ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 269-268

Generalisation Capabilities of a Distributed Neural Classifier

Arnaud Ribert, Abdel Ennaji, Yves Lecourtier

P.S.I. Faculté des Sciences, Université de Rouen
76 821 Mont Saint Aignan Cédex, France
Arnaud.Ribert@univ-rouen.fr

This article describes a new approach to the automated construction of a distributed neural classifier. The methodology is based upon supervised hierarchical clustering which enables one to determine reliable regions in the representation space. The proposed methodology proceeds by associating each of these regions with a Multi-Layer Perceptron (MLP). Each MLP has to recognise elements inside its region, while rejecting all others. Experimental results for a real problem (handwritten digit recognition) reveal an interesting generalisation behaviour of the distributed classifier in comparison to the k-nearest neighbour algorithm as well as a single MLP.

1. Introduction

Neural networks, and more particularly Multi-Layer Perceptrons (MLP) [1] have received a great deal of attention in recent years in the field of pattern recognition. Reasons of this success essentially come from their universal approximation property [2] and, above all, their good generalisation capabilities. However, obtaining good generalisation behaviour is not a trivial task when dealing with complex problems since a suitable neural network architecture has to be determined. Since there is no reliable and generic network building rule currently available [3-6], finding an efficient architecture often requires a large amount of trial and error that must be carried out by a specialist.

The approach proposed in this paper redefines the learning task of a neural network so that a simple network building rule can lead to good generalisation capabilities, while avoiding human assistance. This redefinition follows a "divide and conquer" strategy. The objective is to split a classification problem into several simpler ones. The method investigated in the next section achieves this objective while ensuring a coherency with the structure of the data in the representation space. Section 3 shows experimental results for a handwritten digit recognition problem.

2. Distributing the classification problem

Distributing a classification problem presents two main points of interest. The first one is to give the opportunity to simplify the design and the training of a neural

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 269-268

network by dividing up a given task into several simpler ones. The second advantage (which will not be developed in this paper) is to engineer a modular classifier. This feature implies an easy-to-update system : it will be possible to keep a part of the classifier after an adding of data in the training database.

The simplest problem to be given to a neural network is probably a linearly separable one. Unfortunately, few real problems present this feature. A second class of simple classification problems - although more complex than the previous one - may be encountered for a two class problem such that at least one of them is constituted by a unique pure cluster (i.e. containing elements of the same class only). The proposed "divide and conquer" strategy aims to identify this kind of cluster, called an "islet", in order to provide as many tasks as islets. If N islets are detected, the classifier will thus be constituted from N neural networks, each of them being quite simple to configure and being expected to present good decision boundaries.

To achieve this, the first stage consists of capturing the structure of the data in the representation space. That is to say one determines the number and the constitution of the clusters in it. The problem of distributing a classification task is thus converted into a clustering one. The most commonly used techniques are certainly partitional ones (like k-means [7]) and Self-Organising Maps [8]. The main problem with these methods is that in practice, the number of clusters is required in advance to obtain a good representation of the data. Moreover, in the case of partitional methods, the quadratic criterion of cluster compactness leads to hyper-spherical groups, which does not necessarily match the reality. Conversely, hierarchical clustering methods represent the data without any assumption on their distribution nor the number of clusters in it. In this study, this will be achieved using the supervised information. A hierarchy is built as follows [9] (where a point is considered as a group) :

```
Merge into a single group the two closest points according to the euclidian distance;
While ( There are more than one group ) Do
        Compute the distance between the new group and every existing one;
        Merge into a single group the two closest groups;
EndWhile
```

Figure 1.b shows an example of hierarchy obtained at the end of this process. It can be noticed that the height of a node is proportional to the distance between the groups it links. As one can see in the previous algorithm, the distance between a newly formed group and an existing one has to be defined. Commonly used distances are : the minimum or maximum euclidian distance (called single and complete link) or the average distance between the groups. This choice has a great influence on the representation capabilities of the hierarchy. In order to optimise the building process, and to work with a well-suited metric, the Lance-Williams' formula has been used [10] so that the resulting metric is close to the single link, but avoids the associated chaining effect (data tend to be merged into a single cluster). It is then possible to obtain big clusters whose shape is not necessarily hyper-spheric.

