6. Discussion

6.1 Introduction

This chapter discusses the merits and limitations of each algorithm in the context of the aims of dynamic Neural Networks and compares the operation of the networks on the different input data sets.

Comparisons with static networks were not included because an analytical comparison is sufficient to show the differences. A SOM used with the non-stationary datasets would have only showed that the SOM would incorrectly classify those classes that it was not shown during training.

6.2 Plastic Self Organising Map

The Plastic Self Organising Map was presented in Chapter 2. The aim was to create a Self Organising Map (SOM) that could learn online using non-stationary input sets with an unbounded number of input classes and to do this with only minimal information about the input set. It has been shown that the PSOM could operate on non-stationary inputs without specifying the classes themselves. Some information pertaining to the expected characteristics of a class and their temporal persistence is required for setting the values of parameters a_n, a_{cl} and b_a. Setting these values does not require information regarding the number or position of classes that the network will need to classify. If the PSOM can operate for an arbitrarily large number of iterations without any detrimental effects to the network (such as instability or loss of all neurons), then one can extrapolate to infinite time. Given the test results (Chapter 3), it can be seen that the PSOM can locate classes in the input space for an indefinite amount of time without losing the capability to learn.

6.2.1 Properties of the PSOM

Unlike static Neural Networks, the PSOM continually alters its topology to compensate for the non-stationary nature of the input data.

PSOM does not preserve topographical neighbours, like its counterparts the SOM, GSOM and GNG-U. Although the structure of the network provides some information about the nature of the input data, the locations of the neurons and links on the graph do not faithfully represent a two dimensional view of a high dimensional dataset. For any application where only the similarity of different classes is important, the PSOM technique would not be suitable because the links cannot be used as a measure of spatial similarity alone. The PSOM can be forced to show purely topographical information by setting the link ageing parameter, b_a, to zero. By doing this, temporal information will not be absorbed, but links might still be cut if two neurons are dissimilar enough to such that the Euclidean distance between them is larger than the link breaking threshold, a_r. When b_a is set to zero, links only break through dissimilarity without any temporal interference. This reduces the PSOM to an incremental network, similar to GNG. The PSOM can represent the topology for any arbitrary number of dimensions, whereas GNG-U is limited to the ability to display the network. Therefore it may only represent up to 3 dimensions.

The network does show coarse temporal information about the input space. This is a direct consequence of the ageing process (when $b_a > 0$). Clusters where the neurons are connected by short links represent classes that are persistent within the input set. Those clusters where the neurons are connected by long links correspond to those classes that either occur irregularly or are no longer persistent in the input set and are in the process of being removed.

Given that E is the spatially continuous input space, let A denote the topographical output space that is the arrangements of neurons and links in the network. Let Φ_t denote a non-linear transformation in time and space called a *temporal-spatial feature map*. This feature map maps the input space, E onto the output space, A. This is shown by:

$$\Phi_t : E \rightarrow A \tag{6.1}$$

This differs from typical SOM feature mapping as the feature map encodes time as well as spatial properties.

Any class that has not had a representative input for many cycles is likely to be forgotten by the PSOM network. This is dependent on the ageing parameter, b_a. The neurons associated with this class will become unlinked and then removed. This is a necessary side-effect for such a process, as it must be assumed that a class may not persist during the operation of the network. If the network retained the neurons for an obsolete class, they would still be included in the search for the focus, and thus would make the network inefficient. Information is lost when a neuron is removed from the network. For this to happen, the neuron must have no links connected to it. A useful measure is *temporal persistence*, $t_p(x_p)$, which is the number of iterations a linked neuron, x_p, will persist after the last time it was selected as the focus. Assume the neurons that x_p is connected to do not affect x_p or the links connected to x_p. Let c_s be the length of the shortest link connected to x_p. Temporal persistence is given by:

$$t_p(x_p) = \frac{(a_r - c_s)}{b_a} \tag{6.2}$$

The shortest link, c_s, is a measure of dissimilarity. If its value is large, then neurons it connects are dissimilar and vice versa. As c_s becomes large, the temporal persistence of x_p is diminished. Intuitively, those neuron clusters that are not very stable initially (many long links) are more likely to be lost more quickly than clusters with short links. This is desirable in many cases.

