features have no role in CNN modules (which trained using Bp with adaptive learning rate) and, then, making decisions' by decreasing their weights or even applying pruning procedure to delete them such that performance of the classification process is impoved in terms of accuracy or time. The tests results indicated a success rate nearly 100% when simulated complex patterns are fed to the improved model. Also, the test results indicate that the raise of complexity degree does not cause dramatic effect on the system performance; except the model execution time becomes more than few seconds which considered a promising results.

Keywords— Software agents technology, Cooperative Neural Network (CNN), Modular Neural Network (MNN), Back propagation **(Bp),** adaptive learning rate, classification, pruning.

1 INTRODUCTION

Feature selection is critical to the success of large-scale automated classification schemes since different features reflect different aspects of the dataset. Careful selection features can improve classification [4]. In practice, the optimal subset of features is usually unknown and it is common to have irrelevant or redundant features at the beginning of the pattern classification tasks [9].

So including too many features may introduce irrelevant or distracting attributes that impair performance. Some features that are ineffective individually may be predictive when combined with others. Other features that are predictive individually may become contradictory when combined, so choosing a useful set of features may be a complex process that requires extensive experience and a deep understanding of the problem domain [4]. Therefore it becomes important to select the most representative features from the original feature set such that the recognition rate is retained with considerably reduced feature dimensions [9]. Since applying neural network proficiency to monitored classification is very much powerful in terms of robustness and adaptively, also it is useful for decomposing complex classification tasks into simpler sub-tasks and then puzzle out each sub-task efficiently using a simple module [8]. In our proposed model the pruning mechanism applied on MNN classifiers is adopted to find the optimal subsets of features for each class of the used data sets.

2 BACK PROPAGATION ALGORITHM [1]

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 steps summarize the algorithm work flow:

Step1: Initialize weights

Set all weights to small random values.

Step2: define input and desired outputs vectors

Present a continuous valued input vector: $x_0, x_1, \ldots, x_{N-1}$ and specify the desired output $d_0, d_1, \ldots d_{N-1}$. If the net is used as a classifier then 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 the actual output

Use the sigmoid non linearity to calculate output $y_0, y_1, \ldots, y_{N-1}$.

Step 4: Adapt weights

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

$$w_{ij}^{(t+1)} = w_{ij}^{t} + \eta\, \delta_j\, x_i \qquad \ldots\ldots\ldots\ldots(1)$$

In this equation is w_{ij}^{t} the weight from hidden node i, or from an input to node j at time t, xi is either the output node i or is an input, η is a gain term, and δj is an error term for node j, if node j is an output node, then

$$\delta_j = y_j(1-y_j)\,(d_j-y_j) \qquad ,\ldots\ldots\ldots\ldots(2)$$

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

$$\delta_j = x_j(1-x_j)\Sigma\, \delta_{jm} w_{jk} \qquad ,\ldots\ldots\ldots\ldots(3)$$

Where k is over all nodes in the layers above node j. Convergence is some times faster if a momentum term is added and weight change are smoothed by

$$w_{ij}^{(t+1)} = w_{ij}^{t} + \eta\delta_j\, x_i + \alpha(\, w_{ij}^{t} - w_{ij}^{t-1}) \qquad ,\ldots\ldots\ldots\ldots(4)$$

Where $0<\alpha<1$.

Step 5: Repeat by going to step 2

3 Adaptive Learning Rate

The adaptive learning rate can be adopted to speed up the convergence of the algorithm. For batch training strategy, the learning rate can be adjusted as follows [2].

If $err_i^t \times err_i^{t-1} > 0$ Then $\eta_i^t = \eta_i^{t-1} \times \eta_i^+$

Else If $err_i^t \times err_i^{t-1} < 0$ Then $\eta_i^t = \eta_i^{t-1} \times \eta_i^-$

Else $\eta_i^t = \eta_i^{t-1}$

Where $\eta_i^+ > 1$, $\eta_i^- < 1$

4 Modular Neural Network

Decomposing a complex computational problem into sub-problems is computationally simpler to solve individually, then the solutions of sub-problems can be combined to produce a solution to the full problem; this can efficiently lead to compact and general solutions. Modular neural networks represent one of the ways in which this divide-and-conquer strategy can be implemented [10].

This "Divide and conquer" approach gives a number of advantages to such a neural network. It includes complexity reduction in model, scalability, flexibility in design and implementation, robustness, and computational efficiency. These properties make modeling of problems with a large number of dimensions very efficient and easy while using modularity [3].

A neural network is said to be modular if the computation performed by the network can be decomposed into two or more modules (subsystems) that operate on distinct inputs without communicating with each other. The outputs of the modules are mediated by an integrating unit that is not permitted to feed information back to the modules [11].

