

Designing a Layered Network for Context Sensitive Pattern Classification.

Neil A. Thacker, John E. W. Mayhew.

Last updated
6 / 9 / 2005

This document forms part of the **Recognition and Intelligence Series** available from www.tina-vision.net.

- 2007-001 Retinal Sampling, Feature Detection and Saccades:
A Statistical Perspective.
- 2006-008 Statistical Principles for Selection of Computer Vision Algorithms as
Modules for Visual Perception - Show Me the Errors.
- 1991-001 Designing a Layered Network for Context Sensitive Pattern Classification.
- 1997-002 Supervised Learning Extensions to the CLAM Network.
- 1996-003 Tutorial: Algorithms For 2-Dimensional Object Recognition.
- 1997-005 Speechreading Using Probabilistic Models.
- 2000-002 Solving Shape Based Object Recognition from a Computational Standpoint -
Practical and Physiological Constraints.
- 1995-004 Assessing the Completeness Properties of Pairwise Geometric Histograms.
- 1996-004 Robust Recognition of Scaled Shapes Using Pairwise Geometric Histograms.
- 1996-005 Multiple Shape Recognition Using Pairwise Geometric Histogram Based Algorithms.
- 2007-007 Automatic Identification of Morphometric Landmarks in Digital Images.
- 1999-002 A Feature Representation for Map Building and Path Planning.
- 2001-015 Colour Image Segmentation by Non-Parametric Density Estimation in Colour Space.
- 2001-006 What is Intelligence?: Generalised Serial Problem Solving.
- 1994-002 A Correlation Chip for Stereo Vision.
- 1995-001 Specification and Design of a General Purpose Image Processing Chip.



Imaging Science and Biomedical Engineering Division,
Medical School, University of Manchester,
Stopford Building, Oxford Road,
Manchester, M13 9PT.

Designing a Layered Network for Context Sensitive Pattern Classification.

Neil A. Thacker, John E. W. Mayhew. while at Artificial Intelligence Vision Research Unit University of Sheffield.

Preface

This paper marked our initial attempt to make use of artificial neural networks as the key component in a learning machine vision system. Many other publications followed over the next 10 years and some of the basic ideas, tentatively suggested here, became better worked out. In particular, after several years of searching we finally managed to show that the process of square-rooting the values in a pattern template is the appropriate way to construct a dot product similarity score for Poisson sampled data (Tina memos 1997-001 and 2001-010). We succeeded in producing an invertible (complete) representation of line shape, which could be therefore guaranteed to solve the ambiguity issues presented in this paper (Tina memos 1995-004 and 2002-002). We also managed to generate an architecture which combined the possibility of associative layering with supervised training (Tina memo 1997-002). The combination of these ideas produces a statistically principled approach which has an optimal representation for recognition of line based shape in a system which is guaranteed to learn. As this paper illustrates, all of this was based (from the start) on the physiologically motivated models being developed at the time by Grossberg.

Though other workers have applied statistical principles to network design we believe that we are still the only ones to have identified the link between frequency coding of signals and this way of performing a statistical comparison on a physiologically plausible neural architecture. In addition, though other authors in the area of image analysis (and particularly database retrieval) have since re-discovered the use of histogrammes for data indexing, we believe that we are still the only group which has motivated this approach from the view of a statistical theory of learning. Further ideas, regarding extending this work to more fully solve problems in artificial intelligence, are presented in Tina memo 2001-006. These ideas were intended to form the basis of a research program, but were regrettably shelved as the available funding opportunities within the UK pulled in other directions. It has however, led on to an attempt to relate these ideas to human perception using fMRI studies (Tina memo 2003-012).

Abstract

Many pattern recognition problems in the real world are complicated by noise and imperfect data. It is widely accepted that recognition of such data would be greatly aided by the use of contextual sensibility of the classification. One possible solution to this problem would require a multi-layered pattern classifying network with feedback mechanisms. Recognition of this possibility has led to the development of CLAM (Contextual Layered Associative Memory), an attempt to extend conventional one-layered models to permit layering. Layering has been made possible using an information preserving probabilistic approach to simulate the effect of uncertainty within the network. The new network has been designed to be robust under noise while still maintaining enough flexibility to learn new patterns. This is achieved by a combination of a novel new-node generation algorithm and a simple resonance mechanism. The network has a fixed number of layers which are used to classify accumulated classifications from previous layers. The number of nodes in the network is flexible and a complete classification network can be grown from just a few seed nodes during the course of training. Connectivity is also flexible, connections are generated and maintained according to the demands of the training data. This network is not meant to be a complete solution to the problem of context sensitive classification but a step towards making such networks possible. Its use is demonstrated in the recognition of planar objects given edge vectors.

