2009 IEEE International Advance Computing Conference (IACC 2009)
Patiala, India, 6–7 March 2009

Fuzzy Neuro Systems for Machine Learning for Large Data Sets

Rahul Kala , Anupam Shukla , Ritu Tiwari
rahulkalaiiitm@yahoo.co.in, dranupamshukla@gmail.com, rt_twr@yahoo.co.in
Department of Information Technology,
Indian Institute of Information Technology and Management Gwalior

Abstract— **Artificial Neural Networks have found a variety of applications that cover almost every domain. The increasing use of Artificial Neural Networks and machine learning has led to a huge amount of research and making in of large data sets that are used for training purposes. Handwriting recognition, speech recognition, speaker recognition, face recognition are some of the varied areas of applications of artificial neural networks. The larger training data sets are a big boon to these systems as the performance gets better and better with the increase in data sets. The higher training data set although drastically increases the training time. Also it is possible that the artificial neural network does not train at all with the large data sets. This paper proposes a novel concept of dealing with these scenarios. The paper proposes the use of a hierarchical model where the training data set is first clustered into clusters. Each cluster has its own neural network. When an unknown input is given to the system, the system first finds out the cluster to which the input belongs. Then the input is processed by the individual neural network of that system. The general structure of the algorithm is similar to a hybrid system consisting of fuzzy logic and artificial neural network being applied one after the other.**

The system has huge applications in all the areas where Artificial Neural Network is being used extensively. The huge amount of research over these times has resulted in the creation of big databases, which are impractical for a single artificial neural network to train. Also the system takes us closer to the imitation of the human brain which has specialized segments for all sorts of scenarios. In order to test the working of the proposed system, we applied it over a synthetic dataset. The dataset was built by random inputs. We also applied the algorithm to the problem of face recognition. In both the cases, we got a better learning and a higher efficiency using our system. The time required to train the system was also much less than the original single neural network structure. This shows the impact of the algorithm.

Keywords- Machine Learning, Fuzzy Logic, Artificial Neural Networks, Neuro Fuzzy Systems, High dimensionality data, Clustering

I. INTRODUCTION

Artificial Intelligence has covered almost every domain of human life and study. It finds its applications in various industrial domains. A huge reason for this is the working of Artificial Neural Networks. They are gaining popularity due to the good results and ease of applicability. The Artificial Neural Networks and especially backpropagation algorithm has inspired a lot of research and applications. They have been extensively used in speech and speaker identification [18, 21]. Another major area of application is the face recognition [24, 26] and other biometrics. The Handwriting recognition [7, 11] also is a major growing field. Other areas include robotic navigation control [13, 20, 22], language processing, corner detection, biomedical, financial analysis, credit risk analysis, series prediction, noise removal etc. The research over the years in these domains has attracted the increase in the level of objectives of the research and the increase in expectations from the system [27].

As a result there has been an explosion in the training data sets [1, 4, 5, 6, 9, 12, 25, 27]. The number of training data is always kept high to ensure a good performance of the system. The more the training data, the better is the performance of the system. The high amount of training data set has a few disadvantages. The increase in training data badly increases the computation cost. As a result it becomes very difficult for the system to train. Also it is possible that due to the presence of such a high amount of data, the system fails to train itself.

In this paper we propose a novel approach that solves these problems. The approach is especially useful for a data set with high amount of training data. We use a hierarchical approach. Here first the training data set is clustered [10]. We use Fuzzy C Means clustering to form clusters of the given training data set.

Every cluster has a separate neural network. It trains only the data that belong to its cluster. Likewise there are k neural networks that are trained independent of each other. Whenever an unknown input is given, we first find out the most appropriate cluster by measuring its distance from various clusters. Then its output is taken from that particular neural network.