The equation for temporal persistence (Equation 6.2) can be re-arranged thus:

$$b_a = \frac{\left(a_r - c_s\right)}{t_p\left(x_p\right)} \tag{6.3}$$

The following assumptions can then be made:

- The upper length limit for a link, a_r, can be set to 90 as all other values are proportional to this.

- Given that a_r is 90, one can say that a group of neurons through both time and space would give link lengths of less than 40. Thus c_s can be set to 40.
- The temporal persistence can be measured from a sample of the input data or approximated. This is the longest time between two presentations of a class.

This gives an ad-hoc method for selecting an ageing parameter, given some information regarding the input space.

Like the SOM, the PSOM reproduces the size of classes in the input space. Those classes that cover large sections of the input space, *E*, are represented by more neurons in the neuron space, *A*. This is because each neuron can at best represent an area of the input space governed by the clustering threshold, a_n. If a single class represents more of this space than a_n will allow, then more neurons will be devoted to it.

The PSOM also differs from typical Neural Networks in that it does not require stopping criteria. By the very nature of the data, the PSOM is required to learn for all time. However, by stopping the input data the PSOM may be frozen and evaluated offline.

6.2.2 Merits of the PSOM

The following are merits of the PSOM given by the test scenarios in Chapter 3:

- *Operates for infinite time.* Although not tested, it can be seen that by treating each input pattern identically (by not scaling any parameter by the total time) the PSOM will contain a bounded number of neurons. Traditional Neural Networks (MLP, SOM) have some form of decay parameter (such as learning rate) and thus treat patterns later in the data set differently.
- *No need to set topology a-priori.* The network grows and shrinks to meet the needs of the data set. This is demonstrated in PSOM Test 1 (Chapter 3) where the number of links (and therefore neurons) of the network changed over time (Chapter 3, Figure 3.7). The network both grew to accept new classes and decreased in size to remove old classes, thus operating in a dynamic manner.

This feature is also shared by growing networks (such as Cascade Correlation).

- *Operates in a non-stationary environment.* This can be seen in PSOM Test 1 where a new class was presented to the network after 800 iterations. Rather than reject the class, as a static network would, the PSOM accepted it and grew its topology to accept the new class. The new class does not need to be within the same bounds as the existing classes. After the acceptance of a new class, the network can then interpolate between the given classes. Only the GNG-U is known to operate in this manner. All other Neural Networks (even growing ones) require a fixed number of classes in the input set prior to training.
- *The PSOM can work in a noisy environment.* PSOM Test 5 (Chapter 3) shows that the PSOM can discriminate persistent classes from 'noisy data' while using a bounded number of neurons and links. When classifying noisy data, there is greater sensitivity to the network parameters. This is not novel, as most Neural Networks can operate with noisy inputs.
- *The parameters of the PSOM, a_n, a_{cl} and b_a are robust as shown in the many parameter tests.* This means that the operation of the PSOM is not sensitive to the accurate setting of its parameters. Some heuristics can be used to set initial values of these parameters (Chapter 2). This is not a novel feature; Neural Networks are particularly good at operating with sub-optimal structures or parameter settings.
- *The PSOM may represent an input space of any dimensionality.* Unlike GNG-U, the PSOM may represent any number of input dimensions.

6.2.3 Limitations of the PSOM

Although the PSOM met the goal of a novel, dynamic, Neural Network; there are some limitations to its operation. These limitations may make the PSOM unsuitable for some applications.

- *Some information regarding the size and temporal longevity of the classes in the input set is required before training.* This information is used in the heuristics to set the parameters of the network. This kind of information is not

required by static Neural Networks, although the ordering of the patterns is often a factor in training (pp.23 Haykin 1999).

