Agents Technology Based Cooperative Neural Networks for General Pattern Classification Model

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

Abstract—Software agents technology took an important role in information technology world for solving the complex problems in the real world, they behave like human intelligently, autonomously, cooperatively, and socially to solve problems or to support human users. In this paper an efficient classification model using agent's technology based on Cooperative Neural Network (CNN) is introduced, where a modular Neural Network (MNN)-classifiers (trained using back propagation with adaptive learning rate) attempts to reduce the effect of complex classification problems by decomposing the large complexity classification task into several sub-tasks; each one is handled by a simple, fast and efficient agent (classifier), each agent implemented using neural network where the role of neural network technology in agents is a supporting one. In this work a simulator has been developed to generate data sets that have controllable statistical behaviors with various degree of complexity. The simulators as well as the whole classification model have been built using "vb.net". The developed system was tested using different simulation cases and was able to obtain 100% correct classification accuracy for all cases in optimal execution time (because it takes only 5.5 seconds only to classify 100 sample each of 10 feature).

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

1 INTRODUCTION

Classification is one of the most frequently encountered decision making tasks of human activity. A classification problem occurs when an object needs to be assigned into a predefined group or class based on a number of observed attributes related to that object. Many problems in business, science, industry, and medicine can be treated as classification problems [13], since we are living in a world full of data every day people encounter a large amount of information and store or represent it as data, for further analysis and management. One of the vital means in dealing with these data is to classify or group them into a set of categories or clusters. In order to learn a new object or understand a new phenomenon, people always try to seek the features that can describe it [11], and 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 puzzles out each sub-task efficiently using a simple module [12].

2 AGENTS TECHNOLOGY

An intelligent agent can be defined as a piece of software which performs a given task using information gleaned from its environment to act in a suitable manner so as to complete the task successfully. The software should be able to adapt itself based on changes occurring in its environment, so that a change in circumstances will still yield the intended result [1] [9] [10] [14].

3 NEURAL NETWORKS

Neural network one of the intelligence method based decision making and prediction systems where these methods are seemed to be successful to solve difficult and diverse problems by supervised training methods such as back-propagation algorithm [8].In the early stages of neural networks research, it was believed that the monolithic or unitary neural networks can solve difficult problems of varying difficulty when applied to classification and regression problems. Continued research in the field of artificial neural networks indicated that there are certain problems which cannot be effectively solved by the global monolithic neural networks. This led to the conception of modular neural networks (MNN) [6].

3.1 Back propagation Algorithm

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 [8] [5] [2]. It requires continuous differentiable non-linearity. The following steps summarized the algorithm assuming sigmoid as activation function.

Step1: Initialize weights

Set all weights to small random values.

Step2: Define input and desired outputs vectors

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

Step3: Calculate actual output

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

Step4: Adapt weights

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

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

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

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

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

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

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

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

where $0< \alpha <1$.

Step5: Repeat by going to step 2.

3.2 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 [3]:

if err_i^t x $err_i^{t-1} >0$ then $\eta_i^t = \eta_i^{t-1}$x η_i^+
else if err_i^t x $err_i^{t-1} <0$ then $\eta_i^t = \eta_i^{t-1}$x η_i^-
else $\eta_i^t = \eta_i^{t-1}$

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

3.3 MNN

The inspiration for modular design of neural networks is mainly due to biological reasons, namely, the functioning of the human brain. Recent advances in neurobiological research have proven that the modularity is a key to the efficient and intelligent working of human and animal brains. Economy of engineering design, and complexity issues in artificial neural network learning are the other motivational factors for modular design of neural networks [6].

Systems that can work upon same or different inputs independently is said to show modularity. As such modular neural network are said to be those that comprise of two or more individual neural modules that can independently act on the inputs to produce output [4].

Modular artificial neural networks are especially efficient for certain classes of problems including classification problems, as compared to the conventional monolithic artificial neural networks [6].

3.4 CNN

Cooperative neural networks 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 [12].

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 general, cooperative schemes prove to be more efficient and capable of handling more complex problems than other MNN schemes [7].

4 Methodology

We will explain the structure of the used CNN, training phase, and finally making decision.