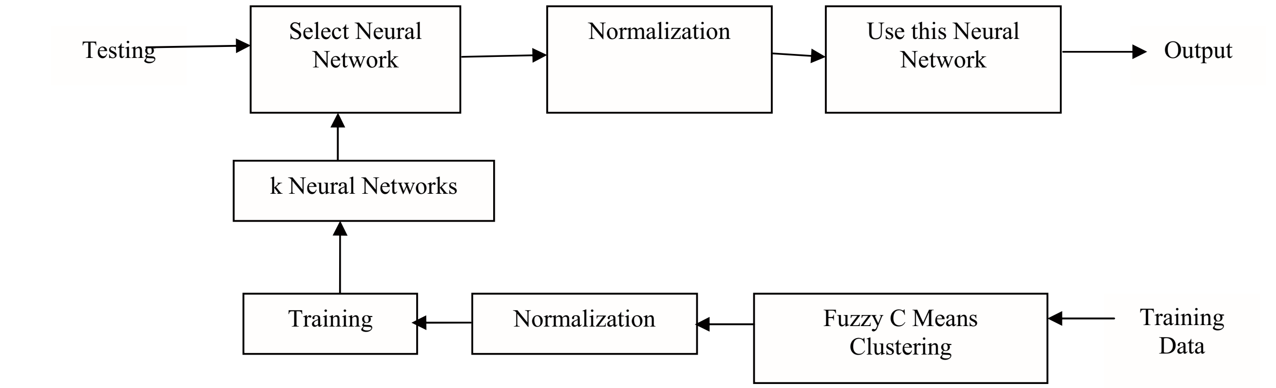

Figure 1: The hierarchical training algorithm

This paper is organized as follows. Section 2 talks about the motivation of the algorithm. Section 3 discusses the algorithm. Section 4 gives the results of the algorithm. Section 5 gives the conclusion.

II. MOTIVATION

Various approaches have been applied in the past for better machine learning. Artificial Neural Networks with back propagation algorithm [7, 11, 13, 18, 20, 21, 24] is one of the most commonly applied approaches. This is a type of supervised machine learning, where the inputs are applied continuously and output is calculated. The error or difference in actual and observed values is back propagated for adjustments of weights. Another common approach is the Kohonen's Self Organizing Maps (SOMs) [19]. This is a type of unsupervised machine learning. Petri Nets and Linear Vector Quantization are other such methods.

Corners' Algorithm [14, 15, 17] is another algorithm that works for instantaneous machine learning. The instantaneously trained neural networks have very low generalization power a very short training time. They are said to be the short term memory of the system.

We also have fuzzy systems that are usually able to solve problems. The neuro fuzzy [8, 16, 23, 26] systems are another way of machine learning. These systems try to build fuzzy logic out of the given data and use back propogation algorithm or steepest descend approach to optimize the model.

Genetic Algorithms [23] and particle swarm optimizations [16] are also being used with Neural Networks for optimization of performance and getting global minima for the system. Various other developments are being applied to improve the mathematical background for various problems [2, 3, 10].

For the reduction in training time of the Artificial Neural Networks, distributed and parallel processing systems [1, 4, 5, 6, 9, 12, 25] are being built. They use various types of partitioning to distribute the whole work of training into multiple terminals.

III. ALGORITHM

In this section we have a look into all the aspects of the working of the algorithm. The major steps of the algorithm are given in Figure 1.

A conventional Neural Network [7, 11, 13, 18, 20, 21, 24] gives the inputs to the system and the output is calculated by the processing by the various nodes of the system. Every time an input is given from the input nodes and the output is collected after processing from the output nodes. The system consists of a number of parallel processing nodes that work on the input to generate the output.

In this system we do not use one, but k number of neural networks. Each of the neural networks caters to the needs of some of the inputs that lie within its range. Each neural network has a known range of inputs that it serves. This is decided by the clustering of the input test data. The numbers of clusters are thus equal to the number of neural networks in the system.

The algorithm has two phases, training and testing. The purpose of training is to make the k Artificial Neural Networks and decide their clusters. The training data is clustered using Fuzzy C Means clustering. This gives us the center of every cluster. We also get the members of every cluster. Every cluster has its own neural network. This neural network is trained independently of all other networks using the members of this cluster. An input is a member of a cluster is its member only when the cluster is the closest to this input in the n dimensional input space. When this process is over for all the k neural networks, we have the networks ready for the testing.

In testing phase, whenever an input is given, we select the best cluster that is closest to the input. We then process the input by this neural network. The general architecture of the system is given in Figure 2.

The algorithm is similar to a hybrid system where the fuzzy logic is applied first and the results are fed into an artificial neural network for the further processing and to generate the output.