Keywords: *neural nets, pattern classification, information preservation, flexible architectures, learning.*

What should networks be learning?

Parallel neural networks of the type described by Kohonen, Grossberg and Hopfield (see Lippman 1987 for a review) have been shown to be well suited to the task of pattern recognition. Generally this involves setting up a network architecture, and then training the network on a set of sample data. The resulting network is then found to be capable of classifying subsequent input in terms of a set of learned exemplars. This result is not surprising as the network learning algorithms are in most respects identical to the "nearest neighbour" classifiers, such as k-means clustering, used in conventional pattern recognition. Generally these classifiers learn to associate a given set of data, in the form of multi-dimensional vectors, with a set of unit vector exemplar patterns z_{subj} . Association of an input vector during and after training with a particular exemplar J is done by taking the one which appears to be the shortest Euclidean distance from the input vector pattern I , which is the minimum of

$$\sum_i (I_i - z_{iJ})^2$$

which for unit vector exemplars is equivalent to finding the maximum of

$$\sum_i I_i \cdot z_{iJ}$$

which is the method used in the Bayesian classifier (and is one of the simplest possible functions of neurons and synapses in neural networks). This method necessarily implies that each component of the input vector has the same intrinsic significance, otherwise the use of the Euclidean distance as a measure of similarity is unfounded. Further, when using the dot product, the weights to each node must be normalised to ensure that the closest node is uniquely specified. This procedure is justifiable if the significance measure used for input is the square-root of something that adds linearly, like a probability. For these reasons such classifiers are not suited to classification of vector components resulting from a set of measurements on arbitrary scales. Further, single layered networks are not influenced in any way by the context in which a given pattern is found.

In general we would expect the input pattern to represent a set of invariants which are representative of the object to be recognised. Obtaining suitable invariants is itself a research question which is influenced but not automatically solved by adopting a learning network for pattern classification. This point will be demonstrated by the choice of invariants used later for object recognition.

Multi-Layering.

Despite the fact that one-layered associative networks do not permit contextual sensibility of classification there seems to be little work published on the subject of layered networks. One layer networks need context influences to be imposed outside of the network architecture [Kohonen 1988]. The possible benefits to be obtained by layering

networks are obvious, the different layers of the network could represent semantic descriptions of the input data at different levels of abstraction. Or the first layer may be used for feature recognition and the next layer for recognition of groups of features with this or some further layer eventually providing object recognition. It may be possible to employ feedback mechanisms to ensure that high and low level descriptions are consistent, so enforcing contextual constraints. An important research aim should be making multi-layering possible and this idea has been central to the work presented here.

With conventional network architecture the first layer is presented with one input pattern at a time. Generation of an input pattern to the second layer will therefore require the buffering of accumulated activity from a complete set of input patterns (defined elsewhere) to the first layer. This is a serial process operating in an otherwise totally parallel system, (parallel implementation by multiple duplication of the input layer is not considered to be a sensible solution).

Order independent pattern classification can be envisaged simply by holding the sum of all activities at the back of the first layer for input to the second. Care has to be taken to generate activity only at those nodes which have useful information so that the signal is not swamped by noise. In a winner-take-all situation only the maximum node J (nearest-neighbour) receives activity. This is clearly unsatisfactory as it conveys no information about how well the input pattern is described by this node in preference to any other. Maximum information would be preserved if the information encoded in the output were a probabilistic measure that the input pattern could be best described by each node classification. Thus we require some knowledge of the likely error on the encoded value of each feature in each pattern and also the likely error on each component of the input pattern. The error on each feature is dependent on the process used to determine the quantity (measurement error) while the errors on the weights are a measure of how accurately the network can encode the information it is given.

