DIVISION OF COMPUTER SCIENCE

A Comparative Study of three Neural Networks that use Soft Competition

Kate Butchart

Technical Report No.211

December 1994

A Comparative Study of three Neural Networks that use Soft Competition

Kate Butchart
comrkb@herts.ac.uk ;
School of Information Sciences
University of Hertfordshire
Hatfield, Herts., UK. AL10 9AB

Abstract - This report provides a comparative study of three proposed self organising neural network models that use forms of soft competition. The use of soft competition helps the neural networks to avoid poor local minima and so provide a better interpretation of the data they are representing. The networks are also thought to be generally insensitive to initialisation conditions. The networks studied are the Deterministic Soft Competition Network (DSCN) of Yair et al. , the Neural Gas Network of Martinetz et al and the Generalised Learning Vector Quantisation(GLVQ) of Pal et al. The performance of the networks is compared to that of standard competitive networks and a Self Organising Map when run over a variety of data sets. The three proposed neural network models appear to produce enhanced results, particularly the Neural Gas network, but in the case of the Neural Gas network and the DSCN this is at the cost of greater computational complexity.

1.0 Introduction

A standard Competitive Learning Neural Network (CLNN) [1] uses gradient descent on a cost function to settle into an equilibrium state. By using gradient descent such networks are prone to finding local minima, the quality of which is highly dependent upon the initial random state of the network. Soft competition, where there is more than one winner, has been used to try to prevent some the problems of finding poor local minima. Networks that also use a temperature coefficient to reduce the "softness" of the competition over time, employing a simulated annealing type approach, may be the most successful in avoiding local minima. This report analyses and evaluates three CLNNs [2,3,4] which use forms of soft competition. The three CLNNs all claim to produce solutions independent of the initial state of the network, and two claim to find near global solutions.

Section 2 of the report looks at clustering, standard CLNNs , vector quantisation and the theory of simulated annealing. In section 3 the three new network models are described. The three network models were run over different types of data and the results of these tests are reported and discussed in section 4.

2.0 Clustering and Competitive Learning Neural Networks

Most clustering algorithms reduce a cost function by iteratively altering the prototype weight vectors in response to the inputs. The cost function that is being reduced determines the form of the update rule used. The most common cost function is based on the Mean Squared Error (MSE) measure that is given as

$$E = \sum_{x} \left\| x - y_{i*} \right\|^2 \qquad (1)$$

where x is input vector

y_{i*} is the winning node

and $$\left\| x - y_{i*} \right\|^2 \le \left\| x - y_i \right\|^2 \qquad \forall i \qquad (2)$$

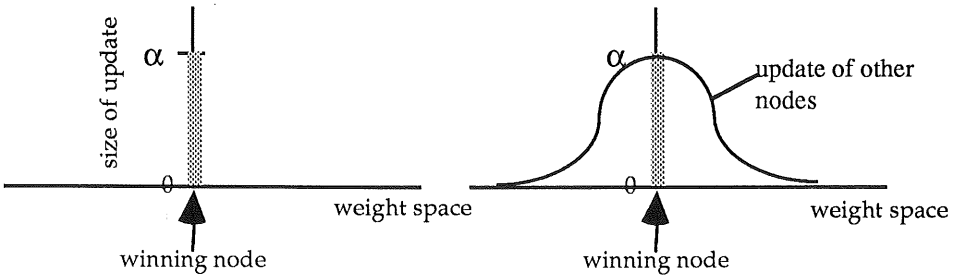
A standard CLNN performs clustering on the input data by reducing the MSE given in (1). The MSE is reduced by updating the winning node for each input using the rule

$$y_i(n) = y_i(n-1) + \alpha[x(n) - y_i(n-1)] \qquad (3)$$

where α is the learning rate and is global to the whole network.

This update rule is directly derived from the MSE and so the network performs gradient descent on the MSE function. If an alternative cost function is to be reduced by the network then a different update rule that is derived from that cost function will be used.

In standard CLNNs just the node that wins the competition to classify the input updates its weight vector, as shown in Figure 1a all other nodes have an update rate of 0 , and they remain unchanged. Some types of CLNN may update more than just the winning node with the other nodes moving to a lesser extent, this is shown in figure 1b.



a) b)

Figure 1 a)the winning node has an update rate of α. All other nodes whatever their position in the weight space in relation to the input are not updated. In part b) the winning node again has an update rate of α , other nodes may also be updated, the size of their update being dependent upon their position relative to the winning node.

The update rule in (3) uses gradient descent to minimise the MSE and as such may well find local minima. The quality of the minima found is very dependent upon the initial random starting weight values of the competitive units.

2 . 1 Vector Quantisation

Vector quantisation is a technique that can be used to map discrete vector sources into a sequence of digital data for transmission or storage. A vector quantiser maps an input vector to one of a finite set of predetermined codevectors. The goal in designing a vector quantiser is to construct a set of codevectors for which the expected distortion, caused by approximating an input vector by a codevector, is minimised. The most common distortion measure used is the mean squared error (MSE). To produce a low MSE two necessary conditions that need to be met are the centroid and nearest neighbour conditions [5]. The centroid condition requires that the prototype vector of a class be at the centroid of the class of inputs it classifies. The nearest neighbour condition states that any input vector must be classified by the nearest prototype vector.

