An Experiment in Task Decomposition and Ensembling for a Modular Artificial Neural Network

Brent Ferguson, Ranadhir Ghosh, and John Yearwood

School of Information Technology and Mathematical Sciences
University of Ballarat,
PO Box 663, Ballarat, Victoria 3353.
bferguson@students.ballarat.edu.au

Abstract. Modular neural networks have the possibility of overcoming common scalability and interference problems experienced by fully connected neural networks when applied to large databases. In this paper we trial an approach to constructing modular ANN's for a very large problem from CEDAR for the classification of handwritten characters. In our approach, we apply progressive task decomposition methods based upon clustering and regression techniques to find modules. We then test methods for combining the modules into ensembles and compare their structural characteristics and classification performance with that of an ANN having a fully connected topology. The results reveal improvements to classification rates as well as network topologies for this problem.

Keywords: Neural networks, modular neural networks, stepwise regression, clustering, task decomposition.

1 Introduction

Feed forward neural networks that have fully connected topologies have in the past had successful application in the problem areas of classification and regression. However their success in part favors databases where the data predominantly describes its classification as a clear function of its features. In addition it has been established in the studies of Quinlan and Collier [12], [4] that neural networks also require conditions of high feature independence to learn optimally. The use of these networks is therefore dependant upon certain conditions that the data may present itself. Adverse conditions that may degrade a neural networks performance may be resolved by appropriate structuring of the network topology.

In recent times, the structured topologies of modular artificial neural networks have attracted growing interest for overcoming the problems encountered by fully connected networks. This interest especially follows from the spiralling accumulation of complex and high dimensional data and the potential for the extraction of useful knowledge from it [5], [6], [10]. The development of modular artificial neural networks in recent reported studies suggest their appropriateness for their application in these circumstances. Modular artificial neural networks can reduce the complexity in problems arising from data having a multiple function quality that cause the condition of learning interference that occurs within fully connected topologies. The difficulty to date has been to find a means to suitably modularize network topology for large and difficult problems.

This work experiments with an approach based upon task decomposition. This is where a large task for solving a large problem is represented in terms of a number of sub tasks. Each sub task becomes a module that specialises in some part of the problem. There are two questions that arise with any approach being:

- how to decompose a problem where little or no knowledge exists?
- how many subtasks are sufficient for adequate solving of the problem?

We contribute towards answering these questions in our approach.

In this work, task decomposition is considered as the decomposition of a complex function into a number of simpler functions. Each of these smaller functions becomes the modules for our ensembles. The task decomposing process we will follow will use commonly used data processing conventions of clustering and regression and include the following operations:

- data selection
- feature selection
- feature refinement

We experiment with different methods for defining modules formed through this process and assess how they contribute overall to the performance of an ensemble of modules for several sub tasks. Where there are several modules trained for the same subtask, we trial three methods for their combination and compare the result for generalizing over test data. We then ensemble modules trained for different subtasks using a small neural network to combine their outputs and compare their characteristics with that of a fully connected neural network. The purpose of this work is to find a basis for a modularising process for large and complex tasks and an indication for the extent of task decomposition to achieve optimal classification rates. For our experiments we have chosen the problem of handwritten character classification for its suitable size and complexity.

2 Background

The early work of Rueckl and Cave [13] demonstrated the increased learning efficiency a neural network has with a modularized topology compared with one that was fully connected when learning the `what and where task'. This problem concerned itself with the recognition of a character that may present itself in one of many orientations that were represented within a grid of pixels. It was discovered that by separating the problem into two sub problems of recognition and location and by organizing the network's topology into two modules having separate hidden layers, the network benefited from improved learning and was able to classify more effectively. Following this work, there have been many investigations over the past decade into how to structure a network's topology in terms of modules to suit a variety of problems.

Schmidt [14] observes that networks of modules are either one of two types. Networks can be composed of multiple networks called modules where each has learned a separate part of the problem. These networks are commonly referred to as a mixture of experts. The outputs of each expert network are combined in a decision process that determines the contribution of each to the overall problem. The decision process may base itself upon a statistical approach such as the majority voting principle or a neural network that learns each expert's contribution such as the gating network described by Jacobs [9]. The other type of modular network described by Schmidt are those that are generally considered as not fully connected. Modules within these networks are defined as regions of the network that appear more densely connected and which are loosely connected to one another. Such networks are typical of those networks whose architectures are found though an evolutionary process. Examples of these networks appear in the work of Boers and Kuipers [2], [3], and Pal and Mitra [11]. In these networks the distribution of connections is the result of an applied genetic algorithm searching for optimal connectivity between neurons or sub network structures.

