In-situ Learning in Multi-net Systems

Matthew Casey and Khurshid Ahmad

Department of Computing, School of Electronics and Physical Sciences,
University of Surrey, Guildford, Surrey, GU2 7XH, UK
{m.casey, k.ahmad}@surrey.ac.uk

Abstract. Multiple classifier systems based on neural networks can give improved generalisation performance as compared with single classifier systems. We examine collaboration in multi-net systems through *in-situ learning*, exploring how generalisation can be improved through the simultaneous learning in networks and their combination. We present two in-situ trained systems; first, one based upon the simple ensemble, combining supervised networks in parallel, and second, a combination of unsupervised and supervised networks in sequence. Results for these are compared with existing approaches, demonstrating that in-situ trained systems perform better than similar pre-trained systems.

1 Introduction

The task of classifying data has been tackled by a number of different techniques. One such approach is the use of mixture models, which uses a combination of models to summarise a data set comprising a number of modes. Such mixture models are 'parsimonious in the sense that they typically combine distributions that are simple and relatively well-understood' [5] (p.267), of which the *mixture-of-experts* (ME) model is a good example. Mixture models are based on the assumption that each constituent of the mixture can classify one segment of the input, and that the combination is able to classify most, if not all, of the input. Such combinations appear intuitive, and have been used on a number of pattern recognition tasks, such as identity [6] and handwriting recognition [16]. The disadvantage with mixture models is the increase in processing time caused by multiple components, however they have a degree of elegance in that they combine a number of 'simple' classifiers.

The constituent classifier neural networks of a multiple classifier combination are further distinguished as either *ensemble* or *modular*; the former refers to a set of redundant networks, whilst the later has no redundancy (of which ME is an example). Such *multi-net systems* (see papers in [12]) typically combine networks in parallel, but the sequential combination of networks has also had some success [10]. Whether in parallel or in sequence, each constituent network of a multi-net system is combined using prior knowledge of how the combination is affected, exemplified by the pre-training of networks before combination. The question here is whether techniques such as ME, which can *learn* how to combine networks, offers any improvement over individually trained systems? In the context of multiple classifier systems, it is important to look at this *in-situ learning*, defined as the simultaneous training of the constituent networks, which ‘provides an opportunity for the individual networks to interact’ [9] (p.222). In this paper we evaluate the use of in-situ learning in the parallel and sequential combination of networks to help assess this as a general approach to learning in multi-net systems.

2 In-situ Learning in Multi-net Systems

In this paper we consider two multi-net systems that exploit in-situ learning [3]. The first is a simple ensemble (SE) trained in conjunction with early stopping techniques: the *simple learning ensemble* (SLE). The second is a novel system consisting of a group of unsupervised networks and a single supervised network that are trained in sequence: *sequential learning modules* (SLM).

Simple Learning Ensemble: There have been two contrasting examples of in-situ learning in ensembles. Liu and Yao [8] defined the *negative correlation learning* algorithm for ensembles that trains networks in-situ using a modified learning rule with a penalty term, whereas Wanas, Hodge and Kamel’s [14] multi-net system combines partially pre-trained networks before continuing training in-situ. Whilst we agree with Liu and Yao that in-situ learning is important, our work differs from theirs and Wanas et al’s in two respects: first we use the same data set to train all of the networks, rather than using data sampling, and second we use early stopping to promote generalisation through assessing the *combined performance* of the ensemble, instead of introducing a penalty term to the error function, exploiting the interaction between networks [9]. Our approach is based upon the SE, but with each network trained in-situ. We use the generalisation loss [11] early stopping metric to control the amount of training based upon the measured generalisation performance.

Sequential Learning Modules: Sequential in-situ learning is a difficult area to develop for supervised classification because it depends upon having an appropriate error to propagate back through each network in sequence. This issue is apparent in the development of multi-layer, single network systems, where an algorithm such as backpropagation is required to assign error to hidden neurons. Bottou and Gallinari [2] discussed how error can be assigned to sequential networks in multi-net systems, but assumed that each such network used supervised learning. Our approach is to use unsupervised networks in sequence coupled with in-situ learning so that no such error is required, only an appropriate input to each network. We employ networks that use unsupervised learning in all but the last network to give an overall supervised system, but which does not propagate back error. This approach also allows unsupervised techniques to be used to give a definite classification through the assignment of a class by the last network.