CLNNs can be used to build vector quantisers. A CLNN performs gradient descent on the MSE function and produces a vector quantiser which meets both the centroid and nearest neighbour conditions. The quality of the set of codevectors found is dependent upon the quality of the minima the network finds. But by using purely gradient descent techniques (albeit with on line rather than batch training) the network can easily fall into poor local minima especially if initial starting weight values are poor. A more reliable method of reducing the MSE needs to be found before CLNNs can be used for vector quantiser design.

2.2 Simulated Annealing

Annealing is a process whereby a collections of atoms is reduced to its lowest energy state. Simulated annealing was first introduced by Metropolis et al.[6] who proposed a simple algorithm that can be used to simulate a collection of atoms in equilibrium at a given temperature. In each step of this algorithm an atom is given a small random displacement and the resulting change ΔE, in the energy of the system is computed. If $\Delta E <= 0$ the displacement is accepted and the state is changed. If $\Delta E > 0$ then the state is accepted with probability

$$P(\Delta E) = e^{-\frac{\Delta E}{K_b T}} \tag{4}$$

where K_b is the Boltzman constant, and T is the temperature

This choice of $P(\Delta E)$ means that the system evolves to a Boltzman Gibbs distribution, where the probability of the system being in a state s is

$$P(s) = \frac{1}{Z} e^{-\beta E(s)} \tag{5}$$

where $\beta = 1/T$ and Z is a normalisation factor defined so that the sum of P(s) over all possible states is unity, and so conforms to a probability density function.

By gradually reducing T over time simulated annealing is performed [7]. For high values of T there is an element of random movement in the system as states that increase E have a fairly high probability of being accepted. This random movement allows the system to escape local minima. At low temperatures a state that increases E is less likely to be accepted. As T is reduced so is the probability of accepting states that increase E and the system gradually evolves towards gradient descent. If the temperature schedule is correct for the problem the global minima should be reached[7].

3 . 0 The Three Networks

The neural networks that are analysed and evaluated in this report are :

* Yair et al's Deterministic Soft Competition Network (DSCN) [4],
* The Neural Gas network of Martinetz et al[2]
* Generalised Learning Vector Quantisation (GLVQ) of Pal et al[3].

3 . 1 The Deterministic Soft Competition Network

This network provides a deterministic form of Soft Competition derived from the principles of simulated annealing .

DSCN does not produce winners and losers, but updates all of the nodes for each input. The nodes are updated at differing rates, the rate of learning for each node being dependent upon: its distance from the input, the current temperature of the network and the current size of its individual counter.

3 . 1 . 1 DSCN Learning Rates

The update rule for each vector is

$$y_i = y_i + a_i P_n(i)[x - y_i] \qquad (6)$$

Hence the elements that effect the rate of learning (the size of the update step) are the values of $P_n(i)$ and a_i

$P_n(i)$ is the probability of output node y_i winning according to a Gibbs distribution

$$P_n(i) = \frac{e^{-\beta\|x-y_i\|^2}}{Z} \qquad (7)$$

where $\beta = 1/T$ (8)

and $Z = \sum_i e^{-\beta\|x-y_i\|^2}$ is the normalisation factor (9)

This means that at high temperatures the $P_n(i)$ value is relatively even between nodes as it is not overly sensitive to the distance between a node and the input vector. As the temperature decreases the $P_n(i)$ values become very sensitive to the distance between a node and the input vector, nodes close to the input gain relatively high values and those further away very low values.

a_i is inversely proportional to the node's individual counter.

$$a_i(n) = \frac{1}{n_i(n)} \tag{10}$$

n_i is the counter for each unit that is updated for each input by the rule

$$n_i(n) = n_i(n-1) + P_n(i) \tag{11}$$

This means that the more often the unit has a high $P_n(i)$, the smaller the a_i value becomes. This adds a form of "conscience" mechanism to the network.

3.1.2 How the learning rates vary according to system temperature and time

Initially when the temperature is high $P_n(i)$ values are approximately uniform so the output nodes are not yet attracted to certain clusters and all migrate towards each input vector. As the temperature lowers those near to the input take a larger step size and those further away a smaller one, as shown in Figure 2. It should be noted that nodes which are close together will have a similar $P_n(i)$ value and are likely to have a similar a_i value. Hence nodes that become close together have very similar learning rates and so can become "stuck' together. Since at high temperatures the nodes all tend to migrate to the same point it is very important the network is not given too high an initial temperature or all the nodes may become merged.