We use the dot product D_j to select the maximum node as this gives invariance to input magnitude and also has the useful property that connections do not need to be maintained to those components of the input pattern that are expected to be zero (or small). If the input pattern is randomly sampled, from its error space over a period of time τ , for each instance of the input pattern a winner node J is chosen and we can use the following equations to generate output O_j from this layer.

$$O_j = \sum_0^{\tau} f(D_j)\delta(j = J)$$

where $\delta(j = J)$ is one if j is the maximum node J and zero otherwise. Provided $f(D_j)$ is reasonably constant over all selected maximum nodes the output from this layer then approaches values which are proportional to the conditional probability that the time varying input pattern is consistent with the stored exemplar patterns for each node. This can easily be envisaged in a neural network as being due to fluctuations on the connection strengths between neurons and the stochastic nature of the input and output signals. The network is actually performing an integral of the input pattern over the regions defined by the node classifications. The method can be compared directly with recent psychophysical ideas on measures of similarity [Ashby and Perrin 1988]. This approach should also enable the network to perform in situations where input data is subject to measurement error, as such fluctuations will be described by smoothly varying outputs over a small group of nodes, rather than the unpredictable changes in output typical of nearest neighbour classifiers. Notice that the most probable node will generally also be the one which would have been chosen as the nearest-neighbour.

A computer Monte-Carlo simulation of this process requires the generation of many trial input patterns before a stable pattern representation is achieved. For a layered network this simulation process takes a long time and a quicker method has to be found if the idea is to be tested. For this reason an analytic approximation has been developed which will now be described.

Probability Estimation.

What we require is the probability P_k that node k would have been chosen as the closest representation of the input pattern according to the Bayesian classification measure, given expected errors on the input pattern δI_i and the connection strengths in the network δz_{ij} . To make any analytic calculation of this sort tractable we must first approximate the distribution of these errors to gaussian functions. The probability that one node K will be chosen in preference to another node L is defined as the probability that

$$R_{K>L} > 0$$

where

$$R_{K>L} = D_K - D_L$$

and

$$D_j = \sum_i (I_i \cdot z_{ij})$$

Assuming that $R_{K>L}$ is normally distributed, this probability can be calculated given that the width of the distribution can be predicted from the errors on the inputs and weights

$$\delta R_{K>L}^2 = \sum_i I_i^2 \times (\delta z_{iK}^2 + \delta z_{iL}^2) + (z_{iK} - z_{iL})^2 \times \delta I_i^2$$

It is now assumed that the probability that node K will be chosen as a maximum is proportional to the product of the probabilities that the node is more likely than any other ¹.

$$P_K = \alpha \prod_j P_{K>j} \quad j \neq K$$

The constant α is determined by enforcing conservation of total probability.

$$\sum_j P_j = 1$$

This approximation has been tested against the Monte-Carlo calculation using various simple error models of the form

$$\delta z = f(z) \quad \text{and} \quad \delta I_i = g(I_i)$$

where the arbitrary functions f and g were scaled powers of the specified parameters. If the input is derived from a frequency distribution then a suitable choice for δz would be $z^{1/2}$. More will be said later about the general problem of choosing suitable models. The method was generally found to be accurate to within a few percent, which is sufficient for our purposes. We have found that only nodes with the highest D_j values (approximately 6 - 10) are needed for accurate calculation of the largest probabilities as the series converges rapidly.

The output from the network is defined as

$$O_j = P_j D_j^2$$

The input to the next layer is then determined by the equation

$$I_j = \left(\sum_0^\tau O_j \right)^{1/2}$$

where the summation implies the temporal addition of the outputs from the first layer in response to the current set of input patterns. This choice is invariant to temporal segmentation and order of the input patterns.

Network Flexibility.

Any flexible learning network which is to be capable of learning during use cannot contain a fixed number of available nodes. Nor is it possible to specify a-priori the relationships between the nodes at each layer. The specification of an interconnected topology, as used in Kohonen's self-organising feature maps, is not applicable to such tasks. An alternative approach would be to assign new nodes whenever required by the demands of the training data. In an unsupervised learning environment a sensible aim would be to generate a node for every distinguishable pattern in the training set (subject to the expected errors in the classification process).

¹This is clearly incorrect, as it would require the computed probabilities to be independant. However, as an approximation which generates a mathematical form which provides the required characteristics it may be regarded as adequate. It can also be interpreted as a simple redefinition of the statistical sampling process.