Although modular neural networks can be generally described as having a refined topology this may not necessarily mean that these networks are simplifications of the fully connected network, sometimes referred to as a monolithic neural network. Happel [8] demonstrates that for some problems, an increase in connections between neurons is needed to achieve effective learning, which results in highly complex architectures. On the other hand Boers [2] has found that modularization may not provide the most optimum topology. Amongst the population of network structures evolved for Rueckl's [13] `what' and `where' problem, a simple 2 layered artmap had produced a result similar to Rueckl's original modular solution.

In this work we organize modules into ensembles where their outputs are input to a single combining module in a similar style to the mixture of experts model. This work concentrates on the problem of modularizing the task of alphabetic character classification.

A single definition for modularity within a modular artificial neural network has not as yet been found. Modularity has mostly resulted from a task decomposition process where an overall task is divided into subtasks. A module that is implemented by a small MLP represents each subtask. An overview of the most common approaches to task decomposition and designs for modular neural networks is given by Auda and Kamel [1].

What has been consistently emerging from this area of research is their suitability for application to very large databases. Schmidt [14] has demonstrated that even by choosing the modules within a network random selection and optimization, the tendency to achieve a more efficient network grows with the size of the dataset. According to Boers [2], these networks generally have better learning efficiency with respect to training times and training stability. In general, the modular neural network achieves higher classification rates but this is dependant not only upon satisfactorily defining the modules but also according to Auda and Camel [1], on how they are to be trained and connected. Depending on one's approach on how to modularize, the result should be in terms of a highly structured topology.

Modular neural network technology may also benefit another area of research where an efficient and effective means is sought to extract an explanation from an artificial neural network trained in a particular task. A trained neural network can be regarded as a black box. How it makes a decision based on its inputs is unclear. Its knowledge is incomprehensibly represented by weight values and transfer functions. This field of research attempts to translate this sub symbolic state into interpretable rules. One of the problems here to be overcome however is also that of scalability. Large problems incur the extraction of numerous rules and the search for relevant rules becomes increasingly difficult computationally. In short, Craven [5] and Golea

[7], explain that current rule extraction techniques struggle with large fully connected networks to find compact sets of understandable rules. Pal [11] asserts that the extraction of rules can be greatly assisted by a more structured network and demonstrates the effectiveness of his modularizing technique based upon a soft computing framework of hybridized technologies.

3 Methodology

Our approach to decomposing the task of character recognition is to create modules or small neural networks that are dedicated to recognizing a particular character and that its function can distinguish this character from others. We estimate that by correctly defining the modules to preform their task then an ensemble of modules coupled with a decision network should classify characters with a greater accuracy than a fully connected neural network would. We examine the results of our experiments at two levels:

- The first level compares four cases of modularisation. The cases are sequenced to allow a series of operations to be added. These operations involve applications of clustering and regression techniques to condition data and find the inputs for modules.
- The second level begins with training a benchmark fully connected neural network trained to classify characters but primarily concerns itself with creating and training ensembles of character modules found for each case. The results for these ensembles are then compared with that obtained for the benchmark.

We analyse the results at both levels to relate the progression of module development to gains in ensemble classification accuracy. We also observe the details of topology both for character modules and for the ensembles such as the number of inputs, hid den neurons and connections they have to perform their task to assess the limits for our modularizing process.

For our experiments, we initially chose to scale the classification task of 26 alphabetic characters to 8 characters and we report the results relative to them. This subset of characters has been decided upon to contain those characters that are most represented by number of examples. Altogether there were 2462 examples to represent the 8 characters that were divided into 1462 examples for the training set and 1000 examples for the test set. The results for the experiments were averaged over 10 trials where training and test sets were drawn from randomized examples.

Module training details – Modules are implemented as three layered, feed forward and fully connected neural networks that use a sigmoid transfer function. Matlab 6.5 was used to train networks of modules in a PC environment using resilient backpropagation. The hidden layer for each neural network was grown using a succession of training cycles to add additional neurons. The number of hidden neurons was fixed when no further improvement in classification accuracy was observed.