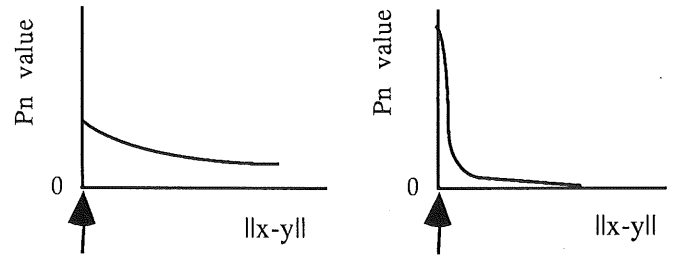


Figure 2 Network P_n values : the P_n values for nodes at a) high temperatures b) low temperatures.

The value of a_i for each unit can be seen as a secondary temperature that reduces over time. The size of the reduction being dependent upon $P_n(i)$, so units with little probability of being the winner maintain a larger learning rate.

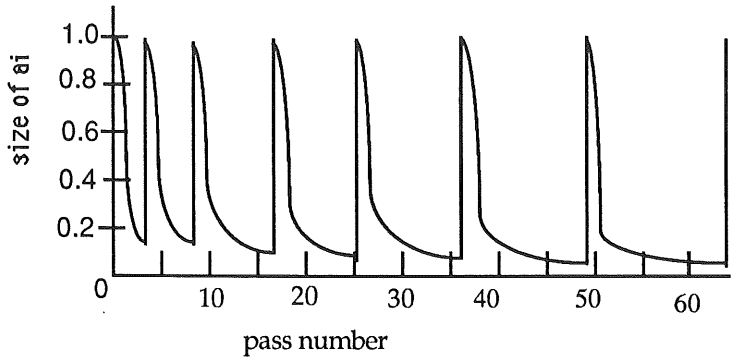


Figure 3 : Network Learning rates for a_i . Periodically all n_i are set to unity , so the value of a_i becomes 1. The size of a_i then decreases. The length of time between resetting the values increases over time, allowing the learning rate of the network to become more settled.

The value of T is reduced at the start of each pass through the input data set. The secondary temperature a_i is set to unity each time the number of passes through the data set equals a perfect square, as shown in Figure 3. The periodic increase in the secondary temperature makes the networks movements more random and so can help to jog the network out of local minima. As the network completes more passes through the data set the periods between re-initialisation of a_i become longer, allowing the network to settle on a solution.

3 . 2 Neural Gas Network

Martinetz et al [2] propose the Neural Gas network that uses soft competition and a temperature coefficient to effect the softness of the competition over time. The network minimises the cost function

$$E = \sum_{x} \sum_{i=1}^{N} e^{-\beta(ki(x,y))} . \|x - y_i\|^2 \qquad (12)$$

where

$\beta = 1/T$

$k_i(x,y)$ is a function which represents the ranking of each input vector x with each weight vector y_i . If y_i is closest to input x then k = 0, for the second closest k = 1 and so on.

At high temperatures many of the nodes in the network add to the cost (error) of the network. As the temperature is reduced only the nodes close to the input contribute significantly to the error, and at very low temperatures effectively just the winning node produces an error. Thus the cost function is reduced to the MSE function in (1) for very low temperature values.

The update rule, that is applied to all nodes, and which reduces the cost

function in (12) is

$$y_i(n) = y_i(n-1) + \alpha e^{-\beta(ki(x,y))}(x - y_i(n-1)) \qquad (13)$$

where α is in the range 0 to 1 and is global to the network.

Figure 4 shows how the size of the temperature effects the update rates for the network. At very low values for T the update is reduced to that of standard CLNNs shown in Figure 1.

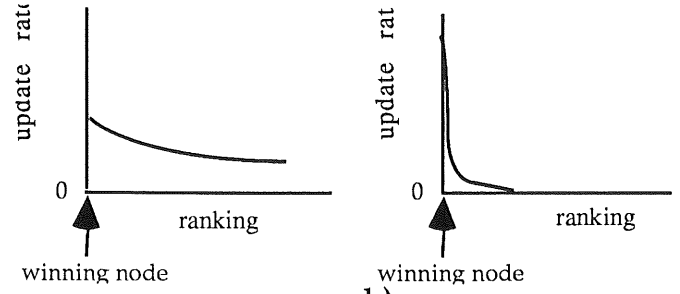


Figure 4 Network update rates : the update rate values for nodes at a) high temperatures b) low temperatures.

This method was compared by Martinetz et al[2] to k-means clustering, a Self Organising Map (SOM) [8] network and the simulated annealing clustering method of Rose et al. [9]. In their tests the Neural Gas network performed the best, converging to better solutions. Although it does not use the simulated annealing theory that is the basis of the DSCN it may well be that in practice this method is as effective.

2.3 GLVQ network

A Generalised LVQ (GLVQ) method is proposed by Pal et al [3]. It uses soft competition as all of the nodes are updated unless the input is classified perfectly. The cost function being minimised is

$$E = \sum_x \sum_i g_{ji} \cdot \|x - y_i\|^2 \qquad (14)$$

where

j is the winning unit

and $g_{ji} = 1$ if i=j otherwise $g_{ji} = \frac{1}{\sum_{k=1}^{N} \|x-yk\|^2}$ (15)

where N is the number of nodes

There is just one learning rate for all of the losers, that is partially dependent upon the classification error produced by the winning node.