Grossberg suggests that new node generation is done on the basis of a comparison between the input pattern and the recovered template. As the template varies with learning due to the action of the reset mechanism unlimited generation of nodes can result. We overcome this problem by assigning a new node if the probability that the input pattern is consistent with the encoded pattern is greater than some value ρ for any node. Thus a high probability is taken as an indication that this region of the pattern space is not well populated. This method has the advantage that nodes are generated with densities which are driven by the variance of the input data and pattern classification accuracy and should produce a node for each distinguishable pattern, exactly as required for unsupervised training. Node generation must eventually cease as an increase in the local node density decreases the probability of assigning a new node the next time the pattern is presented. However, the network retains the flexibility to generate new nodes if a pattern is presented which is well away from the populated region of feature space [Figure 1]. This node generation algorithm automatically ensures that the activity generated by an input pattern is distributed among several nodes, as required if the training data are to have associated errors. New nodes are initialised with weights defined by the current input pattern according to

$$z_{ij} = I_i/|I|$$

As these nodes are generated close to where they are needed subsequent training is rapid and the network gives sensible outputs after only a few presentations of data.

Training algorithm.

Conventional training algorithms, used for nearest neighbour classifiers, train the chosen node to make the exemplar pattern more like the input pattern. If the input pattern is expected to have associated errors and it is to be encoded over a group of nodes there is a weakness in this strategy as the node description does not have a well defined convergence point. Regardless of the amount of previous training the maximum node will always be moved towards the input pattern. The effects of this can be minimised by reducing the rate of learning with increasing training time. However, if the network is supposed to be used continually, and there is no limiting training time which can be specified for the network, then this solution is not relevant. The training algorithm employed by Kohonen enforces a linked topology to identify which of the exemplar patterns are similar to the nearest one and these nodes also get adjusted during training [Kohonen 1984]. It is this feature that brings about self organising properties and map generation. We would also like to exploit this training algorithm but we cannot impose a linked topology as we do not know the dimensionality of the feature space the network will be required to learn. This may not be a problem as information about the similarity of individual nodes is made explicit by the probabilities. It has been found however that simply training nodes towards the input pattern by an amount determined by their probability results in unstable learning with groups of nodes all approaching the same pattern representations.

To find a suitable training algorithm we can appeal again to information preservation. We have already used this principle to determine how activity is distributed amongst a group of nodes by making use of knowledge about the errors present in the network. Optimal encoding of the input pattern over the group of nodes should also be one from which an estimate of the input S_i can be reconstructed. There are clearly a large number of possible ways of doing this but one which has simplicity and is in line with the probabilistic interpretation of the representation is

$$S_i = \sum_j P_j D_j z_{ij}$$

Assuming that the probabilities are approximately constant, we can minimise the difference between this quantity and the input using the following training algorithm.

$$\frac{\partial z_{ij}}{\partial t} = k_j^{-1} P_j (I_i - S_i)$$

where k_{subj} is given by

$$\frac{\partial k_j}{\partial t} = P_j (D_j - \beta k_j)$$

The term k ensures that learning proceeds as if forming a weighted mean of the examples of the input pattern with flexibility for change at a level determined by β . The number of presentations required to retrain part of the network is of the order of β^{-1} , a value of $\beta = 0.1$ was generally chosen.

This training algorithm has a well defined convergence point and the network automatically stops learning once a group of nodes is capable of generating an accurate representation of the input. This algorithm is thus stable and also trains nodes into positions which are well separated. Another feature of this training algorithm is that it would be expected to work sensibly even if the input and training cycles were asynchronous, this can be used as a justification for our choice for S_i . Notice also that the method reverts to the nearest neighbour training algorithm if the probability for one node is unity, in this respect this new algorithm can be regarded as an extension of the old.

Feedback Mechanisms and Resonance.

Contextual classification will only be achieved if the results from classifications at one layer in the network can be used to influence classifications in lower levels. This implies that some feedback mechanism is required. For the network to perform optimally in its task, of recognising a group of invariants as a particular pattern, it is necessary for that pattern exemplar to have associated with it a maximum subset of the features from this group. This is the fundamental idea which Grossberg embodies in his Active Resonance Theory (ART) [Carpenter and Grossberg 1987], whereby a short term resonance process operates which modifies the input pattern according to past experience, effectively buffering the network against noise. This is done using a feedback mechanism involving bi-directional connections z_{ij} and z_{ji} between nodes L_i and L_j . A template is generated for comparison with the input and any inconsistent features are suppressed. A generalisation of this method has been developed to enable the mechanism to operate with real numbers [Thacker and Mayhew 1988]. This method makes available the feedback information which will be necessary for context sensitive classification. The resonance mechanism reported here is a within-layer process, extension across layers may provide between-layer classification consistency and so context sensibility.