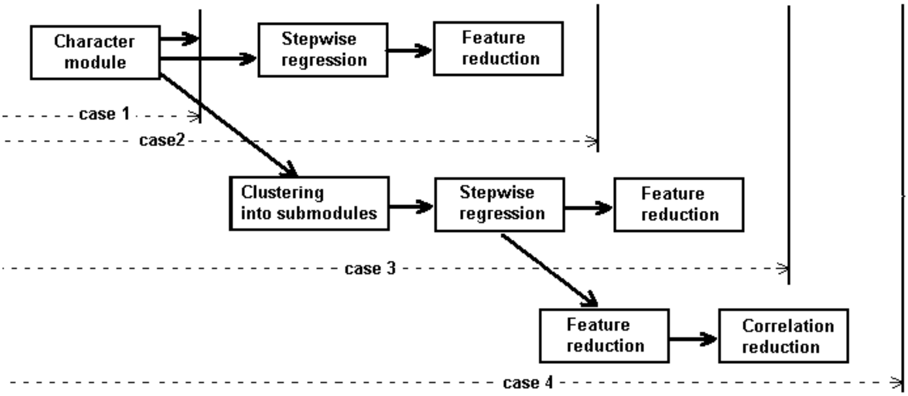


Fig. 1. Progressive module development at each case

Level 1 - Details of Module Definition Experiments

Case1 - Modularization at the character level. Modules are created for each character using all of the 100 available features of the feature vector for inputs. Each character module is trained with the training set where the examples for this character are distinguished from the others.

Case 2 - Modularization at the character level with feature reduction. In this case modules are created for each character with a reduced number of inputs. This follows with the application of a stepwise regression algorithm to find the most relevant subset of features that are associated with the examples for a particular character. The algorithm proceeds in steps in entry mode that adds features one at a time if the significance of the subset is improved by 0.05 or more and rejects the inclusion of a feature if the significance alters in excess of 0.1.

Case 3 - Modularization within character level with feature reduction. This case explores modularization within the examples for a character. Fig. 1 illustrates the process for defining case 3 modules. The examples are clustered with a self organizing map SOM into 4 groups. The value of 4 has been chosen to reflect a moderate number of clusters and is also influenced by the estimated number of handwriting styles that a character may be formed with. It is only desired to observe the usefulness of clustering at this point in developing our approach and the determination of an optimal number will be left for future consideration. The inputs for each cluster are found through stepwise regression similarly to the modules of case 2. For the purposes of regression, the examples for this cluster are distinguished from the remainder of the training and test sets, which also includes the examples from the other 3 clusters. Modules so defined become the submodules of the decision module. Each submodule undergoes the training process until complete. At this stage the training set is propagated through the submodules where the decision module trains to associate their outputs with the character class.

Case 4 - Modularization within character level with feature reduction and correlation reduction. Case 4 modules have been defined similarly as for case 3 modules with the addition of further reduction of the input feature set by removing those features that are highly correlated to another. Referring to fig. 1 for case 4 module definition an additional process followed feature reduction by stepwise regression. This consideration investigates the conditions with which correlated features can be removed. This process involves searching a correlation matrix produced for the feature subset found in the previous step for highly correlated feature pairs. Three experimental rules were constructed for deciding which feature should be removed from the subset.

Let A, B be feature pairs having a correlation in excess of 0.7.

- If A is correlated to any other feature above lower limit=0.5 then remove A from feature subset.
- If B is correlated to any other feature above lower limit=0.5 then remove B from feature subset.
- A or B is not correlated to any other feature above lower limit then remove A from feature set if P value greater than B otherwise remove B.

The upper and lower correlation limits have been set by trial and error in prior experimental determination.

Comparison study. Modules for a particular character developed in each of the four cases are compared for their classification accuracy and structural details. That is the number of hidden neurons and network connections there are for the module to perform its subtask.

Where there are several modules existing for the same subtask such as the submodules of case 3 and case 4. A means is sought to combine their outputs into one so that a comparison can be made with case 1 and case 2 modules. We trial three different methods and select the highest classifying one for use as a decision module in our comparison tests. The three methods are outlined in fig. 2.

