An Investigation into the performance and representations of a Stochastic, Evolutionary Neural Tree

K. Butchart, N. Davey, R.G. Adams

K. Butchart@herts.ac.uk, N. Davey@herts.ac.uk, R.G. Adams@herts.ac.uk
School of Information Sciences
University of Hertfordshire
Hatfield, Herts., UK. AL10 9AB
tel (01707) 284321

Abstract: The Stochastic Competitive Evolutionary Neural Tree (SCENT) is a new unsupervised neural net that dynamically evolves a representational structure in response to its training data. Uniquely SCENT requires no initial parameter setting as it autonomously creates appropriate parameterisation at runtime. Pruning and convergence are stochastically controlled using locally calculated heuristics. A thorough investigation into the performance of SCENT is presented. The network is compared to other dynamic tree based models and to a high quality flat clusterer over a variety of data sets and runs.

§1 Introduction

In this paper we report a thorough comparative analysis of the performance of a new unsupervised neural net architecture, SCENT, first introduced in [Butchart et al., 1996b]. This architecture contains two unique features: firstly in its combination of evolutionary growth with stochastic pruning and secondly in its dynamic, autonomous parameterisation; the user supplies no information to the net aside from the training data. Here we evaluate the network against the best networks with equivalent functionality using a collection of data sets chosen to provide a variety of clustering scenarios and in the full paper present a detailed description of the nature of the hierarchical representations induced.

§2 Dynamic Neural Trees and Stochastic Clustering

Unsupervised clustering using neural networks is a well established technique, with the simple competitive net (SCN) being the elementary model and Kohonen's self organising maps (SOMs) the most popular manifestation [Hertz et al., 1991]. In a SOM the classifying units are arranged with some spacial topology, usually a grid. Recent research has investigated the possibility that the units could be arranged in a tree structure; such an arrangement has two potential benefits: searching a tree is fast and any hierarchical information in the data may be explicitly represented in the tree. Most clustering algorithms, neural net or otherwise, expect the number of classes used to be predefined; this is an obvious problem if no a-priori knowledge about the data is available.

dynamically evolve the appropriate structure in response to the data . Such an approach can naturally lead to tree structures - a unit may produce child nodes if it feels it is not classifying its local data with sufficient granularity.

For the purposes of the results reported here we have compared our model with two other dynamic neural trees, the dynamic competitive learning technique of Racz and Klotz [Racz and Klotz, 1991], and the Neural Tree of Li [Li et al., 1992]. Apart from technical differences between the models SCENT has one fundamental difference: it requires no initial parameter setting. This is an important quality, as moving the need for the number of clusters to be specified, to one of specifying some other less transparent parameters is of doubtful benefit.

Most learning rules in neural networks, when viewed holistically, perform gradient descent on an error function and unsupervised clusterers are no exception. This approach has the well documented problem of local minima in the error function, often seen as "dead" units in a simple competitive neural net. Several researchers have proposed the addition of stochasticity, often in conjunction with a simulated annealing technique, as a means of overcoming this difficulty for competitive networks. One of the best of these clusterers is the Neural Gas model of Martinez [Maritnez et al., 1993] which we use as another source of comparison.

The idea of stochasticity at an associated temperature is also used in the SCENT model as described in the next section.

§3 The SCENT Model

3.1 Classification and Learning

In the SCN model a search through all nodes is undertaken at each input presentation for the node that is closest, which is designated the winner. The winner then moves towards the input vector. In a tree structured network, the search is recursive and thereby restricted to the winning branch, by the following algorithm:

Find the *best-child* of *current*, that is closest to the *input*
Move *best-child* towards *input*
Set *current* to *best-child* and recurse

Once learning is completed classification is provided by leaf nodes, whilst the rest of the tree provides a hierarchical classification. The search for a winner either in the learning or test phase is therefore O(db), where d is the average depth and b is the average branching factor, which contrasts with an $O(b^{d-1})$ search of all leaf nodes.

3.2 Growth and Pruning

As already stated growth decisions in SCENT are devolved to units in the tree. A unit is allowed two modes of growth, it may produce children or it may produce a sibling. A node may only produce children once, and when it does so it creates a pair, offset from the parent by a small amount of noise. Further children are generated by an existing child producing a sibling, once again a noisy copy. The decision that a node makes to grow is determined in the SCENT network by *relative activity.* Broadly speaking this is the ratio of the number of times a node has won to the number of times its parent has won. In order for the performance of the network to be relatively invariant with respect to the order of presentation of the inputs, activity is calculated over a sliding window with a linearly decreasing weighting, as originally used in Li's Neural Tree.