- *It is difficult to set parameters iteratively*. If there is no *a priori* information about the input data set, one might attempt to set the network parameters iteratively. This is a difficult process because any network output is not dependent on a single parameter. High recognition error, for example, could be due to either large values of the Node Building Parameter, Clustering Threshold *or* Link Ageing Parameter. Of course, one may set these values arbitrarily small and then increase them iteratively but this has no guarantee of convergence. This is a limitation that all Neural Networks suffer from. Static Neural Networks require not only the parameters (such as neighbourhood size or learning rate) but the topology of the network to be set before training.
- *The PSOM does not preserve spatial topology of the input space*. This means that two neurons connected with long links in the PSOM network are not necessarily dissimilar in terms of input space. These long links might mean that the neurons represent points in the input space that are *close* together but have not been presented to the network for a long time. This may be undesirable to those wishing to have a precise map of the input space. This can be achieved by setting the Link Ageing parameter to 0, but then there is little ability beyond the neighbourhood function (controlled ultimately by the Node Building Parameter) to break links between dissimilar neurons. This is in contrast to the GNG family of algorithms (Fritzke 1994-1997) and the SOM (Kohonen 1982), which all preserve topology in the input space. Thus, the PSOM is not suitable for applications that involve the creation of a topologically correct map of the input space.
- *Parameters are tuned to known data and are static thereafter*. This is a concern if the classes captured prior to training (for setting the network parameters) are not representative of the system as a whole. If the classes used to set the parameters (using the heuristics given in Chapter 2) are much larger than the classes presented later, then two different classes may be classified as one. This is undesirable, but one must assume that some information about the properties of the system is known a-priori.

- *Detection of novelty is difficult in noisy environments.* Novelty is noted when the recognition error output of the PSOM rises above the node building parameter (a_n) threshold. In noisy environments, this occurs often, and new neurons are added. These neurons are unlikely to persist (as they correspond to one-off "noise classes") so they cannot be considered as novel classes. An additional system can be added to monitor the neurons of the new classes, but this addition is not in the spirit of the PSOM.

6.3 Extended Plastic Self Organising Map

The EPSOM was presented in Chapter 2 to solve some of the problems with the PSOM. The clustering threshold, a_{cl}, controlled the neighbourhood of the PSOM and it was found in the initial tests that this parameter had little effect on the ability of the network to cluster. Therefore, the neighbourhood function was replaced with a smooth sinc function. Also, the link lengths of the EPSOM were scaled against a_n to allow the neighbourhood function to more effectively break up dissimilar groups. Further testing showed that the lack of control of the cluster size led to difficulties in setting the node building parameter and so the PSOM gave better results in testing on the real world Radar dataset.

6.3.1 Merits of the EPSOM

The merits of the EPSOM are similar to those of the PSOM. The following merits are additional to those mentioned for the PSOM earlier.

- *Fewer variables to set.* The PSOM requires the Cluster Threshold, Node Building Threshold and the Link Ageing Parameters to be set. The EPSOM does not use the clustering threshold and thus there is one fewer variable to set.
- *Smooth neighbourhood function.* The discontinuous neighbourhood function used by the PSOM has the effect of pushing apart very similar neurons on either side of the clustering threshold. This EPSOM treats all similar neurons in a similar manner. This function is similar to that in the original SOM algorithm (Kohonen 1982).

6.3.2 Limitations of the EPSOM

The limitations of the EPSOM were much the same as those given for the PSOM with the following additional factors.

- *Difficult to set node building parameter.* As the neighbourhood function now depended purely on the node building parameter, it needed to be set such that the outlying patterns in of a class were still classified the same as the typical patterns from that class while at the same time patterns from different classes were classified differently. With the EPSOM, the clustering threshold is a proportion of the node building parameter and this proportion could not be changed to suit the classes of the input set. This caused problems when labelling the neurons with the correct class number.

6.4 Temporal Plastic Self Organising Map

The TPSOM was introduced in Chapter 4. The TPSOM was created by altering what the links represented and then adjusting the learning algorithm. The motivation for this was to encode purely temporal information in the links and thus preserve the topology of the temporal input space. It was found after the creation and initial testing of the TPSOM, that the resulting graphs and structures were similar to those of Markov models. The TPSOM classifies as a Dynamic Neural Network (as defined in Chapter 1) because it operates on non-stationary and potentially infinite pattern sets. The TPSOM was tested on normal data sets (e.g. TPSOM Test 1), those generated by Markov processes (e.g. TPSOM Test 4) and a real world example of some text (TPSOM Test 8). For noise-free data sets, the TPSOM found structure within the data sets and produced graphs that compared favourably with the data. The TPSOM performed less well on the noisy artificial data set, where patterns belonging to a noise class in the input space were added to the data set. The structure produced showed little evidence of any definition. This was probably due to the presence of a high density of noise patterns compared to the density of the desired patterns. A lack of structure was also found in the Radar Data set, but this was expected as the manner in which the radar pulses arrive tends to be random. The TPSOM was shown to classify well, with low recognition error and consistent labelling classification.