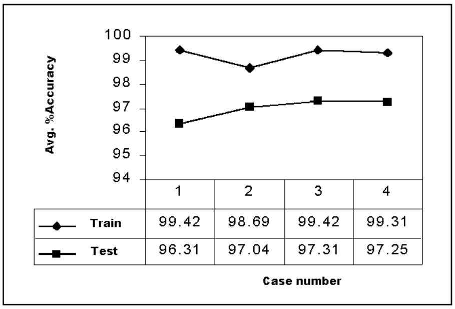
- Combining method 1: is based upon the majority vote principle. On the basis of two or more submodule outputs that strongly indicate the classification of a particular character, the higher value is output from the combining process otherwise the lowest value is output.
- Combining method 2: the outputs are multiplied by a weighted value. This value results from the number of training set examples there are from the clustering process that defines this submodule divided by the total number of examples for the four submodules.
- Combining method 3: a small neural network inputs the submodule outputs and trains to associate the inputs with an output character classification.

Level 2 - Module Ensemble Experiments

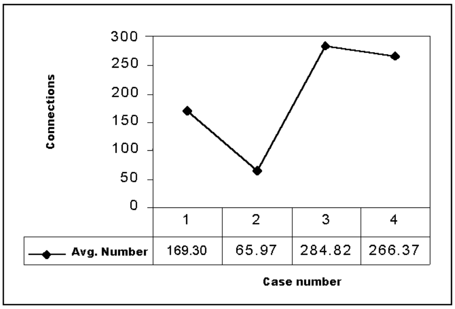
Level 2 looked at propagating the accuracy obtained for modules at the character level to the ensemble level. Experiments were conducted to train ensembles of all eight character modules found for each case and compare the training and test set accuracy for each case. To combine the modules for each case, a neural network module inputs their outputs and trains to learn to associate their output status with one of eight character classifications. For comparison purposes a fully connected neural network was trained with the same dataset and serves as a benchmark.

4 Results and Discussion

The tasks at level 1 to evaluate modules resulting from methods described for case 1 to case 4 modularisation with reference to fig. 2, reveal overall a tendency for improved test set accuracy for each of the cases with the exception of case 4 having a slight decrease. The immediate reduction of average connection numbers referring to fig. 3 in case 2 to that of case 1 is expected as the result of feature selection due to regression. This trend is not carried through to case 3 and case 4 where both cases show large increases in connections relative to case 1 modules.



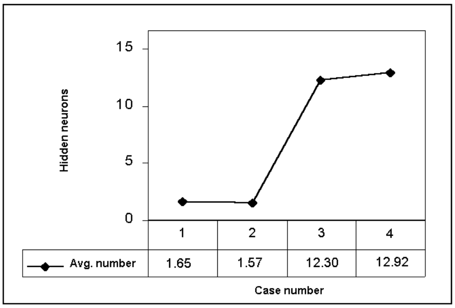
	1	2	3	4
—◆— Train	99.42	98.69	99.42	99.31
—■— Test	96.31	97.04	97.31	97.25



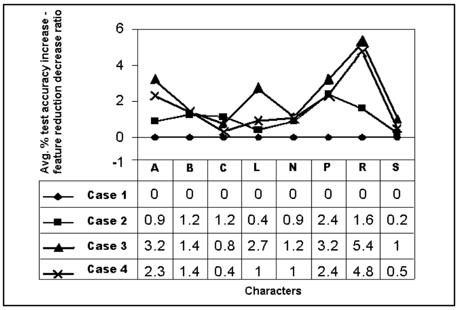
	1	2	3	4
—◆— Avg. Number	169.30	65.97	284.82	266.37

Fig. 2. Avg. module accuracy

Fig. 3. Avg. module connections



	1	2	3	4
—◆— Avg. number	1.65	1.57	12.30	12.92

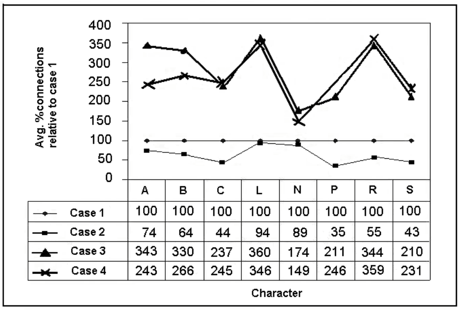


	A	B	C	L	N	P	R	S
—◆— Case 1	0	0	0	0	0	0	0	0
—■— Case 2	0.9	1.2	1.2	0.4	0.9	2.4	1.6	0.2
—▲— Case 3	3.2	1.4	0.8	2.7	1.2	3.2	5.4	1
—✕— Case 4	2.3	1.4	0.4	1	1	2.4	4.8	0.5