5 Cooperative Neural Networks

Cooperative neural networks (CNN) have been used for the last few decades in a broad variety of applications. It is suitable to use if the application is conclusive or decision based, like in classification or clustering problems. Specifically, areas such as data mining and financial engineering, has lot of use of cooperative neural network [8]. CNN is specifically useful for applications with a wide range of overlap in the input-space; they give enough information which enables the voting scheme to assign testing samples to their correct modules. Moreover, the specialized modules dedicated to the high-overlap regions are capable of drawing quite complex boundaries. In

ISSN 2229-5518

general, cooperative schemes prove to be more efficient and capable of handling more complex problems than other schemes [6].

6 Pruning

If we have executed the weight decay long enough and notice that for a neuron in the input layer all successor weights are 0 or close to 0, we can remove the neuron, hence losing this neuron and some weights and thereby reduce the possibility that the network will memorize. This procedure is called pruning [5].

7 Methodology

Through this section, we will explain the structure of the used CNN, and the used pruning procedure.

(A) CNN structure

Each module composed of three layers; input layer contain n nodes, where n is the number of class features, hidden layer contained p nodes, where:

$$p = n/2 + 1 \qquad \text{,.................................(4)}$$

And output layer contains one node, its output is either near to 1 (in case the sample belongs to the class) or not.

(B) Pruning Procedure

choosing a useful set of features is a complex process that requires extensive experience and a deep understanding of the problem domain in case of very complex data set or when the number of distinctive features are few, but neural networks were able to detect features those have no role in CNN modules (which trained using Bp with adaptive learning rate) decisions' by decreasing their weights then the applied pruning procedure delete them.

Pruning Procedure Steps:

Set RD = right decisions at time t
Set RD_old = right decision at time t-1

Step1: Test data and find **RD**
Step2: While **RD_old <=RD**
For each class
Call Pruning_sort procedure.
Delete the feature which has least sum_ weights value and delete its connections.
Train and save new set of weights.
End for
Set **RD_old =RD**
Test data and find **RD**
End while
Step3: Save pruning information.

Where in the previous procedure **Pruning_sort procedure** the features are sorted according to their weights and re-test the data set by calling Pruning_Test procedure, both procedures are explained in the following.

Pruning_sort procedure

Step1: Initialize $sum\_wights_i = 0$, where $0 < i <= N$; N is the number of input nodes

Step2: For each hidden node H_j set:

$$sum\_wights_i = sum\_wights_{i} + w_{ij}$$

Step3: Sort features according to the value of sum_weights (in ascending order).

(C) Pruning_Test Procedure

Inputs: W_set, Sample, Index

Steps:

```
for I=0  to W.count - 1
    set  Output= forwardphase (sample, W_set (i))
    if (Round (output) =1) then
        Add (I) to Class_list
end for
if (class_list.count=1)  and (class_list[0] = Index)
    then Return True
else
    for I=0 to Class_list.count-1
        Set ms = MSE( forwardphase (Sample,
                      W(Class_list) ) , 1)
        Add ms to MSE_list
    end for
    Set Temp_index = class index which have least
                      MSE from 1
    if (Temp_index=index)  then  Return True
    else Return False
end if
```

Table (1): The adopted system parameters during the training & test phases (for Experiment 1)

No. Hidden Units	6
Activation Function at Hidden Layer	Sigmoid_bipolar
Training Time	5.562 sec
Right Decisions	100
Wrong Decisions	0

8 Results

Experimental results show the efficiency of the proposed system for pattern classification in tested simulated cases, where the number of discriminative features are apportion with the size of data set. next the results of two experiments, first experiment classify 10 classes each class contain 10 samples each sample composed of 10 features(where 30% of the data samples used for testing only (unseen from the network) ,and 70% of the samples used for training for both experiments). And then the complete data set classified to obtain the results.

Table (1) shows the result of experiment1 where the classification model was able to classify all samples correctly in 5 s only even there are unstable features but because the stable features (7, 5, 3, and 2) and the power of CNN in wide range of overlap in the input-space; they gave enough information which enables the voting scheme to assign testing samples to their correct modules.

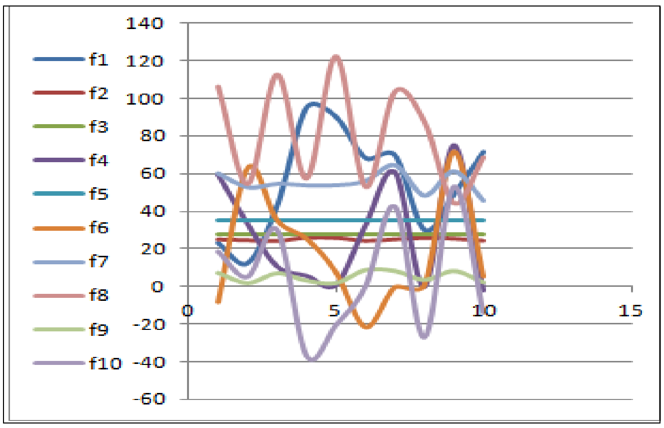

Figure (1): Features behavior in class1 (Experiment 1)

Second experiment we increase the problem complexity by make number of classes 10 ,samples of each class are 5 only each of 5 feature only and decreasing the number of discriminative features, and making most of features had unstable behaviors; then the network could not learn how to classify the samples with the same efficiency without pruning phase.