The Markov Chain tests were of two sorts. Firstly, a static Markov Chain, where the state transition matrix (STM) of the Markov Chain generator does not change and secondly a Dynamic Markov Chain, modelled as two static chains with switching from one to the other. For the static Markov Chain, it was found that the error for the difference in system STM and TPSOM STM reduced over time until it reached a value of 0.12 after 600 iterations. This is not a final value, as it is expected that the error $\rightarrow 0$ as time $\rightarrow \infty$. The dynamic Markov Chain used two STMs to represent two static Markov chains and then switched from one to the other after 500 iterations. The classes themselves did not change, just their order. The TPSOM managed to alter its topology such that it fitted the new STM after 130 iterations from the change.

The final dataset was the real-world vowel pair dataset. In this test, the vowel pairs from Charles Dickens' Great Expectations were presented to the network. The whole text was to be represented by the network and so link removal was switched off by setting the link removal threshold to 0. The difficulty came in trying to set the learning rate such that an accurate representation of the text was produced. When the learning rate was large, then the most recent patterns had more effect on the structure of the network. With a small learning rate, the probabilities on the links did not differ sufficiently from the initial link values. This could be improved by changing the manner in which new links are added and existing links are updated. Further tuning might be performed but no major improvement in performance is expected.

6.4.1 Merits of the TPSOM

- *TPSOM can represent Markov-like processes*. The TPSOM can be used to represent Markov chain processes without any prior knowledge of the number of classes.
- *TPSOM can represent switching Markov processes*. Typical Markov chains are static, the order of the classes does not change during operation. However, the TPSOM can operate on switching Markov processes. Typical Markov systems are assumed to be a single static chain, but with the TPSOM, the chain can be modified during learning.

- *TPSOM gives a temporally correct map of the input space.* The graph output of the TPSOM gives an unambiguous temporal representation of the input space.
- *TPSOM can be used to represent the structure of a piece of text by representing its vowel pairs.*

6.4.2 Limitations of the TPSOM

Although the TPSOM could represent a Markov chain, there are some limitations to its operations and areas where its performance could be improved:

- *Only one neuron represents a class.* Non-regular patterns are often better represented with multiple neurons. However, in the TPSOM, there is one neuron per class. Most other Neural Networks use several neurons to represent a single class. In this manner, the TPSOM is less robust than static techniques.
- *Sensitive to noise.* The addition of a noise class can reduce the clarity of the graph diagram such that little understanding can be gleaned from it. The only information that can be gained is the presence of noise or the lack of structure.
- *Parameter Sensitivity.* As a result of having only one neuron per class, the TPSOM is more sensitive to the setting of its parameters. Static Neural Networks outperform the TPSOM in this respect as having more than one neuron to represent a class allows them to be robust against marginally incorrect parameter settings.
- *Learning Rate Sensitivity for static systems.* When dealing with static systems, the setting of the learning rate is a compromise between the initial link values and the need for memory storage. The learning rate controls the length of the TPSOM's memory through time.

6.5 Network Comparisons

The following table (Table 6.1) is broken into two sections, firstly to outline the main differences in structure and algorithm between the four types of network investigated in this thesis and to include the results from the technique evaluation. Test numbers

are given for reference back to the results in Chapters 3 and 5. Each test number corresponds to the algorithm test in the respective column. For example, Test 4 in the second column (PSOM) equates to PSOM Test 4, given in Chapter 3.