The second step of the distribution process consists of labelling the vectors according to their class (in the supervised meaning). In this way, it is possible to compare the supervised and unsupervised information provided respectively by a vector label and the hierarchy. Afterwards, an analysis of the composition of the sub-

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 269-268

trees reveals the presence of islets (i.e. sets comprising at least P elements from the same class, P being user-defined). Figure 1.c shows the resulting distribution after applying such a technique. Each islet is then learnt by an MLP which has to solve a two class problem : recognise its associated islet while rejecting every other element. Since the elements of an islet are close to one another, it can be expected that the problem is simple enough to allow a basic MLP building principle to be efficient.

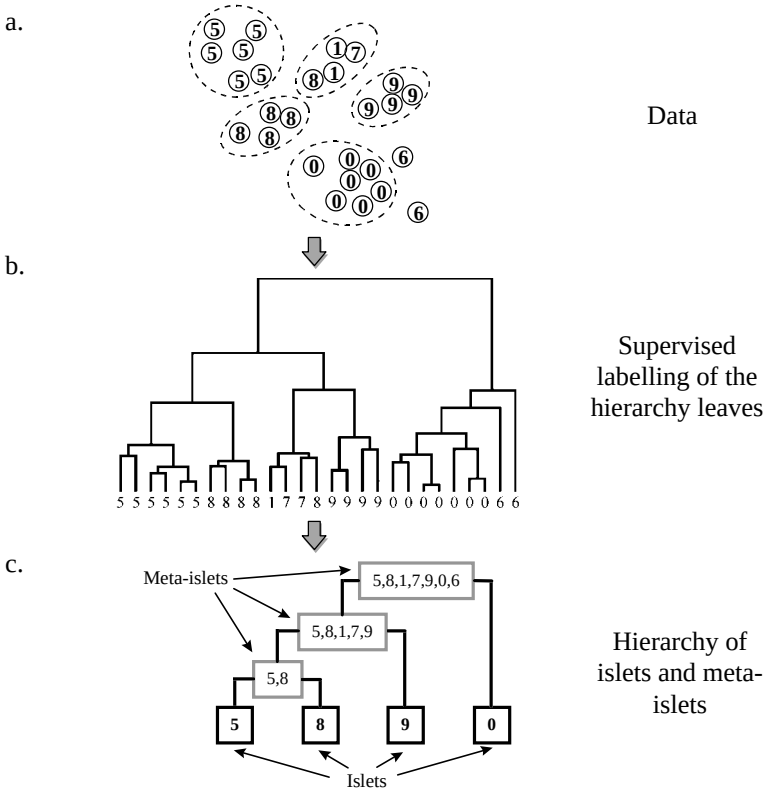


Figure 1 : An example of determination of islets and meta-islets

As might be expected, all elements will not be associated with an islet. Indeed, a certain amount of elements are located near the boundaries, so they will not appear in a pure sub-tree. Moreover, depending on the minimum size of an islet, large fluctuations on their number may arise, so that the percentage of elements associated with an islet may also vary in a large scale. Consequently, corresponding neural networks will not have learnt the whole database but only the most reliable part. A supplementary classifier has thus to be used in order to obtain high performance. Non-parametric methods seem to be best-suited, since they do not require any real learning stage and their computing cost can be reduced by the cooperation process. In practice, a K-nearest neighbours classifier has been used according to the following rule : simulate each neural network for the unknown pattern; if only one of them recognises the element, then take the decision of its class, else take the decision of the K-NN.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 269-268

3. Experimental results : a handwritten digit recognition problem

Tests have been performed for a handwritten digit recognition problem over the NIST database. The feature vector considered is constituted by the 85 (1+4+16+64) grey levels of a 4 level resolution pyramid [11] (see figure below).