4.1 CNN Structure

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

$$p = n/2 + 1 \qquad ,\ldots\ldots(5)$$

The output layer contains one node, its value is close to 1 when the input vector belongs to the class, otherwise the node value is significantly less than 1.

4.2 CNN Training

Each module is trained using Bp (online mode) with adaptive learning rate; the training is stopped when network error is less than a proposed minimum error or the max number of iterations is exceeded.

4.3 Making Decision In CNN

CNN make decision if a sample belongs to a class using saved thresholds values to define some rules those identify each class samples(as in fig 1), where the test procedure take as inputs the sample and the index of a class to decide if the sample belong to it or not (return a Boolean value), as well as the weights sets of all classes in the data set, then test the sample with all weights sets if the sample belong to more than one class then return false (which means misclassification).

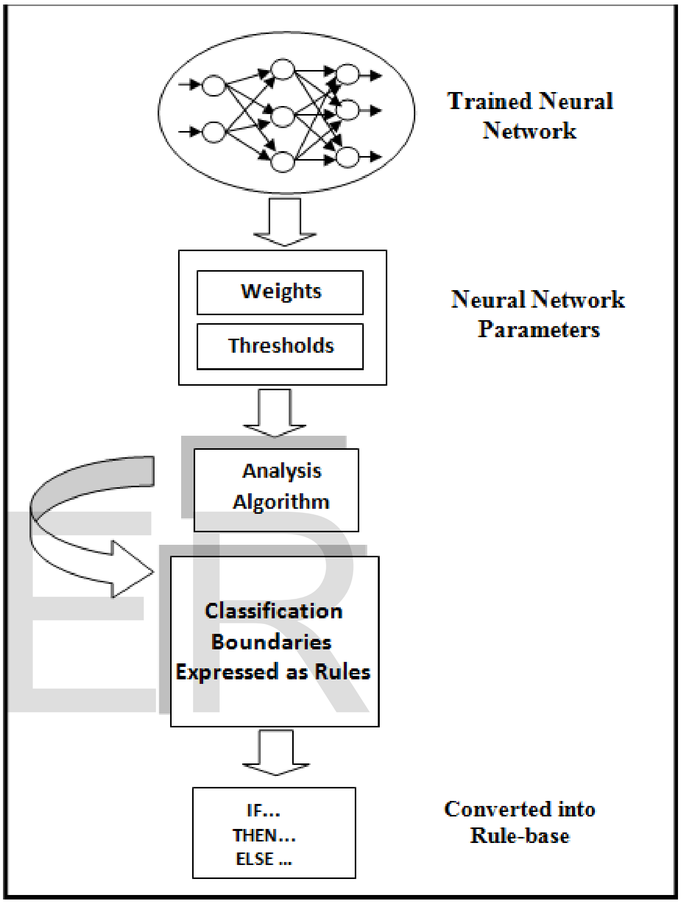

Figure (1): Classifier agent processing

5 Results

Experimental results show the efficiency of the proposed model for pattern classification in the all tested simulated cases, next the results of two data sets classification, each data set contains 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. And then the complete data set tested, the proposed methods (as well as the traditional neural classification method using Bp in batch and online modes) tested in two data sets the first one the features in all classes had a stable behavior while the second experiment some features had unstable behaviors and some other features were overlapped to increase the problem complexity. Next figures show features behavior in class1 of the used data set in experiment1 and experiment2 respectively.

Table (1): Difficult data set training results

Method	Input nodes	Hidden nodes	Output Nodes	Right decisions	Wrong decisions	Time
proposed method	10	6	1	100	0	5.562 s
Bp(batch mode)	10	6	10	86	14	4.291 h
Bp(online mode)	10	6	10	99	1	1.146 h