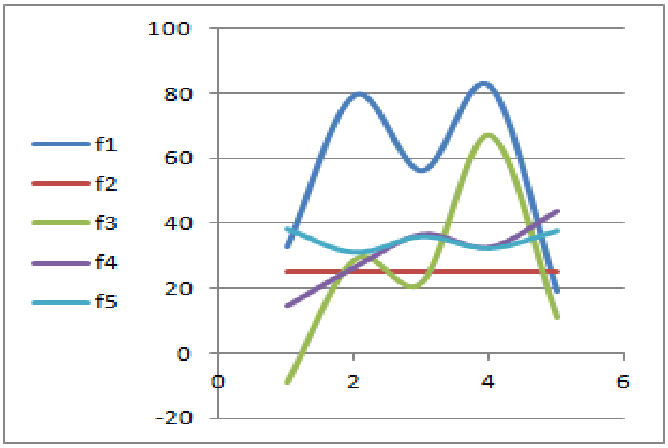

Figure (2): Features behavior in class1 (Experiment 2)

Table (2): Training & run information's (Experiment 2)

No. Hidden Units	4
Activation Function at Hidden Layer	Sigmoid_bipolar
Training Time	2.175 sec
Right Decisions	25
Wrong Decisions	25

Table (2) is not our final result for this experiment, the classifier agent would call pruning procedure to complete solving the problem.

As pruning procedure delete one feature (which has the least W_Sum) from every class. After each class lost a feature the weight set of each class are updating by deleting deleted feature connections and then train again and save new weights, then testing data set again if no improvement in classification results then another feature would be deleted.

Table (3) Final Pruning Results (Experiment2)

No. Deleted Features	2
No. of Right Decisions	50
Pruning Execution Time	2.877 sec

Then the total execution time of this experiment could be performed as the following:

Total time= Initial classification time + Pruning time
= 2.8772179 +2.1750999
=5.052318 s

So as seen in table (3) pruning procedure was able to delete weak features which restrict network learning and improve classification accuracy without taking long time.

9 CONCLUSIONS

The objective of this paper is to improve the efficiency of pattern classification model based on the back-propagation (BP) algorithm for decision support system proposed by us in [7]; by we improved its ability to choose a useful set of features. To evaluate the improved model, we simulated hard cases of classification data, the proposed model gave 100% correct classification rate within short execution time even. If we had complicated data set the pruning procedure was able to delete the weak discriminating features (which restrict the network learning).

10 REFERENCES

[1] Amit G., Y. P. Kosta, Gaurang P., Chintan G.,"Initial Classification Through Back Propagation In a Neural Network Following Optimization Through GA to Evaluate the Fitness of an Algorithm", International Journal of Computer Science & Information Technology (IJCSIT), Vol. 3, No 1, Feb 2011.

[2] Arti G., Manish S.," Performance of ANN using Back Propagation Algorithm for Medical Diagnosis System ", International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970) Volume 2 Number 1March 2012.

[3] Bipul P., Tarun J., Vishal K. and Tarush G.," Evolutionary Modular Neural Network Approach for Breast Cancer Diagnosis ", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, January 2012.

[4] Charles W. , Massimiliano V.," A Neural Network method for Automated Feature Selection in Financial Time Series Forecasting ", Journal of Neural Comput & Applic (2012) 21:969–977,DOI 10.1007/s00521-012-0830-8, Springer-Verlag London Limited , 2012.

[5] David K.," A Brief Introduction to Neural Networks ", 2005.

[6] Gasser A., Mohamed K.," Modular Neural Network Classifiers: A Comparative Study", Journal of Intelligent and Robotic Systems 21: 117–129, 1998.

[7] Hanaa M. M , Loay E. G.,Ban N. D., " Agents Technology Based Cooperative Neural Networks for General Pattern Classification Model", accepted for publication in International Journal of Scientific & Engineering Research Volume 5, January-2014 .

[8] Sanjiban S. R.," Probability method of reliability For cooperative neural network ", Annals. Computer Science Series. , Vol. 8 Issue 1, Special section p160-168, 2010.

[9] Sung-Nien Y., Ming-Yuan L. ," Conditional mutual information-based feature selection for congestive heart failure recognition using heart rate variability ", Journal of computer methods and programs in biomedicine 108(2012) 299–309, Elsevier Ireland Ltd, 2012.

[10] Vineet R. K., Xin Y., Bernhard S., Yaochu J., HeikoWersing," Co-evolutionary Modular Neural Networks for Automatic Problem Decomposition ", IEEE Congress on Evolutionary Computation ,2005.

[11] Farooq A.," Biologically Inspired Modular Neural Networks", Ph.D thesis, Virginia Polytechnic Institute and State University, 2000.