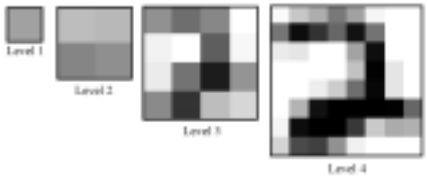


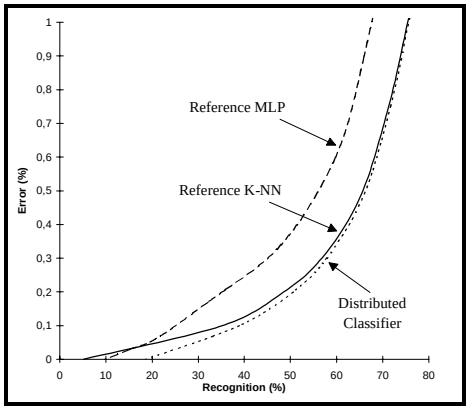
Figure 2 : Representation of a 2 using a 4 level resolution pyramid

In order to test the generalisation capabilities of the distributed neural classifier, two experiments have been carried out. The first experiment involves a small training database (660 instances of digits to be learnt) and a larger test set (21,000 instances). Training and test databases of the second experiment consisted of respectively 20,000 and 61,000 elements. For both configurations, three classifiers have been compared : a single MLP, a K-NN classifier and a distributed neural classifier. The neural networks involved in this last were trained using the classical back-propagation algorithm while their structure was found applying a simple rule : several architectures were considered (comprising 2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 50-20 or 100 hidden units) ; when the speed of convergence of a given architecture was too low, the next configuration was tested. The first architecture to achieve success in learning all the elements was retained. Single MLPs have not been trained like this, but with a trial-error procedure, because their generalisation performance was not satisfactory (however, the found architectures are not expected to be the best achievable). The curves in figures 3 and 4 represent the average error rate (i.e. Nb of bad decisions / Total Nb of presented digits) according to the recognition (i.e. Nb of recognised digits / Total Nb of presented digits), over 5 different training and test databases (cross-validation procedure). Rejection rate is thus given by 100-(error + recognition).

3.1 Training on 660 instances, test on 21,000

The following curves have been obtained by two different ways, depending on the considered classifier. The K-NN curve is obtained in decreasing k, while requiring that the k nearest neighbours are of the same class. The neural network curves are obtained in increasing the minimum value of the maximum output of the network : high thresholds will engineer low error rates. In this configuration, 7 islets of more than 15 elements have been detected. The distributed classifier consisted of 5 hidden unit networks. Since the training base is very small, only 44% of its elements were assigned to an islet. In spite of this poor rate, the distributed classifier noticeably improves the recognition rate in comparison to both K-NN and the single MLP. The benefits of using a distributed classifier also increase as the error rate decreases, although only 17% of the decisions are taken by the networks for a 0% error rate.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 269-268

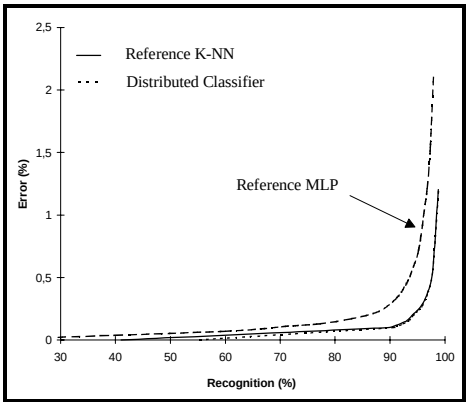


Error	Ref. K-NN	Ref. MLP *30 h.u.*	Distrib. classifier
Max.	93.0	90.7	93.0
3.0%	86.8	81.4	87.1
2.0%	83.2	76.6	83.4
1.0%	75.2	66.8	75.5
0.5%	65.4	55.3	66.1
0.4%	61,6	50.7	62.4
0.3%	56.8	44.2	57.9
0.2%	48.6	34.9	50.6
0.1%	35.2	24.8	38.9
0.0%	5.1	9.8	18.7

Figure 3 : Performances of 3 classifiers (660 digits on training set; 21,000 on test set)

3.2. Training on 20,000 instances, test on 61,000

In this test, an average of 120 islets of more than 15 elements were detected, and 76% of the training set was assigned to an islet. Most of the neural nets (88%) presented a single layer of 10 hidden units, while two had 2 hidden layers (of 50-20 units). It can thus be said that, as expected, learning an islet is a rather simple problem.