3 Evaluating In-situ Learning with Classification

The classification of an arbitrary set of objects is regarded as an important exemplar of learnt behaviour. We use well-known data sets [1], which have been used extensively in benchmarking the performance of classification systems, observing the behaviour of the proposed systems. We use the artificial MONK's problems [13] to test generalisation capability, whilst the Wisconsin Breast Cancer Database (WBCD) [15] is used to test pattern separation capability using real-life data (Table 1).

Table 1. Details of data sets used for experiments. For the MONK's problems, the validation data set includes the training data, which is also used for testing.

Data Set	Input	Output	Training	Validation	Testing	Examples/Class %	Notes
MONK 1	6	1	124	432	-	50:50	
MONK 2	6	1	169	432	-	67:33	
MONK 3	6	1	122	432	-	47:53	5% misclassified
WBCD	9	2	349	175	175	66:34	16 missing values

SLE systems consisting of from 2 to 20 multi-layer perceptrons (MLPs) trained using backpropagation were constructed to determine the effect of ensemble complexity on generalisation performance. Each network within the ensemble had the same network topology, but to generate diversity in the networks, each was initialised with different random real number weights selected using a normal probability distribution with mean 0, standard deviation 1. The backpropagation with momentum algorithm was used with the Logistic Sigmoid activation function, using a constant learning rate of 0.1 and momentum of 0.9.

For the SLM systems, we restrict ourselves to combining a self-organising map (SOM) [7] and a single layer network employing the delta learning rule. Neither of these is capable of solving a non-linearly separable classification problem; our hypothesis is that an in-situ trained combination of these can solve these more complex problems. The basic SOM algorithm was used on a rectangular map of neurons, with a Gaussian neighbourhood and exponential learning rate. To ensure that the output of the SOM can be combined with the single layer network, the output is converted into a vector by concatenating the winning values from each of the neurons, with '1' associated with the winning neuron and '0' for all other neurons. The single layer network using the delta learning rule had a constant learning rate of 0.1, and a binary threshold activation function.

Table 2. The number of input, hidden and output nodes per data set for each of the constituent networks used for the single network and ensemble systems (hidden nodes selected as in [13]).

System	MONK 1	MONK 2	MONK 3	WBCD
MLP MLP (ES) SE (ES) SLE (ES)	6-3-1	6-2-1	6-4-1	9-5-2

In order to understand the generalisation performance of the SLE and SLM systems, we compare the percentage test responses against those generated for single MLPs trained with and without early stopping, as well as simple ensembles formed from 2 to 20 MLPs pre-trained with early stopping. The architecture used for the various systems is shown in Table 2 and Table 3. Each of the systems underwent 100 trials to estimate the mean performance, training either for a fixed 1000 epochs, or with early stopping (ES) for a maximum of 1000 epochs.

Table 3. The different architectures used for the SLM system, shown as the topology of the SOM and the single layer network. For the SOM this is the number of inputs and nodes in the map. For the single layer network this is the number of input and output nodes.

System	MONK 1	MONK 2	MONK 3	WBCD
SLM	6-5x5: 25-1 6-10x10: 100-1 6-20x20: 400-1	6-5x5: 25-1 6-10x10: 100-1 6-20x20: 400-1	6-5x5: 25-1 6-10x10: 100-1 6-20x20: 400-1	9-5x5: 25-2 9-10x10: 100-2 9-20x20 400-2

3.1 Experimental Results

For each of the benchmark data sets, Table 4 shows the percentage mean number of correct test responses for the MLP, SE, SLE and SLM systems. Only the configuration of each system giving the highest mean test percentage is shown.

Table 4. Results for systems with the highest mean test response, with the number of networks / SOM configuration and mean test response, with standard deviation.