The update rules for the network are

for the winner $$y_i = y_i + \alpha(x - y_i)\frac{D^2 - D + \|x - y_i\|^2}{D^2} \quad (16)$$

for the losers $$y_i = y_i + \alpha(x - y_i)\frac{\|x - y_i\|^2}{D^2} \quad (17)$$

where $$D = \sum_i \|x - y_i\|^2 \quad (18)$$

α is global to the network and reduces over time

When the match between winner and input is perfect then losing nodes are not updated. As the mismatch between the winning weight vector and input vector increases the impact on the losing nodes also increases.

The GLVQ performs gradient descent on a cost function other than the MSE . In so doing it may be able to find a solution that provides a lower MSE than can be found by performing gradient descent on the MSE function itself, particularly if the initial starting conditions are poor.

4 . 0 The Comparative tests.

The DSCN, Neural Gas network and the GLVQ network were implemented and run over a variety of data sets. A standard CLNN , a SOM, and CLNNs with conscience[10] and leaky learning [1] mechanisms were also run over some of the data sets to provide performance comparisons. The CLNN with a conscience mechanism is equivalent to the Frequency Sensitive Competitive Learning method (FSCL) [11,12].

The networks were run over a model problem where the global minima could be calculated, data from more than one Gaussian cluster and data from a single Gaussian cluster.

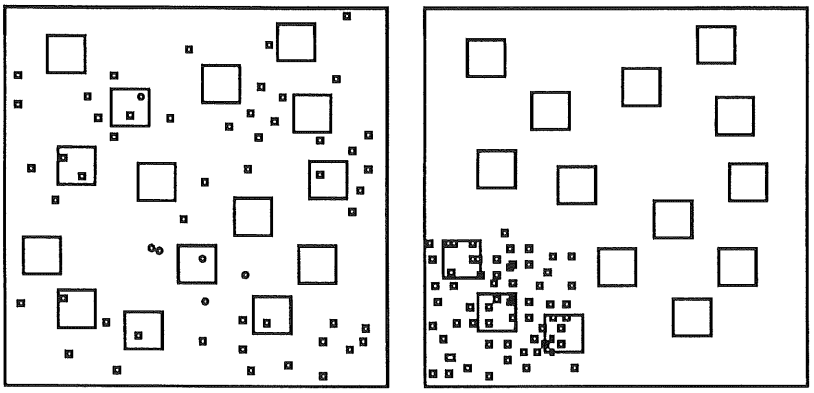
For the sake of comparison when an error measure was required the same MSE measure was used for each network type.

3 . 1 Squares data - a model problem

To test the performance of the networks in minimising the MSE, a data distribution was chosen for which the MSE global minimum can be

calculated for a large number of vectors. This data distribution was used by Martinetz at al. in their Neural Gas paper[2].

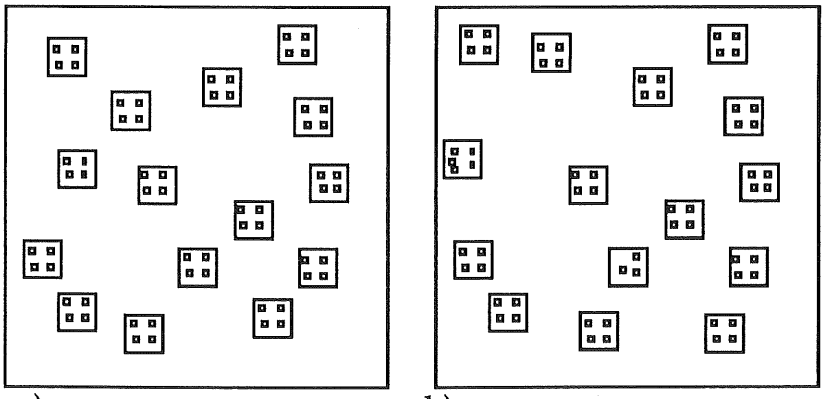
The clusters of data points are of a square shape and are separated from each other. The number of nodes was chosen to be four times the number of clusters, hence the final optimal configuration of the weight vectors is four vectors for each cluster arranged in the known optimal configuration for a single square. Figure 5 shows an example of the square positions (the positions were selected at random). The networks were run with initial starting positions for the weight vectors that were well spread over the input area, as shown in Figure 5a, and with starting positions that were not evenly spread over the input area, shown in Figure 5b.



a) b)

Figure 5 Typical weight vector starting positions for the Squares data. The large squares represent the 15 regions where the input vectors are generated with equal probability. The small squares represent the weight vectors of the nodes, two initial starting positions are shown, a) where the vectors are well spread out over the input area, and b) where the vectors are grouped in one corner of the potential input area.

4 . 1 . 1 Neural Gas network Performance



a) b)

Figure 6 Neural Gas representation . The neural gas network representation of the data distribution after 60000 adaptation steps. For the configuration of the square clusters in a) the network has found the optimal state. b) shows a more frequently occurring

situation where the network has come close to the optimum.