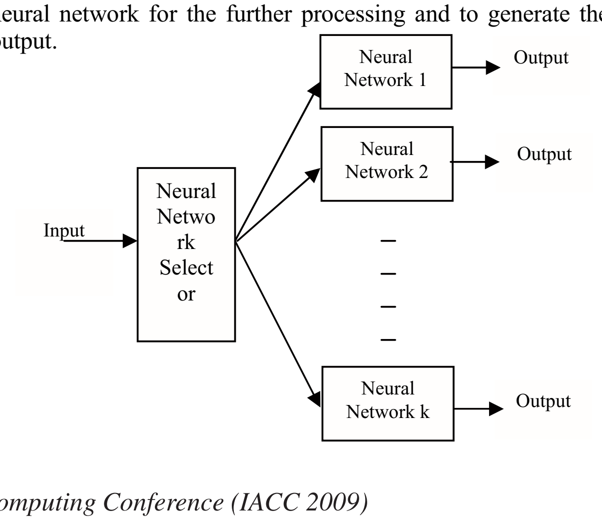

Figure 2: The hierarchical structure of the algorithm

We discuss more about the various phases of the algorithm in detail in the next few subsections. We discuss the two phases of training and testing, one by one.

A. Training

The purpose of this stage is to form the k Neural Networks, each trained by its set of inputs. Each neural network has its own range in which it operates.

Fuzzy C Means Clustering: Fuzzy C means Clustering is a clustering algorithm that is extensively used for clustering of input data in various applications. In this algorithm, we need to specify the number of clusters in advance. The algorithm clusters the input data into these many clusters. We get the centers of the clusters at the end. This clustering algorithm is called a fuzzy algorithm, as any data element is not a member of just one cluster, rather it is associated with more than one clusters each with a different membership degree. This clustering algorithm also gives us the membership degrees of the various inputs. The clustering of any general input is given in figure 3.

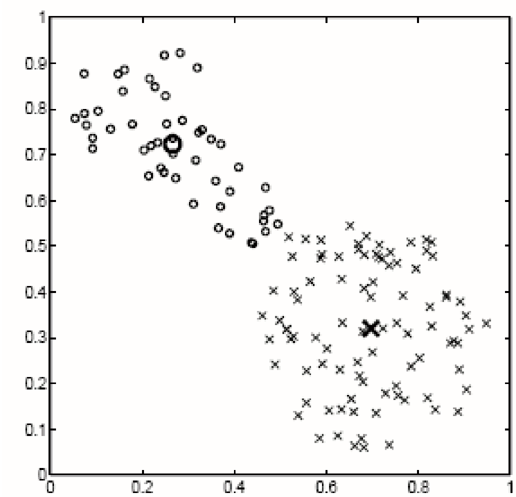

Figure 3: The 2 clusters clustered by the algorithm (shown as large characters)

In our algorithm, we first of all need to decide the number of clusters. This value depends of the level up to which we want the system to be distributed. The higher value of k would result in larger number of neural networks. The inputs would be distributed over a larger number of systems. This would improve the training time. However, a neural network may fail to give good performance if the number of training data in it is too small.

By analyzing the training data set and the system requirements, we decide the value of k. The other parameters that we need to fix in this algorithm are maximum number of iterations, minimum improvement and exponent. Once we run the algorithm, the data is analyzed and the various clusters are formed. At the end of the algorithm, we get to know the centers of the k clusters that we had intended to find.

Suppose that our training data is
$<I_{11}, I_{12}, I_{13}, I_{14} \ldots I_{1n}>$,
$<I_{21}, I_{22}, I_{23}, I_{24} \ldots I_{2n}>$,
…..
$<I_{m1}, I_{m2}, I_{m3}, I_{m4} \ldots I_{mn}>$

Using Fuzzy C Means Clustering we get the k centers as:
$<C_{11}, C_{12}, C_{13}, C_{14} \ldots C_{1n}>$,
$<C_{21}, C_{22}, C_{23}, C_{24} \ldots C_{2n}>$,
…..
$<C_{k1}, C_{k2}, C_{k3}, C_{k4} \ldots C_{kn}>$

Now we allocate every input to a corresponding cluster. This is decided based on the basis of the distance between the input and the cluster. We select the cluster with the least distance from the given input. The distance between any two points $X<x_1, x_2, x_3, x_4 \ldots x_n>$ and $Y<y_1, y_2, y_3, y_4 \ldots y_n>$ is given by (1).

$$d(X,Y) = \sum_{i=1}^{n} (x_i - y_i)^2 \tag{1}$$

The cluster p to which any given input I belongs is given by (2). We select only one value of p.

$$\text{Cluster}(I) = \{ p : d(I,C_p) \le d(I,C_i) \text{ for all } 1 \le i \le k \} \tag{2}$$

We look at all the training data and decide the cluster to which they belong. We group up data and hence at the end each cluster has some input training data associated with it. Through this step, we have clustered the input data into various clusters.