The analytic probability calculation permits the comparison of various possible error models for performance in noisy conditions, where noise is defined as spurious contributions to the input which would not be described by the error model, (as frequently observed in data and generally termed flyers). In order that the classification process should not be influenced by extraneous noise on component I_n of the input pattern, both the quantity $R_{K>L}$ and its error must be unaffected by its presence. From the above equations it is clear that this requirement will only be satisfied if the connection weighting z_{nK} and z_{nL} are both zero and the errors on these weights are also zero. This is achievable in two parts. Firstly, the error on the upward weights must be a function of the weight value

$$\delta z_{ij} = f(z_{ij}) \quad \text{with} \quad f(0) = 0$$

Secondly, features observed to be inconsistent must be suppressed, using for example the resonance mechanism.

A simple resonance implementation involves assigning three variables to each node, an input I_i , an activity A_i and a template feedback value T_i . Input is made to layer i and used to calculate the change in input to layer j as specified previously. The activity and feedback for each node are then determined by the equations

$$T_i = P_j D_j z_{ji}$$

and

$$\text{if } T_i - 2\sigma_{max} < I_i < T_i + 2\sigma_{max}$$

$$\frac{\partial A_i}{\partial t} = k(I_i - A_i)$$

else

$$\frac{\partial A_i}{\partial t} = -kA_i$$

where σ_{max} is the maximum expected variance of I_i . In the final layer k there can obviously be no feedback mechanism so the equations are replaced by

$$\frac{\partial A_k}{\partial t} = k(I_k - A_k) \quad \text{always}$$

When the activity in the network has stabilised at some non-zero value the activities I and A are used to train the bi-directional weights using a simple extension to the learning algorithm already described.

$$\frac{\partial z_{ij}}{\partial t} = k_j^{-1} P_j(A_i - S_i)$$

$$\frac{\partial z_{ji}}{\partial t} = k_j^{-1} P_j(I_i - T_i)$$

Testing the network

A network schematic together with variable definitions can be found in [Figure 2]. There are several possible ways of organising the training of the layers. We proceed by presenting groups of patterns to the network, the first layer is trained on the individual patterns and then the second layer is trained on the accumulated response of the the first layer to the whole group.

The envisaged use of the network as a memory module is summarised in Figure 3. Each input pattern instigates the generation of a feedback template pattern for direct comparison with the input. A set of probabilities for the most probable node classes are also generated, providing access to any additional information which has been attached to these nodes during training. The information provided by the network can be used at any point during training, although performance will be poor initially and improve while the network stabilises. It should be remembered that this network is designed to work in an unsupervised environment. Supervision in the form of occasional corrective feedback should allow extensions to the training algorithm to improve performance, this is currently being investigated.

Planar object recognition.

The specific task of recognition of planar objects was chosen to demonstrate the working of this network architecture. Firstly an input representation needed to be developed which had a suitable degree of invariance. It was decided that the invariance properties required were invariance under rotation, translation and scale. By choosing pairwise relationships, for example the angle between any pair of lines, we obtain measures which are invariant under translation and rotation.

The representation that we have chosen involves creating a histogram of all of the orientations of lines within the input object relative to a reference line. Each entry is weighted with the product of the length of the two lines involved. There is one such histogram for each line segment in the object. However, not all of these patterns will be unique, in particular any segments of the same straight line will generate the same pattern differing only by a normalisation constant. Thus the representation is invariant under line breakage, but more importantly this means that construction of the object representation involves simply a loop over a polygonal approximation to the object [Figure 4]. Each histogram is convoluted with a gaussian, with a width given by the expected minimum orientation measurement accuracy, to give the pattern some degree of rotation invariance in the presence of measurement error. The square-root of the histogram values are used for input to the first layer in accordance with the normalisation procedure operating on stored patterns. The input to the next layer is constructed from the integrated output from the first layer as described previously, this output has the property that relative contributions from each input histogram are proportional to the length of the reference line so that again correct account is taken of the significance of the contribution of this feature in indexing the maximum node in the second layer. This method should work for any arbitrary curve as any curve can be approximated to any arbitrary precision with the relevant number of linear segments.

This object representation is ideally suited to the network we have described. The first layer is trained to learn to recognise the pairwise relationships between a straight section and all straight sections in the object, then the second layer is trained to recognise the pattern of activity generated by the first layer in response to the complete set of patterns for the object.