The Neural Gas network performed well over this data. For some configurations of the clusters it was able to find a global minimum, as shown in Figure 6 a), and for all configurations found a solution that was close to the optimum. On average the Neural Gas performance was 12 % worse than the optimum.

The solution found by the network did appear to be independent of the starting position of the network, with the difference in the quality of the solution being less than 0.1% for differing start positions.

4.1.2 DSCN performance

This network was originally designed as a vector quantiser for single source data and did not prove to be very suited to clustered data such as this. The problem with the network is finding the correct temperature schedule. If the temperature is too high the weight vectors of the nodes tend to merge together, if it is too low the network freezes before it has found a good solution. The authors of the network provide a heuristic for start temperature of the network over single source data but not for clustered data.

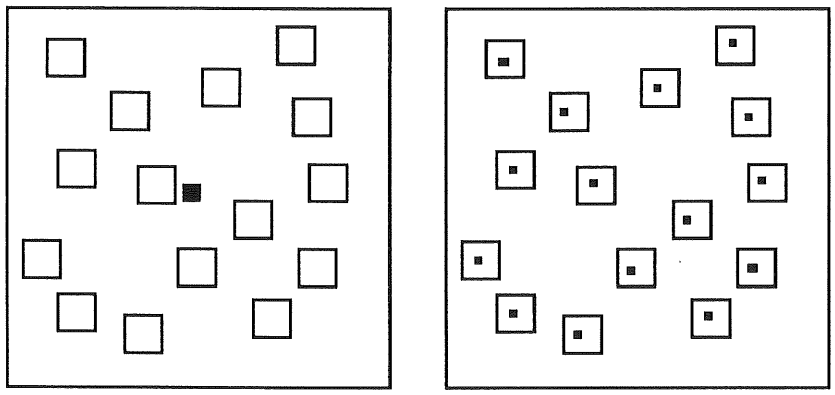


Figure 7 DSCN representation of Squares data
The DSCN representation of the "Squares" data. The left picture shows the result if the temperature is too high, all the vectors have converged to the same position. The right hand picture shows the best representation I was able to gain from the network over this data. Again some of the nodes have merged together but each square is represented by at least one vector, most squares have between 2 and 5 vectors merged at their centre.

Suitable temperature schedules for the network were investigated empirically. Figure 7 a) shows the result formed using too high a temperature, where all of the weight vectors have converged to the same point. Once the weight vectors become very close the network is not able to separate them due to its method for calculating the learning rate for the nodes. (Nodes that are close together will have the same learning rate).

The tendency of the nodes to stick together also occurs at lower temperatures. Figure 7 b) shows the best solution the network found over this data , and here again the nodes have become very close together but this time in groups , with each square having a group of nodes at its centre. Inevitably with nodes so close together the network is not able to reduce the distortion error as would be hoped. This solution represents an ideal solution given just 15 of the 60 nodes. The local clustering of the other 45 nodes added nothing to the functionality of the network. The average distortion values for the network were 370% greater than the optimum.

The network converged to similar states independent of the starting positions and so was not effected by changes in the starting positions.

4.1.3 GLVQ network performance

The network performed reasonably well over this data, but it was not able to bring all the nodes into the competition. On average one third of the nodes did not move into a position to be able to classify any of the data points. This prevented the network from getting close to the optimum value. The value of E for this network was on average 180% greater than the optimum.

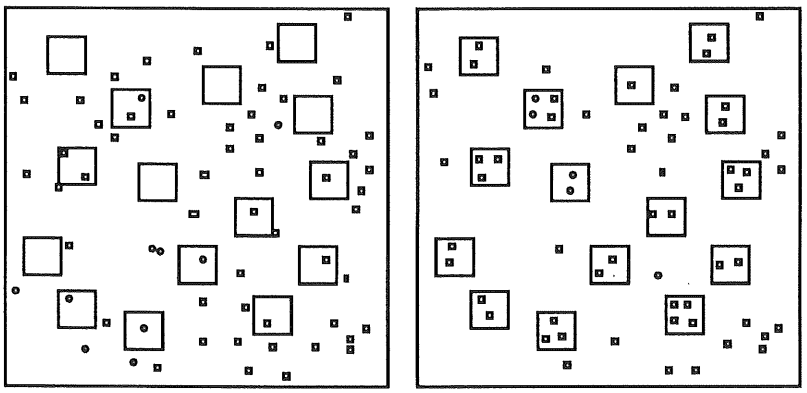


Figure 8 A typical GLVQ representation of the data.
The initial starting positions of the weight vectors is shown in the left hand picture. The final positions of the vectors are on the right picture, which shows that many of the weight vectors have not moved into positions to classify input data points.

4.1.4 Comparisons with Other Networks

Figure 9 shows the average performance of all seven types of network used. The Neural Gas network was certainly the best performer over this data and proved to be resilient to poor initial conditions. The GLVQ was clearly better than any of the others. Its distortion measure doubled from 1.47 to 2.99 under poor starting positions which was a little surprising given its claim to be position independent. However even taking the worst figure it still outperformed all the networks except the Neural Gas.