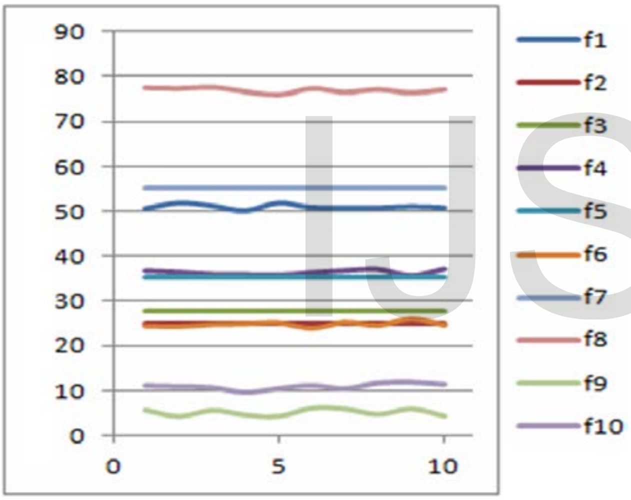

Figure (2): Features behavior in class1 (experiment1)

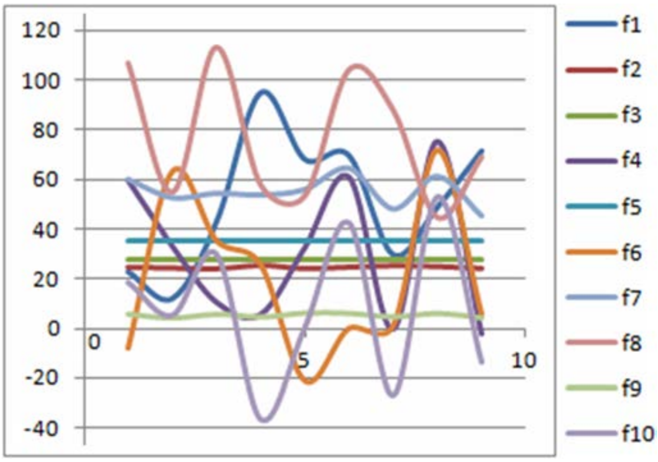

Figure (3): Features behavior in class1 (experiment2)

6 Conclusions

The objective of this paper is to design of pattern classification model based agent's technology implemented as neural networks (CNN) using BP with adaptive learning rate as learning algorithm. To evaluate of this model, we simulated varies cases of classification data with different settings for network factors, the proposed model gives 100% correct classification ratio of the datasets in optimal execution time.

7 Future Work

The only limitation of this research is that it did not use real field data. So our future work is to test the model on a real field of data, also we would try to prune the inputs of the network to eliminate weak features.

Table (2): Simple data set training results

Method	Input nodes	Hidden nodes	Output Nodes	Right decisions	Wrong decisions	Time
proposed method	10	6	1	100	0	3.662 s
Bp(batch mode)	10	6	10	100	0	2.568 h
Bp(online mode)	10	6	10	100	0	0.145 h

8 References

[1] Adel Al-Jumaily , Mohamed Al-Jaafreh ," Multi-Agent System Concepts Theory and Application Phases, Mobile Robotics, Moving Intelligence", Jonas Buchli (Ed.), ISBN: 3-86611-284-X, InTech ,2006.

[2] 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.

[3] 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.

[4] 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.

[5] Devendra K. C.," Soft Computing Techniques and its Applications in Electrical Engineering", Springer-Verlag Berlin Heidelberg, 2008.

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

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

[8] Insung J., Gi-Nam W.," Pattern Classification of Back-Propagation Algorithm Using Exclusive Connecting Network ", International Journal of Electrical and Electronics Engineering, 2008.

[9] Jeanne S.," Intelligent agent technology ", Cork Institute of Technology, Christophe Pincemaille, 2008.

[10] J.V. Capella, A. Bonastre, R. Ors," An Advanced and Distributed Control Architecture .Based on Intelligent Agents and Neural Networks ", IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing System: Technology and Applications, 0-7803-8138-6/03, 2003.

[11] Rui X. , Donald W., " Survey of Clustering Algorithms ", IEEE Transactions on Neural Networks, Vol. 16, No. 3, MAY 2005.

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

[13] Sandhyahe S.," Neural networks from applied sciences and engineering ", Taylor & Francis Group, 2007.

[14] Zhang S., Hu Q., Wang D.," Application of Software Agent to e-Commerce Consumer Buying Support ",IEEE Conference Publishing, 2007.