System	MONK 1		MONK 2		MONK 3		WBCD	
	Nets	Test %	Nets	Test %	Nets	Test %	Nets	Test %
MLP	1	84.44 ±12.15	1	66.29 ±35.21	1	83.39 ±47.57	1	95.90 ±3.93
MLP (ES)	1	57.13 ± 8.74	1	65.21 ± 2.68	1	63.10 ± 6.83	1	82.34 ±9.61
SE (ES)	3	55.75 ± 7.70	18	66.25 ± 0.81	18	66.03 ±23.10	20	91.94 ±1.69
SLE (ES)	20	**90.21 ± 6.16**	20	69.49 ± 1.24	19	78.57 ± 4.69	20	92.95 ±1.06
SLM	10x10	75.63 ± 4.78	20x20	**75.09 ±26.06**	10x10	**84.10 ± 1.76**	20x20	**97.63 ±0.83**

First we note that for the MONK 1 and 2, the SLE system gives a comparatively better generalisation performance when a relatively large number of networks are combined, with the performance of the SE decreasing with successively more networks. Here a more complex in-situ trained system gives better generalisation, in contrast to the far less complex pre-trained system. For MONK 3 and WBCD, the SLE improves upon the early stopping MLP and SE systems, but not the fixed MLP trained for 1000 epochs. The improvement in generalisation performance can be attributed to the increased training times experienced by the SLE algorithm with increasing numbers of networks as compared with the MLP with early stopping systems. For example, for MONK 1 with 2 networks, the maximum number of epochs is 27 (excluding outliers), which increases to 521 epochs for 20 networks. However, all these are less than the fixed 1000 epochs for the MLP systems, yet give a similar level of performance.

For the SLM system, we note that the sequential combination of networks successfully learns to solve each non-linearly separable task. This is perhaps surprising given that neither is individually capable, and despite the somewhat complex nature of the SLM systems with relatively high numbers of neurons. For MONK 2, 3 and WBCD, the SLM system out-performs the other single network and multi-net systems. For

MONK 1 the results are better than both the SE and MLP with early stopping, but do not improve upon the SLE or fixed MLP.

The results for the SLM also show how the number of neurons within the SOM affects the overall performance of the system, perhaps in a similar way to the number of hidden neurons in an MLP. Here, increasing the map size tends to give both improved training and generalisation performance, reaching a peak commensurate with over-fitting. For MONK 1 the response for the 10x10 map is better than for the 20x20 map, despite giving a 100% training response, as compared with the 10x10 response of 89.65%. Furthermore, increasing the map size also produces more reliable solutions in that the standard deviation decreases, whilst still maintaining a similar level of generalisation performance.

3.2 Discussion

These preliminary results are encouraging, and demonstrate that in-situ learning in parallel and sequential combinations of networks can give improved generalisation performance, as demonstrated by the results for SLE, and especially the SLM systems. Putting these into context with other reported results shows that they compare well, but it is recognised that some further investigation is required.

For the MONK's problems, optimal generalisation results have been reported with 100%, 100% and 97.2% for MONK 1, 2 and 3 respectively [13]. For the SLE systems the maximum values are 98.4%, 74.5% and 83.1%, and for the SLM systems 84.7%, 81.0% and 87.5%, showing that, whilst there is a small spread of values, further tuning is required to improve the maximum. Here, of interest is the way in which the results demonstrate the use of unsupervised learning in a modular system, giving a significant improvement in generalisation as compared with existing supervised techniques (MONK 2 and 3). For the WBCD data set, the SLM system with a mean of 97.63% again out-performs the SE, and is comparable to other multi-net systems such as AdaBoost with 97.6% [4]. Further work is required to assess the properties of these techniques with other data sets, and especially how the combination of unsupervised and supervised learning can be further exploited for classification tasks.

4 Conclusion

In this paper we have explored whether the use of simultaneous, in-situ learning in multi-net systems can provide improved generalisation in classification tasks. In particular, we have presented results for in-situ learning in an ensemble of redundant networks, and the in-situ learning in a sequential system, the latter of which builds upon the principle that 'simple' networks combined in a modular system are parsimonious, through the combination of supervised and unsupervised techniques.