Table 6.1 Comparison of the dynamic Neural Networks

Facet or Test	PSOM	GNG-U	EPSOM	TPSOM
Structure / Algorithm Comparison				
Neuron Structure	Reference vector	Reference Vector	Reference vector	Reference vector
Link represents…	Part spatial, part temporal	Spatial.	Part spatial, part temporal. Scaled given the node building parameter	Probability of the next input pattern belonging to the next neuron
Links unidirectional?	No	No	No	Yes
Initial topology	3 neurons, 2 links	0 neurons, 0 links	3 neurons, 2 links	1 neuron, 0 links
Parameters	Node Building, a_n Clustering Threshold, a_{cl} Link Ageing, b_a	Node adding parameter (λ), Node update amount (ε_b), Neighbourhood update amount (ε_n), Maximum link length (a_{max}), Error Reduction (α), Forgetting factor (β),Utility Threshold Measure (k), Maximum number of nodes	Node Building, a_n Link Ageing, b_a	Node Building, a_n Link Ageing, b_a Learning Rate, b_r
Neighbourhood function type	Non-linear discontinuity	Linear	Smooth 'sinc' function	Probabilities balanced
Ageing	Link aged by value of b_a parameter each iteration	Neuron utility and error decline by α and β respectively.	Link aged by value of b_a parameter each iteration	No link ageing
Input Class represented by single neuron or group?	Group of neurons	Group of neurons	Group of neurons	Single neuron
Test Comparison				
Artificial Data Set (5 or 7 classes)	5 Classes. Classified classes correctly.	Not Applicable	5 Classes. Classified classes correctly.	7 Classes. Classified classes correctly.
Effect of Node Building Parameter (a_n) – Artificial Data Set	Test 2. Range of a_n is 0.2 and 0.45. Small values of a_n lead to low errors, large networks and many classes found.	Not Applicable	Test 2. Range of a_n is 0.15 to 0.4. Small values of a_n lead to low errors, large networks and many classes found.	Test 6. Range of a_n is 0.2 to 0.35. Small values of a_n lead to low errors, large networks and many classes found.
Effect of Cluster	Test 3. No	Not Applicable	Not Applicable	Not Applicable

Threshold (a_{cl}) – Artificial Data Set	appreciable effect on the network. High values lead to poor classification.			
Effect of Link Aging Parameter (b_a) – Artificial Data Set	Test 4. When set to 0, the network retains all information. When >2, then the network reclassifies each new pattern as a new input.	Not Applicable	Test 3. When set to 0, the network retains all information. When >4, then the network reclassifies each new pattern as a new input.	Not Applicable
Effect of Learning Rate (b_r) – Artificial Data Set	Not Applicable	Not Applicable	Not Applicable	Test 7. Range is 0.001 to 0.05. Small values mean that each pattern has little effect on the network and vice versa.
Pure Noise Pattern Rejection Capabilities	Test 5. Can still classify persistent classes but less persistent ones can be split. Difficult to detect novelty in noisy environments.	Not Applicable	Test 4. Can still classify persistent classes but less persistent ones can be split. Difficult to detect novelty in noisy environments. Classified better than the PSOM.	Test 2. Only the very persistent classes give any structure to the network.
Effect of Node Building Parameter (a_n) – Noisy Artificial Data Set	Test 6. Small a_n leads to large networks and smaller error. As with noiseless system but with a smaller range: $0.2 < a_n < 0.3$	Not Applicable	Test 5. Small a_n leads to large networks and smaller error. As with noiseless system but with a smaller range: $0.2 < a_n < 0.35$. Larger range than the PSOM.	Not Applicable
Effect of Cluster Threshold (a_{cl}) – Noisy Artificial Data Set	Test 7. Has little effect on the operation of the network, although should be kept small to aid link ageing process.	Not Applicable	Not Applicable	Not Applicable
Effect of Link Ageing (b_a) – Noisy Artificial Data Set	Test 8. Small values gives low error but large networks and vice versa. Setting b_a to 0 is not recommended as network will not be able to reject the noise class.	Not Applicable	Test 6. Small values give a low error but large networks and vice versa. However, b_a does not have much effect on the very persistent classes.	Not applicable