Normalization: The input data may be having very high ranges. Since we are using backpropagation algorithm, we need to make sure that data in every step lies within workable ranges. If the input data is very high, it is possible that it may overshoot the workable range in any step of internal processing. This is a common observation while working with the Artificial Neural Network toolkit of MATLAB. If the input data sizes are even slightly higher, the results are badly affected.

Hence it is usually preferred to keep all inputs between -1 to 1. This improves the performance. In order to make sure that the inputs are within range, we apply a step of normalization. In this step every input is divided by the maximum value that the input can take. Usually we take this maximum value as the maximum value that is present in the training input. This is given by equation (3).

$$I_{new} = \frac{I_{old}}{\text{Max}_{i=1}^{\text{cluster size}}(|I_i|)} \tag{3}$$

If the outputs are also quite large, then we need to normalize them as well. This can be done by dividing them with the maximum output, as we did for the input. This would ensure that the output is within the range of –1 to 1, under which the artificial neural network would be able to perform well.

These maximum values need to be preserved, as we would need them in the testing phase as well. While testing also, whenever an input is applied, we first find out the cluster to which it belongs. Then we divide the input by the same value as we did for all the training data of that cluster.

Training: The most commonly used method for training is the back propagation algorithm. In this algorithm, the inputs are applied one after the other and the outputs are calculated. The actual results to each of these inputs are known. The difference is calculated between the actual and the observed outputs. This error is back propagated to all the layers. The layers uses the concept of steepest descend to calculate the new weights. The weights are changed to the new weights. Next time when same inputs are applied, the errors are smaller. This process is repeated for a few iterations or epochs.

We have the k clusters ready, each with a set of normalized data sets from the previous step. We use the back propagation algorithm in all these k neural networks and train them independently. The limited inputs are given and hence training is a lot better and faster.

B. Testing

Once the k Neural Networks are ready, we need to test them for unknown inputs. This is done in the testing phase of the algorithm.

Selecting the Cluster: We have k neural networks that are ready. We need to select one of them. This selection is done by finding the cluster that is closest to the given input. This selection is the same as given in equations (1) and (2).

Normalization: As discussed above, we need to normalize the input as we did for the training data. Every cluster had a maximum value per input which we used to divide the input to make it lie within the range of -1 to 1. We divide the input by the same value as we did for all the training data of that cluster. This value is given in (3).

Use of Neural Network: We have selected the Neural Network and the normalized input to be processed. We now apply the input to this neural network using the forward pass of the artificial neural network. The various nodes process the inputs and pass the results to the subsequent layers. The final answer is given by the output node. This is the output to the given input that may be interpreted as per the requirements of the system.

IV. RESULTS

In order to test the working of the algorithm, we tested it both for the synthetic data and the face recognition data set. We give both the results in the next sections.

A. Synthetic Data Set

We generated a random problem where the system was to make a Yes/No decision, based on the 5 inputs given to it. We generated 1000 random cases comprising of 5 random numbers as input and a random binary output. We used 4 clusters in clustering of these inputs. The clusters generated are shown in figure 4. The 4 neural networks were trained independent of each other and at the end of training 4 neural networks were ready.

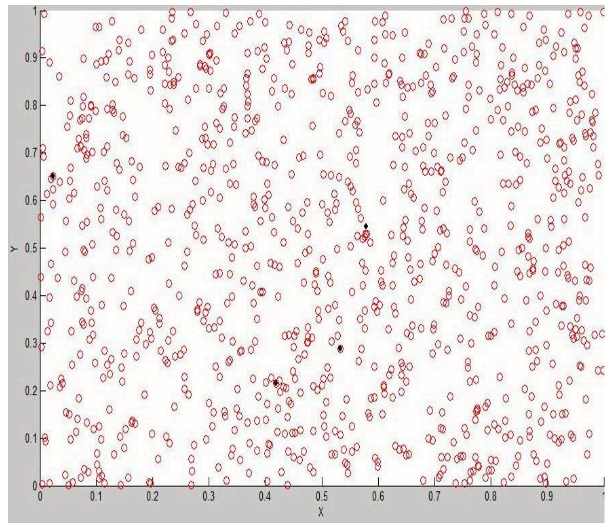

Figure 4: The 4 centers of clusters formed from training input (shown by black dots)

We then tested the system. It was observed that by using only the conventional Neural Network, the performance was found to be 71.1% correct. But by using this algorithm we got an accuracy of 79.1%. The increase in accuracy clearly shows the impact of the algorithm. Also the time needed to train the neural network was much less in the latter.