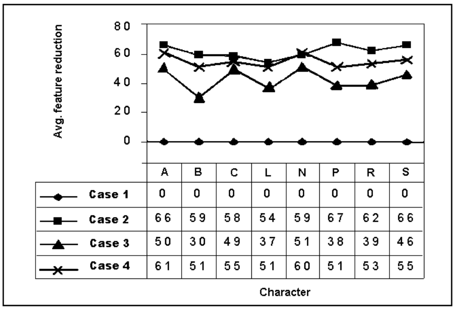
Fig. 4. Avg. hidden neurons

Fig. 5. Avg. reduction to features

This trend is accompanied by a sharp increase in the number of hidden neurons for both cases as indicated in fig. 4. However it was noted that the numbers of hidden neurons in a submodule of case 3 or case 4 be comparable to the number in a case 2 module. When considering case 3 submodules together, the total number of hidden neurons should therefore be optimally less than or equal to four times the number in a case 2 module. The average number of hidden neurons observed is greater than six times the number, which may suggest the presence of a false cluster. This circumstance may well improve by decreasing the number of clusters.



	A	B	C	L	N	P	R	S
Case 1	100	100	100	100	100	100	100	100
Case 2	74	64	44	94	89	35	55	43
Case 3	343	330	237	360	174	211	344	210
Case 4	243	266	245	346	149	246	359	231

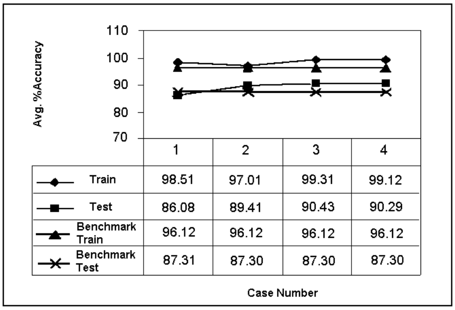


	A	B	C	L	N	P	R	S
Case 1	0	0	0	0	0	0	0	0
Case 2	66	59	58	54	59	67	62	66
Case 3	50	30	49	37	51	38	39	46
Case 4	61	51	55	51	60	51	53	55

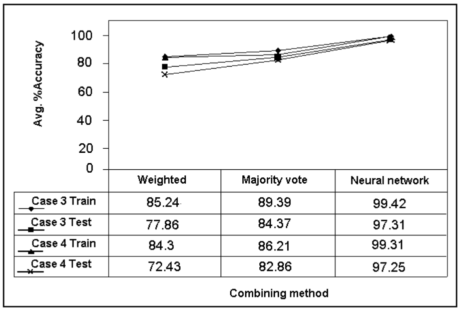
Fig. 6. Avg. improvement to connections

Fig. 7. Avg. feature reduction

When assessing the possible gain in terms of test accuracy improvement to feature reduction for modules relative to case 1 in fig. 6, case 4 modules appeared to follow the trend of case 3 modules for six of the eight characters with a slightly lower ratio. Considering this and the improved feature reduction of these modules over case 3 modules and being closer to that of case 2 modules that is indicated in fig. 7, case 4 modules may perform better given an optimal number for the submodule clustering. This would verify that further reduction of features from the input space on the basis of their correlation with other features is plausible but requires further investigation. The test classification to feature reduction ratio plotted for the four cases of character module in fig.5 indicates overall in favour of case 3 module development.



	1	2	3	4
Train	98.51	97.01	99.31	99.12
Test	86.08	89.41	90.43	90.29
Benchmark Train	96.12	96.12	96.12	96.12
Benchmark Test	87.31	87.30	87.30	87.30



	Weighted	Majority vote	Neural network
Case 3 Train	85.24	89.39	99.42
Case 3 Test	77.86	84.37	97.31
Case 4 Train	84.3	86.21	99.31
Case 4 Test	72.43	82.86	97.25

Fig. 8. Avg. ensemble accuracy

Fig. 9. Avg. Combination accuracy

Combining the sub modules of case 3 and case 4 immediately shows a marked loss of test set accuracy for both the summing of weighted inputs method and the majority voting process. See fig. 8. Although the voting process improves the result over the weighting method, it does not preserve the learning at the sub module level. Linear recombination of sub module outputs does not appear to be supported for this dataset. The use of a neural network to combine the submodule outputs outperformed the other two methods and was the choice to use for the comparison tests between cases.

In level 2 experiments for module ensembles, the test set results in fig. 8 indicate improving accuracy from case 1 through to case 3. This trend also improves upon the benchmark result.