Error	Ref. K-NN	Ref. MLP *50-20 h.u.*	Distrib. classifier
Max.	98.8	97.9	98.8
0.5%	97.5	93.5	97.6
0.4%	96.9	92.3	97.0
0.3%	96.0	90.1	96.1
0.2%	94.1	85.5	94.3
0.1%	89.9	69.0	90.6
0.0%	41.0	15.1	54.8

Figure 4 : Performances of 3 classifiers (20,000 digits on training set; 61,000 on test set)

The previous behaviour also appears in this test, although the differences between the K-NN and the distributed classifier occur for smaller values of the error rate. Thus, for a 0% error rate, the recognition rate of the distributed classifier is 14% higher than the K-NN one, whereas in this configuration only 41% of the decisions are taken by the networks. This difference shows a fundamental dissimilarity between built boundaries (which are implicit in the case of the K-NN algorithm). Training neural networks on islets enables them to recognise an element from a class C whereas its 50 or 55 nearest neighbours are from another class. Boundaries engineered by the

learning of islets (even following a simple network building rule) are therefore particularly efficient. It can be noticed that single neural networks rarely implement such boundaries since no explicit learning rules exist to find them.

4. Conclusion

This article deals with the problem of finding a well-suited MLP architecture for a given problem, without any human expertise. The proposed solution consists of distributing the classification problem in several simpler sub-problems (called islets) which are determined by a supervised hierarchical clustering procedure. Experimental results for a real classification task show that training a neural network to solve such a sub-problem leads it to define efficient decision boundaries, specially when low error rates are required. Indeed, a simple network building strategy permitted the recognition of difficult patterns for the K-NN classifier, and produced better results than a purpose-designed MLP. The reliability of neural network decisions for low error rates is thus significantly improved. Further experiments should lead to a better characterisation of these boundaries to provide explicit rules for high-performance network building.

References

[1] Rumelhart D., McClelland J. : Parallel distributed processing, explorations in the microstructure of cognition. Vol 1. Cambridge, MA : MIT Press, 1986.
[2] Hornik K. : Approximation Capabilities of Multilayer Feedforward Networks, Neural Networks, Vol 4, p. 251-257, 1991.
[3] Alpaydin E. : GAL : Networks that grow when they learn and shrink when they forget. IJPRAI, Vol. 8, (1), pp. 391-414, 1994.
[4] Ash T. : Dynamic node creation in backpropagation networks. Connection Science, Vol. 1, (4), pp. 365-375, 1989.
[5] Autere A. : Comparison of genetic and other unconstrained optimization methods. In D.W. Pearson, N.C. Steele, R.F. Albrecht (Eds.), Artificial neural nets and genetic algorithms. Springer-Verlag, (pp. 348-351), 1995.
[6] Belew R.K. et al. : Evolving network : using the genetic algorithm with connectionnist learning. In Langton C.G. (Eds.), *Artificial life II, SFI studies in the sciences of complexity*, Vol X, Addison-Wesley, (pp. 511-547), 1991.
[7] MacQueen J.B. : Some methods for classification and analysis of multivariate observations. Proceedings of the 5th Berkeley Symp. on Math. Statistics and Probability, Vol. 1, pp. 281-297, 1967.
[8] Kohonen T., Kangas J.A., Laaksonen J.T. : Variants of self-organising maps, IEEE. Transactions on Neural Networks, Iss. 1, pp. 93-99, 1990.
[9] Jain A.K., Dubes R.C. : Algorithms for clustering data., Prentice Hall, 1988.
[10] Lance G.N., Williams W.T. : A general theory of classificatory sorting strategies 1. Hierarchical systems., *Computer Journal*, Vol. 9, pp. 373-380, 1967.
[11] Ballard D.H., Brown C.M. : Computer Vision. Prentice Hall, 1982.