3.2.1 Outwards or Downwards

Having decided to grow, a leaf node must decide whether it should spawn children or a sibling and to do this it makes use of its *tolerance* value. The tolerance of a node is the radius of its classificatory hypersphere. In order for finer grained classification to be performed in lower levels of the tree, tolerance is reduced from parent to child. The reduction is scaled by the success of the parent in placing those vectors it classifies within tolerance. If most activity is within tolerance the children are given tolerance values of significantly smaller size - reduction is up to 40%.

If a leaf node finds that the majority of the vectors it classifies are within tolerance then it is classifying a spatially compact cluster and should therefore produce children to subclassify this group. On the other hand, if most vectors are not within tolerance, there may be large scale structure that should be represented by another node at the same level - so a sibling is spawned.

3.2.2 Pruning

A leaf node does not have a guarantee of continued existence and it may die at birth or later in life. Short term pruning, or *growth rejection*, takes place if a new node does not reduce the error of its parent sufficiently; this decision is taken at the epoch end immediately following its creation. A leaf node that has established itself in the tree may still be removed if its long term performance is inadequate. The performance is measured by comparing the activity of the node against a threshold that decreases in lower levels of the tree; node removal is only performed at epoch boundaries.

It is in the pruning process that stochasticity is used. In early growth it is useful for the network to create much tentative new growth, and also for longer term pruning to be more common. To this end *growth rejection*, is initially made less likely - it becomes a probabilistic process, inversely proportional to a temperature value that is initially high and decreases exponentially over time. Similarly long term pruning is made more likely early in a node's life - it is stochastic, but directly proportional to temperature. The addition of this mechanism leads to a more reliable performance by the model when compared to its non-stochastic predecessor, CENT [Butchart et al., 1995]. See [Butchart et al.,19 96a] for comparative results.

3.2.3 Parameterisation

As described earlier it is important for data exploration that a clusterer does not require the user to specify a set of opaque parameters that will define the final classification. In previous dynamic tree based clusterers two key values, tolerance and threshold had to be initialised. In the SCENT model the threshold value is subsumed in the node creation process and the tolerance value is calculated at run time. The root node has a tolerance such that 2/3 of the data lies within tolerance; this is simply accomplished by an initial pass through the data set, which will also position the root node roughly at the mean of the data. Subsequent tolerance values for descendants of the root are calculated recursively from this initial value as described above.

§4 Tests

The four neural net models identified earlier: NGas, Neural Tree (Li), Dynamic Learning (RK) and SCENT were used to cluster seven data sets summarised in the following table:

Set 1	2D single source Gaussian
Set 2	20D single source Gaussian
Set 3	2D Uniform within a square
Set 4	16 even clusters in 4 groups, 2D
Set 5	10 clusters with varying density and size
Set 6	Varied Clusters with hierarchical stucture
Set 7	Anderson's IRIS data [Everit, 1993]

Each model was run over each data set for 15 complete runs. For all the networks except SCENT, experimentation was initially performed to find a satisfactory set of parameters. In particular the number of nodes in the NGas model was set to be roughly that number of the leaf nodes that SCENT produced. The results produced are averages over these 15 runs.

It is very difficult to judge the performance of a hierarchical cluster, but relatively straightforward to measure the quality of a flat classification. The aim of a clusterer is to produce low Sum Squared Error (SSE) but in a dynamic network this is complicated by the ability of the net to use an unbounded number of nodes. A more reasonable network comparison can therefore be obtained by using the product of SSE and number of nodes.

The results are summarised in the following table, in which the column marked "*" is the product of SSE and Nodes. Data set 5 is shown in Figure 1.

	NGas			Li			RK			SCENT		
Set	SSE	Nodes	*	SSE	Nodes	*	SSE	Nodes	*	SSE	Nodes	*
1	15	16	**240**	22	26	**572**	24	57	**1368**	20	28	**560**
2	679	16	**10864**	766	110	**84260**	346	685	**237010**	678	27	**18306**
3	11	25	**275**	11	43	**473**	48	20	**960**	21	29	**609**
4	125	32	**4000**	194	55	**10670**	602	42	**25284**	248	25	**6200**
5	272	16	**4352**	298	49	**14602**	455	45	**20475**	330	27	**8910**
6	132	16	**2112**	125	40	**5000**	79	56	**4424**	129	28	**3612**
7	1	16	**16**	6	8	**48**	3	32	**96**	2	36	**72**
Average	176	19.6	**3123**	203	47.3	**16518**	222	134	**41374**	204	28.6	**5467**

The presentation of the results is to some extent unfair to the tree based networks as many nodes are not used as classifiers, and in this light the performance of SCENT is exceptionally strong. Its SSE by #nodes is roughly double that of NGas, which implies that the leaf nodes are almost certainly out-performing NGas. The performance gap to the other two dynamic trees is substantial.