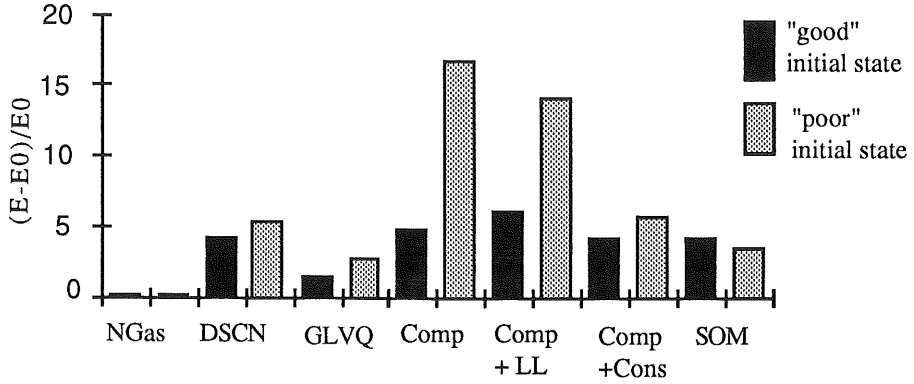


Figure 9 Average performance of the networks
The average performance of each network is shown from both good and poor initial positions.E0 is the known global minimum of E.

The DSCN network performed comparably with the Competitive network with a conscience mechanism, and a little worse than the SOM. All three were not badly effected by poor starting positions. The standard competitive network , and the competitive network with a leaky learning mechanism produced acceptable performances over the data with good starting positions . Both networks did very badly with poor initial conditions thus emphasizing the random nature of the solution found with such networks, unless good initial positions can be guaranteed.

4.1.5 Convergence Rates

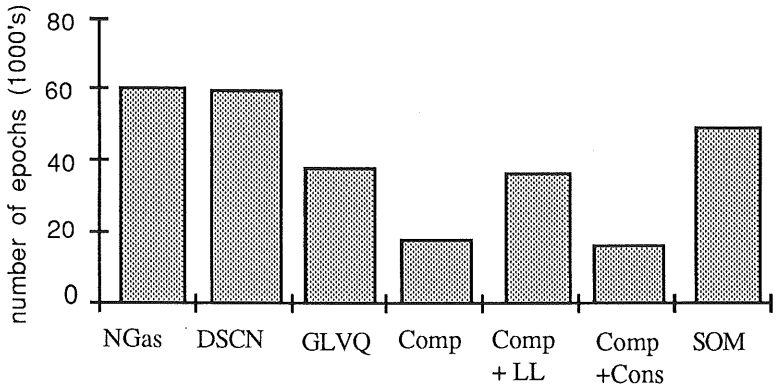


Figure 10 : the number of presentations of inputs required for each network to converge to its optimum solution.

Figure 10 shows the number of input presentations required by the networks to converge to their optimum solutions. It should be noted that the networks that took longer to find their optimum solutions , such as the Neural gas

network and the SOM would still converge to a reasonable solution with a smaller number of presentations.

The Neural Gas network clearly outperformed the others over this data, as was to be expected given that Martinetz et al. used this data in the original Neural Gas paper. The Neural Gas network is well suited to clustered data, as also it appears is the GLVQ. The DSCN was not suited to this type of data and performed badly (although this could possibly be accounted for by the fact that a suitable temperature schedule was not found).

4.2 Gaussian Clusters

The three networks were also run over more standard data consisting of Gaussian clusters.

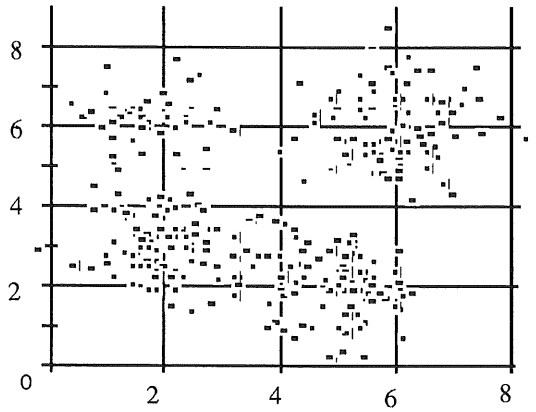


Figure 11 : Test 1 input data

4.2.1 Test 1

The first test consists of four clusters , three of which contained 100 data points and the fourth 50 data points, as shown in Figure 11. All clusters had a normal Gaussian distribution. The networks had seven competitive nodes. The optimal solution is for each of the clusters consisting of 100 data points to be represented by two of the output nodes , and the smaller cluster by just one.

For a good initial state of the weight vectors both the Neural Gas and GLVQ networks were able to find a solution close to the optimal configuration. The DSCN did not find a solution as close to the optimal configuration and consequently produced a poorer MSE.

For a poor initial weight vector state the Neural Gas performed the best with an MSE almost the same as for the good initial state data. The SOM network performed well in comparison with the other networks and outperformed

the DSCN network.