B. Face Recognition Data Set

In this dataset, we were needed to identify the faces of people. The various faces are taken, in form of images and preprocessed to correct their size and extract the relevant parts of the images. The features are extracted using the image processing toolbox in MATLAB. The features that were extracted were distance between two eyes, length and width of mouth and eye, position of mouth and eye, distance between eye and mouth. These features were used as the inputs to the systems.

We classified the inputs into a set of 2 clusters and distributed the input data set into these 2 clusters. These were trained independent of each other. While training we observed that the number of epochs that were required was much less than the original system. We also observed the system getting trained a lot more easily. Hence this system performed much better than our original system of single neural network.

The performance of the single neural network was measured to be 87.5%. After applying the clustering, the performance got raised to 92.5%. This shows that the system is able to perform much better as a result of clustering and the increase in performance is observed.

V. CONCLUSION

In this paper we saw that by using a hierarchical nature, we have successfully implemented an intelligent system. The system was much easier to train and gave better performance. The system was easily able to solve the face identification problem. The system hence can be easily used in place of the conventional Neural Networks. This system can be a big boon for the systems where the training data sets are too large in size. They can be used in all the problems that currently use Back Propagation Algorithm or other algorithms in Artificial Neural Networks.

In this system, we first used Fuzzy C Means clustering to classify the whole data into data clusters. This gave each cluster a range of inputs which it works upon. Each cluster has its own Artificial Neural Network. Each cluster is trained using the training data sets that belong to it. The training is carried out using the Back Propagation algorithm. At the end of the testing phase we have k Neural Networks that are ready to perform.

At the testing phase we have an unknown input. We first find the cluster that would be processing this input. This is the cluster with the least distance from the input. Then we use the Neural Network corresponding to the selected cluster to process the input and find out the output to this input.

In this paper we have formed a system where we cluster the data and hence setup a hierarchical structure of the system. It may be possible sometimes that the training data becomes too low in some clusters. This may force re-clustering with some restrictions. The actual method of the same is yet to be studied. It is also preferred for the clusters to be of equal size due to performance reasons. The clustering algorithm may be modified to incorporate these changes. Also we need to make some means to identify up to what limit we can cluster the data. There needs to be an optimal value of the parameter k as both the extremes will reduce the system performance. This also needs to be studied in future.

REFERENCES

[1] Alves, R.Lde.S.; de Melo, J.D.; Neto, A.D.D. and Albuquerque, A.C.M.L, "New parallel algorithms for back-propagation learning", Proceedings of the 2002 *International Joint Conference on Neural Networks*, 2002. IJCNN '02, pp 2686-2691, 2002

[2] Amin, Md. Faijul; Murase, K., Single-layered complex-valued neural network for real-valued classification problems, Neurocomputing (2008), doi:10.1016/j.neucom.2008.04.006

[3] Ang, J.H. et al., Training neural networks for classification using growth probability-based evolution, Neurocomputing (2008), doi:10.1016/j.neucom.2007.10.011

[4] Azari, N.G. and Lee, S.-Y, Hybrid partitioning for particle-in-cell simulation on shared memory systems, Proc. of 11th *International Conference on Distributed Computing Systems*, 20-24 May 1991, pp. 526-533.

[5] Babii, Sorin; Cretu, Vladimir; Petriu, Emil M., "Performance Evaluation of Two Distributed BackPropagation Implementations", Proceedings of *International Joint Conference on Neural Networks*, Orlando, Florida, USA, August 12-17, 2007

[6] Campobello, Giuseppe; Patane, Giuseppe and Russo, Marco, "An efficient algorithm for parallel distributed unsupervised learning", *Neurocomputing*, Volume 71, Issues 13-15, August 2008, Pages 2914-2928

[7] Draghici Sorin, "A neural network based artificial vision system for licence plate recognition, *international Journal of Network Security, International Journal of Neural Systems*, Vol. 8, No. 1, 1997

[8] Er, Meng Joo; Zhou, Yi, 'A novel framework for automatic generation of fuzzy neural networks', *Neurocomputing,* 71 (2008) 584–591