When attempting to judge the quality of the hierarchies produced by the tree based networks the criteria to look at are more subjective. However for the data which is designed to be hierarchical we would like the induced structure to reflect that of the data, so that, for example, the top level of the tree should contain representatives for each of the major groups in the data and subsequent layers should reflect subgroupings. Some behaviour is not desirable, specifically a tendency to produce long thin branches which do not add to the semantic content of the tree. It was found that both Li and RK had a tendency to do this which could only be avoided by very careful selection of parameters, although even this was not always possible. The very high average node count for RK is explained by this phenomena. A typical example of the classication tree produced by SCENT for data set 5 is shown in Figure 2, together with the position of the classifying nodes. It can be seen that the network has produced a near optimal classification and a reasonable tree structure. In the full paper a thorough analysis of the hierarchical representations of SCENT will be presented.

One further point is worth noting: the results from SCENT were reasonably repeatable - tree stuctures produced varied across runs but were generally consistent.

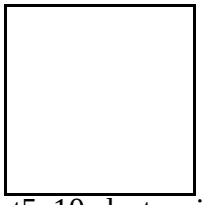
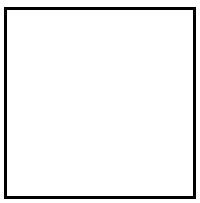


Figure 1: Data set5, 10 clusters in 4 large groups



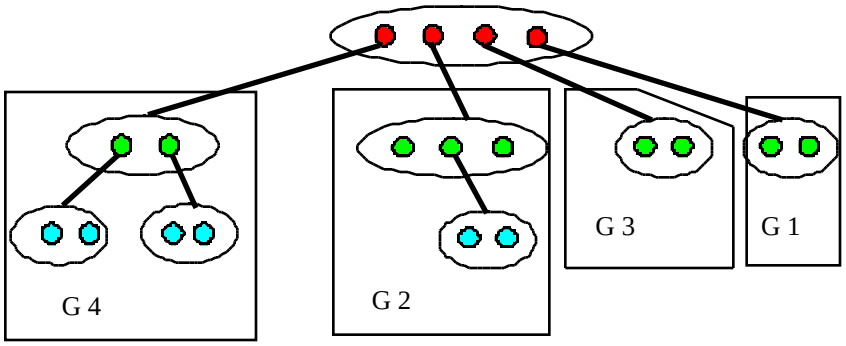


Figure 2: the position of nodes in a typical run of SCENT, and below the organisation of the nodes in the resulting tree.

§5 Conclusions

Using neural nets to perform data exploration is difficult; most models require the preimposition of a maximum number of clusters, and will normally classify the data to utilise all classificatory units. Some recent architectures have attempted to dynamically create tree structures to overcome the need for prescribing cluster number and to give a hierarchical view of the data. Their use though has been problematic, with a tendency to display great sensitivity to initial parameter values. SCENT attempts to overcome this problem by being completely autonomous. In the tests reported here the model produced excellent results when viewed as a straightforward clusterer, giving comparable performance to the NGas network. It also produced useful, compact and repeatable tree structures to represent any hierarchical information in the data.

References

Butchart, K., Davey, N., Adams, R., (1995) "Hierarchical Classification with a Competitive Evolutionary Neural Tree", submitted to Neural Networks.

Butchart, K., Davey, N., Adams, R. (1996a)"A Comparative Study of two Self Organising and Stuctually Adaptive Dynamic NeuralTree Networks", in *Neural Networks and their Applications* , J.G.Taylor Editor, John Wiley,.

Butchart, K., Davey, N., Adams, R. (1996b)"Hierarchical Classification with a Stochastic Competitive Evolutionary Neural Tree", proceedings of ICNN96, Vol. 2, pp 1372 - 1377.

Everit B.S. (1993) "*Cluster Analysis*", Edward Arnold , London,.

Hertz, J.,Krogh A., Palmer, R., (1991)*"Introduction to the theory of Neural Computation"*, Addison Wesley, .

Li, T., Tan , Y., Suen, S. and Fang, L,. (1992) "A structurally adaptive neural tree for recognition of a large character set." In: *Proc. 11th IAPR International Joint Conference on Pattern Recognition*, vol II,pp187-190,

Martinetz , T. , Berkovich , S. and Schulten , K. (1993) "Neural-Gas Network for Vector Quantisation and its Application to Time-Series Prediction." *IEEE transactions on Neural Networks.*, vol. 44 (4) .

Racz,J. and Klotz, T. (1991) “Knowledge Representation by dynamic competitive learning techniques.” *SPIE Applications of Artificial Neural Networks II* , vol. 1469, pp 778-783,