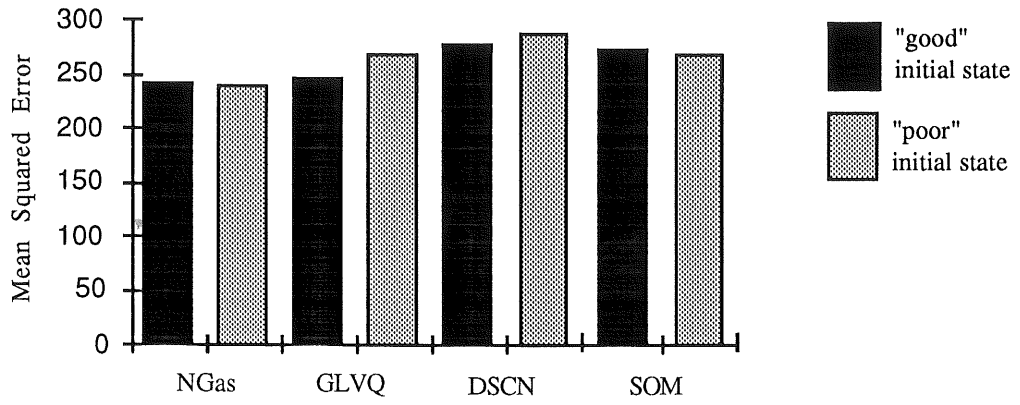
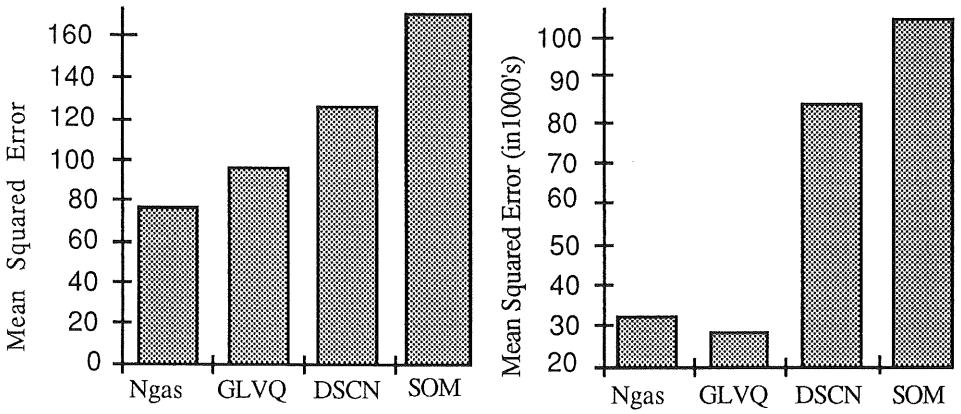


Figure 12 : Networks Mean Squared Error Performance for Test1. A good initial state exists where the initial positions of the output nodes are evenly spread across the input space. Poor initial states are defined as ones where the output nodes are initialised in asmall area of the input space away from any of the inputs.

4 . 2 . 2 Tests 2 and 3

The networks were also run over two dimensional Gaussian data with more noisy inputs . For the second test there were three dense clusters of 200 points each , and one sparse cluster covering the whole input area that gave the effect of a lot of noisy inputs. The networks had 64 competitive units.

The data was extended to four dimensions for test 3.



a) test 2 b) test 3

Figure 13 : Mean Squared Error performance for tests 2 and 3

As in all the previous tests the Neural Gas network performed the best in test 2. In test 3, where the input data was in four dimensions and effected by quite

a lot of noisy inputs the GLVQ network outperformed the Neural Gas network, though both were much better than the DSCN and SOM networks.

The problem the networks had to solve was determining how many of nodes should cover the clusters and how many the sparse noisy inputs. GLVQ tended to spread the nodes across the whole area as did the SOM. The Neural Gas network provided significantly more nodes in the dense clusters but still provided cover for the noisy inputs.

4 . 3 Gauss Markov Source

All of the previous test have used data with the inputs clustered into groups. The tests in this section were run over data with just one cluster of inputs produced from first order Gauss-Markov processes of the form $x_n = \alpha x_{n-1} + w_n$, where w_n is an independent identically distributed Gaussian process with zero mean and unit variance. The coefficient α defines the level of correlation between inputs. The tests were run with data of 2 , 4 and 8 dimensions, with differing degrees of correlation between inputs.

Correlation	NGas	GLVQ	DSCN	SOM	Comp + Cons
0.0	162	159	198	259	227
0.2	175	173	218	272	241
0.6	229	214	287	412	321
0.9	792	790	986	1356	1134

Table 1 MSE for networks over two dimensional data from a Gauss-Markov Source, with varying correlation between inputs.

Correlation	NGas	GLVQ	DSCN	SOM	Comp + Cons
0.0	356	341	376	420	385
0.5	459	446	472	579	478
0.9	1357	1380	1442	1773	1422

Table 2 MSE for networks over four dimensional data from a Gauss-Markov Source, with varying correlation between inputs.

Correlation	NGas	GLVQ	DSCN	SOM	Comp + Cons
0.0	1713	1717	1728	1908	1720
0.5	2282	2340	2398	2608	2263
0.9	6166	6215	6295	6958	6288

Table 3 MSE for networks over eight dimensional data from a Gauss-Markov Source, with varying correlation between inputs.