[9] [9] Estevez, Pablo A; Paugam-Moisy, Helene, Puzenat , Didier , Ugarte, Manuel, "A scalable parallel algorithm for training a hierarchical mixture of neural experts", *Parallel Computing* 28 (2002) 861–891

[10] Feng, Zhonghui; Zhou, Bing and Shen, Junyi, "A parallel hierarchical clustering algorithm for PCs cluster system", *Neurocomputing* Volume 70, Issues 4-6, January 2007, Pages 809-818

[11] Graves Alex, Fernandez Santiago, Liwicki Marcus, Bunke Horst, Schmidhuber Jurgen, „Unconstrained Online Handwriting Recognition with Recurrent Neural Networks, *Advances in Neural Information Processing Systems* 20, 2008

[12] Hanzalek Zdenek, "A Parallel algorithm for gradient training of feedforward neural networks", *Pattern Computing*, 24(1998), 823-839

[13] Jia, Zhen, Balasuriya, Arjuna and Challa, Subhash, " Sensor fusion-based visual target tracking for autonomous vehicles with the out-of-sequence measurements solution", *Robotics and Autonomous Systems* Volume 56, Issue 2, 29 February 2008, Pages 157-176

[14] Kak, S.C.;" New algorithms for training feedforward neural networks", *Pattern Recognition Letters*, 15, 295-298 (1994).

[15] Kak, Subhash C.," On generalization by neural networks", *ELSEVIER Information Sciences* 111 (1998) 293-302

[16] Lin, Cheng-Jian; Hong, Shang-Jin, 'The design of neuro-fuzzy networks using particle swarm optimization and recursive singular value decomposition', *Neurocomputing* 71 (2007) 297–310

[17] Madineni, K.B., " Two corner classi¯cation algorithms for training the Kak feedforward neural network". *Information Sciences* 81, 229-234 (1994).

[18] Mohanty, S.; Bhattacharya, S; 'Recognition of Voice signals for Oriya Language using wavelet Neural Network', *ACM International Journal of Expert Systems with Applications*, Vol 34, Issue 3, pp 2130-2147, April 2008

[19] Ozdzynski, Piotr; Lin, Andy; Liljeholm, Mimi and Beatty, Jackson, "A parallel general implementation of Kohonen's self-organizing map algorithm: performance and scalability", *Neurocomputing* Volumes 44-46, June 2002, Pages 567-571

[20] Pagac, D., Nebot, E. M. and Durrant. W., H., "An evidential approach to map building for autonomous robots," *IEEE Trans. On Robotics and Automation*, vol.14, no.2, pp. 623-629, Aug. 1998.

[21] Papakostas, G. A., Karras D. A., Mertzios B. G., and Boutalis, Y. S, 'An Efficient Feature Extraction Methodology for Computer Vision Applications using Wavelet Compressed Zernike Moments', *ACM International Journal of Information Sciences*, Vol 177, Issue 13, 2007

[22] Purwin, Oliver; D'Andrea, Raffaello; Lee, Jin-Woo; "Theory and implementation of path planning by negotiation for decentralized agents ", *Robotics and Autonomous Systems* Volume 56, Issue 5, 31 May 2008, Pages 422-436

[23] Sandhu , Parvinder Singh; Salaria, Dalwinder Singh; Singh , Hardeep, 'A Comparative Analysis of Fuzzy, Neuro-Fuzzy and Fuzzy-GA Based Approaches for Software Reusability Evaluation', *Proceedings of Worls Academy of Science, Engineering and Technology* Volume 29 May, 2008, ISSN 1307-6884

[24] Shukla, Anupam; Tiwari, Ritu, 'Fusion of Face and Speech Features with Artificial Neural Network for Speaker Authentication', *IETE Technical Review*, Vol 24, No 5, September-October 2007, pp 359-368

[25] Suresh, S.; Omkar, S.N. and Mani, V, "Parallel Implementation of Back-Propagation Algorithm in Networks of Workstations", *IEEE Transactions on Parallel and Distributed Systems*, Vol 16, No 1, pp 24-34, January, 2005

[26] Taur, J.S.; Tao, C.W., 'A New Neuro-Fuzzy Classifier with Application to On-Line Face Detection and Recognition', *Journal of VLSI Signal Processing* 26, 397–409, 2000

[27] Vieira, Armando; Barradas, Nuno; 'A training algorithm for classification of high-dimensional data', *Neurocomputing* 50 (2003) 461 –472