Acknowledgements. The authors would like to thank Antony Browne and the two anonymous reviewers for their helpful comments.

References

1. Blake,C.L. & Merz,C.J. *UCI Repository of Machine Learning Databases*. http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA.: University of California, Irvine, Department of Information and Computer Sciences, 1998.
2. Bottou, L. & Gallinari, P. A Framework for the Cooperation of Learning Algorithms. In Lippmann, R.P., Moody, J.E. & Touretzky, D.S. (Ed), *Advances in Neural Information Processing Systems*, vol. 3, pp. 781-788, 1991.
3. Casey, M.C. *Integrated Learning in Multi-net Systems*. Unpublished doctoral thesis. Guildford, UK: University of Surrey, 2004.
4. Drucker, H. Boosting Using Neural Networks. In Sharkey, A. J. C. (Ed), *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, pp. 51-78. London: Springer-Verlag, 1999.
5. Jacobs, R.A. & Tanner, M. Mixtures of X. In Sharkey, A. J. C. (Ed), *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, pp. 267-295. Berlin, Heidelberg, New York: Springer-Verlag, 1999.
6. Kittler, J., Hatef, M., Duin, R.P.W. & Matas, J. On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20(3), pp. 226-239, 1998.
7. Kohonen, T. Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.
8. Liu, Y. & Yao, X. Ensemble Learning via Negative Correlation. *Neural Networks*, vol. 12(10), pp. 1399-1404, 1999.
9. Liu, Y., Yao, X., Zhao, Q. & Higuchi, T. An Experimental Comparison of Neural Network Ensemble Learning Methods on Decision Boundaries. *Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN'02)*, vol. 1, pp. 221-226. Los Alamitos, CA: IEEE Computer Society Press, 2002.
10. Partridge, D. & Griffith, N. Multiple Classifier Systems: Software Engineered, Automatically Modular Leading to a Taxonomic Overview. *Pattern Analysis and Applications*, vol. 5(2), pp. 180-188, 2002.
11. Prechelt, L. Early Stopping - But When? In Orr, G. B. & Müller, K-R. (Ed), *Neural Networks: Tricks of the Trade, 1524*, pp. 55-69. Berlin, Heidelberg, New York: Springer-Verlag, 1996.
12. Sharkey, A.J.C. Multi-Net Systems. In Sharkey, A. J. C. (Ed), *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, pp. 1-30. London: Springer-Verlag, 1999.
13. Thrun, S.B., Bala, J., Bloedorn, E., Bratko, I., Cestnik, B., Cheng, J., De Jong, K., Dzeroski, S., Fahlman, S.E., Fisher, D., Hamann, R., Kaufman, K., Keller, S., Kononenko, I., Kreuziger, J., Michalski, R.S., Mitchell, T., Pachowicz, P., Reich, Y., Vafaie, H., van de Welde, W., Wenzel, W., Wnek, J. & Zhang, J. *The MONK's Problems: A Performance Comparison of Different Learning Algorithms*. Technical Report CMU-CS-91-197. Pittsburgh, PA.: Carnegie-Mellon University, Computer Science Department, 1991.
14. Wanas, N.M., Hodge, L. & Kamel, M.S. Adaptive Training Algorithm for an Ensemble of Networks. *Proceedings of the 2001 International Joint Conference on Neural Networks (IJCNN'01)*, vol. 4, pp. 2590-2595. Los Alamitos, CA.: IEEE Computer Society Press, 2001.
15. Wolberg, W.H. & Mangasarian, O.L. Multisurface Method of Pattern Separation for Medical Diagnosis Applied to Breast Cytology. *Proceedings of the National Academy of Sciences, USA*, vol. 87(23), pp. 9193-9196, 1990.
16. Xu, L., Krzyzak, A. & Suen, C.Y. Several Methods for Combining Multiple Classifiers and Their Applications in Handwritten Character Recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22(3), pp. 418-435, 1992.