Tables 1 , 2 and 3 show the MSE for each network over 2 , 4 and 8 dimensional data respectively. The lowest MSE for each correlation is underlined in each table.

Over this single source data the GLVQ and Neural Gas networks produced very similar results. The DSCN did not perform as well as had been hoped over this data, and, in fact, the Competitive network with a conscience mechanism produced results that were similar and sometimes better than the DSCN over this data.

5 . 0 Network Evaluations and Conclusions

5 . 1 Does Soft Competition provide enhanced performance

The Neural Gas and GLVQ networks consistently performed better than the standard networks in terms of the MSE produced over all the tests performed. The relative performance is improved most noticeably when the data consists of more than one cluster and there are a large number of nodes. Over single source data the enhancements are not as great especially when the dimensionality is increased.

5 . 2 How the networks compare with each other .

The GLVQ and Neural Gas networks outperformed all the other networks over all the tests run. Over data with more than one cluster Neural Gas is certainly the best performer, in terms of the MSE reached. For single source data Neural Gas and GLVQ produced fairly equal performances. The DSCN network generally outperformed the standard networks but did not do as well as the Neural Gas and GLVQ networks. The main problem encountered by the DSCN was that the nodes converged to the same point if the temperature was too high. The network therfore had to be started with a low temperature that caused it to be too static and unable to find the optimum solution.

5 . 3 Sensitivity to parameters and initial conditions

The DSCN is very sensitive to the temperature value. If this value is too high the network provides very poor performance, if it is too small the network again performs badly. The GLVQ and Neural Gas appear to be relatively resilient to alterations in parameter values.

A major strength of both the Neural Gas and DSCN networks is the fact that they appear empirically to be resilient to the quality of the starting positions. The other networks tended to be quite badly effected by poor starting conditions.

5 . 4 Computational load and Speed of Convergence

The DSCN and Neural Gas networks both take up to twice as long to converge than the other networks. This is caused by the need to allow time for a suitable temperature schedule.

Computationally the DSCN and Neural Gas are again the most expensive, the Neural Gas network needs to rank all of the nodes for every input and calculate their individual update rates. The DSCN has to calculate the probability of each node winning according to a Gibbs distribution, then calculate the update value for every node before updating them all. The GLVQ network is not significantly more computationally expensive than a FSCN or SOM.

5.5 Conclusions

The Neural Gas network provides a reliably good MSE over many different data types , and is resilient to initial starting positions and variations in parameter settings. This is however gained at the expense of an increased computational load and slower convergence rate. The GLVQ network does not perform quite as well as the Neural Gas network over certain types of data and is not as resilient to poor starting conditions. It does, though, perform better than the standard networks converging at a similar rate and with a computational load equivalent to all but the standard (winner update only) CLNN. The DSCN network is capable of better performance than the standard networks but is overly sensitive to the temperature schedule implemented.

References

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

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

[3] **Pal , N . , Bezdec , J. and Tsao, E.,** "Generalised Clustering Networks and Kohonen's Self-Organising Scheme." *IEEE transactions on Neural Networks* , July 1993, vol. 44 (4).

[4] **Yair , E. , Zeger , K. and Gersho, A.** , "Competitive Learning and Soft Competition for Vector Quantiser Design." *IEEE transactions on Signal Processing,* 1992 . vol. 40 (2)

[5] **Linde, Y., Buzo, A. and Gray, R.M.,** "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28,pp. 84-95, Jan 1980.

[6] **Metropolis , N . , Rosenbluth , A. W. , Rosenbluth , M.N. ,Teller , A.A . and Teller , E.** "Equations of state calculations by fast computing machines," *Journal Chemical Physics .*, 1953, vol. 21, pp. 1087-1091.

[7] **Kirkpatrick , S. , Gelatt C . D . and Vecchi , M. P. ,** "Optimisation by simulated annealing," *Science,* 1983, vol. 220 , pp 671-680.

[8] **Kohonen,T.** , *"Self Organisation and Associative memory"*. Third Edition Springer Verlag, 1989.

[9] **Rose , K . , Gurewitz , F . and Fox , G . ,** "Statistical mechanics and phase transitions in clustering," *Physical Review Letters,* 1990 , vol. 65(8), pp 945-948.

[10] **DeSeino, D .** "Adding a conscience to competitive learning," in *Proceedings IEEE International Conference Neural Networks,* 1988,pp .I117-I124.

[11] **Ahalt, S.C. , Krishnamurthy, A. K., Chen, P., and Melton , D.E.,** "Competitive learning algorithms for vector quantization," *Neural Networks,* vol. 3,no. 3, pp. 277-290, 1990.

[12] **Krishnamurthy, A. K., Ahalt, S.C. , Melton , D.E. and Chen, P.,** "Neural Networks for Vector Quantisation of Speech and Images,"*IEEE Journal on Selected Areas in Communications,* vol. 8, no. 8, pp. 1449-1457,1990.