5 Conclusion

In this study we have applied our approach to successfully decompose the problem of handwritten alphabetic character classification. We have decomposed the transitional representation of this dataset into components we refer to as modules. The ensembles of modules found for each defining case, show a comparative increase in test set classification accuracy favouring modules found using both clustering and step wise regression to those found using only regression. Further, the ensembles improve upon the benchmark fully connected neural network when comparing test set accuracy and topology. The case three ensemble having an average test set accuracy of 90.43% and the benchmark having 87.3%. The case four ensembles show the likely possibility for improved structure with a comparative reduction of inputs with a reasonable preservation of test accuracy. At present you task decomposition approach is limiting at case 3 modularisation.

The result of the trial that compared methods for combining submodules developed for the same task showed in favour of using a small neural network to combine the submodule inputs. Case three modules produced an indicative comparative result with an average test score of 99.42% compared with 89.39% for the majority vote and 85.24% for weighted contribution.

In summing our approach we follow a two fold process of module creation and module refinement. The approach separates the examples associated with each character into subsets. Submodules are formed initially by clustering each subset and then a step-wise regression procedure is applied to refine the module inputs. Suitably sized neural networks are then found to represent the submodules and to combine their outputs, for both submodules having a common subtask and for ensembles of modules having different subtasks

The success of our approach has been observed for the reduced dataset of handwritten characters and the implication for the construction of a complete artificial modular neural network classifier for 26 characters is supported by the results of our experiment.. It is expected that an improvement to both network topology and generalization will result over the use of one large fully connected neural network.

Further work needs to be undertaken to confirm our method to assess its suitability for broader application. Tests will need to be carried out with different representations of data for other problems of varying dimensionality. Our approach is expected to improve from further experimentation to find an optimal number of clusters for each subtask.

References

1. Auda, G., Kamel M., Modular Neural Networks: A Survey, International Journal of NeuralSystems, Vol 9, (1999) 129-151.
2. Boers, E.J.W., Kuipers, H., Biological Metaphors and the Design of Modular Artificial Neural Networks, Masters Thesis, Leiden Univesity, Netherlands, (1992).
3. Boers, E.J.W., Kuipers, H., Happel, B.L.M., Sprinkhuizen-Kuyper, I.G., Designing Modular Artificial Neural Networks, Computing Science in the Netherlands, Proc.(CSN'93),Ed.: H.A.Wijshoff, Strichting Mathematisch Centrum, Amsterdam, (1993) 87-96.

4. Collier, P.A., Waugh, S.G., Characteristics of Data Suitable For Learning With Connectionist and Symbolic Methods, Dept. of Computer Science, University of Tasmania, (1994).
5. Craven, M., Shavlik, J., Rule extraction: Where do we go to from here?, Working paper 99-1, University of Wisconsin Machine Learning Group, (1999).
6. Fayyad, U. M., Data Mining and Knowledge Discovery: Making sense out of Data, IEEE Expert Intelligent Systems and Their Applications, Vol 11, (1996) 20-26.
7. Golea, M., Tickle, A., Andrews, R., Diederich, J., The truth will come to light, IEEE Transactions on Neural Newtorks, Vol 9, (1998) 1057-1068.
8. Happel, B., Murre, J. M., Design and evolution of modular neural network architectures, Neural Networks, Vol 7, (1994) 985-1004.
9. Jacobs, R.A., Jordan, M.I., Barto, A.G., Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks, Cognitive Science, Vol 15, (1989) 219-250.
10. Nayak, R., Intelligent Data Analysis: Issues and Challenges, Proc. of the 6th. World Multi Conferences on Systematics, Cybernetics and Informatics, July 14-18, 2002, Florida USA.
11. Pal, S., Mitra, S., Rough Fuzzy MLP: Modular evolution, rule generation and evaluation, IEEE Transactions Knowledge and Data Engineering, Vol 15, (2003) 14-25.
12. Quinlan J.R., Comparing Connectionist and Symbolic Learning Methods, Dept. of Computer Science, University of Sydney, (1993).
13. Rueckl, J., Cave, K. R., Kosslyn, S. M., Why are `what' and `where' processed by separate cortical visual systems? a computational investigation, Journal of Cognitive Neuroscience, Vol 1, (1989) 171-186.
14. Schmidt, A., A modular neural network architecture with additional generalisation abilities for high dimensional input vectors, Masters Thesis, Manchester Metropolitan University, (1996).