Radar Data Set	Test 9. Can classify individual radar pulses with a bounded error. Also, the novelty of new pulses can be detected when the error spikes above the norm.	Not Applicable	Test 7. Did not classify as well as PSOM. Possibly due to a less well defined neighbourhood edge.	Test 3. Aim was not to classify but to investigate if there was any structure. No temporal structure was found, as expected.
Static Markov Chain	Not applicable	Not Applicable	Not applicable	Test 4. Temporal map of the input system created automatically. STM error was small (0.12) and still decreasing after 600 iterations.
Dynamic Markov Chain	Not Applicable	Not Applicable	Not Applicable	Test 5. Although the STM of the input system changed after 500 iterations, the network adjusted.
Vowel Pairs	Not Applicable	Not Applicable	Not Applicable	Test 8. Managed to find major features.
2D Data Set	Classified both classes after 50 input classes after 170 iterations. Switched to new distribution without problem.	Classified both classes after 5000 iterations. Switch to new distribution quickly and without problem.	Not Applicable	Not Applicable
Application Area	Detection of novelty and dynamic classification in noise-free systems.	Representation of topology of unknown, non-stationary input distribution.	Detection of novelty and dynamic classification in noisy systems.	Investigation of temporal structure in systems and representation of dynamic Markov systems. Representation of text.

The (E)PSOM algorithms were tested with the Markov Chains. No additional temporal structure could be gleaned from the output graph of the network and the output was similar to that found with the tests with the other artificial sources. The poor operation of the (E)PSOM algorithm on the radar dataset might be due to the irregular shape of the class boundaries. The artificial data set uses square-shaped class boundaries and the neighbourhood function of the (E)PSOM is hyper-spherical. Irregular shapes should be better classified with an algorithm where each neuron

covers a different volume of the input space (such as the GPSOM, given in Chapter 7).

6.5.1 PSOM vs EPSOM

The EPSOM gives improved performance on the artificial data sets and is easier to initialise as there is one fewer variable to set. The PSOM uses a sharper neighbourhood function, which results in it clustering well defined classes better than the EPSOM. Unfortunately, this only occurs in a noise-free environment and the EPSOM outperforms the PSOM when noise is present. The labelling algorithm of the EPSOM is not as efficient as that of the PSOM as there is no way of controlling the cluster size. The EPSOM has a tendency to classify a class covering a large area of the input space as two separate classes. This is particularly noticeable in the radar dataset, where sensitivity to the Node Building Parameter led to incorrect labelling of the EPSOM neurons.

6.5.2 (E)PSOM vs TPSOM

The (E)PSOM algorithms are treated as one when comparing with the TPSOM. It is important to remember that the motivation for the TPSOM was that the topology of the (E)PSOM graphs gave ambiguous information about the input space. It was possible to get the (E)PSOM to give a spatial map by setting the Link Ageing parameter to 0, but then the algorithms became very similar to the GNG algorithms (Fritzke 1994-1997). The TPSOM only uses one neuron per class and thus represents a class using one hyper-sphere, whereas the (E)PSOM topologies use many neurons for one class and thus may represent any arbitrarily shaped class. The TPSOM is more useful when the spatial properties of the system classes are well understood but the temporal information requires investigation. The (E)PSOM can give little unambiguous information about the temporal structure of the data set but can classify more reliably.

6.5.3 (E)PSOM vs GNG-U

The (E)PSOM algorithms are treated as one when comparing with GNG-U. Initially, the GNG-U algorithm requires a large number of parameters to be set at the start.

Some of these can be arbitrarily set but there are no heuristics recommended for this process. Given the tuned parameters in Fritzke's work, GNG-U requires many more iterations to form a representation but does so without any knowledge of the pattern densities in the input space. GNG-U also preserves the topology of the input space, something that the (E)PSOM does not. Preservation of the input topology is desirable for systems with 3 dimensions or less but for high dimension systems, GNG-U in its current form will not represent the system at all. All members of the PSOM family can represent an arbitrary number of dimensions. Also, GNG-U requires a stopping criterion, in that there is a concept of the maximum number of nodes that the network can have. Without this parameter, GNG-U will continue adding neurons to the simulation, whilst decreasing the overall average utility. If there are more neurons representing the same region of input space, the usefulness (or utility) of each neuron is decreased. This in turn increases the sensitivity of the algorithm to the removal of neurons. The difficulty comes when trying to decide how many neurons will be required for representing the unknown input space. Both the (E)PSOM and GNG-U suffer if this a-priori knowledge regarding the input space is not available.