Unfortunately there is no easily computable model for the expected error on the derived histograms so a choice was made for the error model on the basis of scale invariance of the input pattern and a form which could be expected to perform well in the presence of noise. The preferred choice was for the expected error in the input to each layer to be zero and the error on the weights to be given by

$$\delta z_{ij} = \kappa z_{ij}$$

given the expected measurement error on the angle between any two features and the width of the convolving gaussian we find that $\kappa = 0.15$ provides a good limiting resolution for the network. This simple error model provides sufficient smearing of node classification to make multilayering possible, although the ability of the network to distinguish between two similar inputs is not optimum.

Results

The TINA system [Porrill et al. 1987] was used to generate 3d representations of five planar objects viewed at various positions and orientations in the visual field. Nine random views of these objects were taken and their 3d polygonal descriptions processed to generate pairwise histogram files for input to the network. The network was trained on 200 random selections from these files, by which time node generation was observed to have ceased with approximately 80 nodes in both layers. At each presentation the most probable node in the second layer of nodes was labelled with the object name. This is a simple method of learning supervision which does not affect the networks learning but makes it possible to test the performance of the network at any stage. An extension to this architecture which allows the computation of the conditional probability of classification for each object is currently being developed and will be the subject of a further paper.

The network was found capable of correctly identifying any subsequent views of the objects [Figure 5]. Overlap was observed between the representations for square and diamond (a square with one corner missing)[Figure 5(c) and (d)]. This behaviour is essential if the method is to be extended to deal with partial occlusion. The network was also presented with a pentagonal star [Figure 5(f)] which it classified as a pentagon, instead of generating a new node, due to the ambiguity of our pairwise histogram representation. The feedback and resonance algorithms worked as expected with spurious noisy features excluded from learning. The network was found to be flexible with the ability to generate new nodes when presented with unfamiliar patterns at any stage during learning. Subsequent performance of the network with familiar patterns was unaffected by the presence of the new nodes, as it needed to be for the network to be stable.

Discussion

We have developed a self generating self organising pattern recognition network which has a recognition acuity fixed by the level of expected noise on the patterns being classified and an amount of noise intrinsic to the networks performance. We now have to ask how this network might be generally useful in the field of object recognition.

An object recognition strategy can be specified in three parts;

1. obtaining information from sensors.
2. invariant-pattern construction from data obtained in 1).
3. learning the patterns from 2) using a pattern recognition strategy.

This network design falls cleanly into 3), in doing so it places some constraints on the form of the invariant data which can be generated by 2). In particular the Euclidean distance measure must be a reasonable indication of the similarity of the invariant patterns. For reasons of robustness these patterns must be relatively independent of both errors in data acquisition 1), and construction of the invariant representation 2), although reasonable errors can be allowed for as we have described.

Such networks as this are particularly susceptible to errors in data segmentation during the invariant pattern construction process. It might be argued that segmentation and recognition should be done in parallel as the segmentation process requires information from recognition to be made explicit. If this is the case then automatic segmentation needs to become a fundamental part of the network algorithms. Here the feedback from the network may make an iterative segmentation algorithm possible, this is being investigated.

We feel that this network is a step towards making context sensitive pattern recognition possible, as we have now a network which solves the major problems associated with making multi-layering possible. The specifications we set out to achieve, a self generating layered network, have been met. From the point of view of object recognition generally it is clear that work is needed in the generation of invariant representations. The limitations of these processes will have to be used to provide new specifications for the performance of future learning networks. One of the characteristics of this network is that the input of features to the first layer is order independent, much work is needed to extend the design for use in such fields as word recognition.

Conclusion

The function of popular nearest neighbour training algorithms has been examined and it has been argued that such classifiers are best suited to classifying patterns for which all components are uncorrelated and carry information

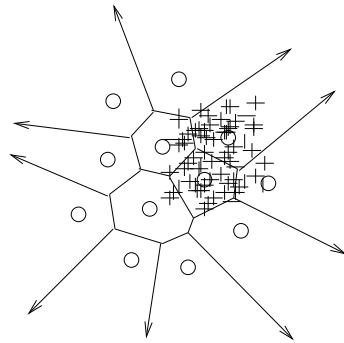
with equal significance. It has been stated that 'winner take all' algorithms represent decision processes which do not preserve information. It has been argued that for the capabilities of networks to advance past those of conventional clustering algorithms it will be necessary to develop multi-layered architectures which permit contextual influences during classification. The current trends in network design and training algorithms have been drawn upon to design a multi-layered network with feedback which attempts to preserve information by classifying patterns over groups of nodes.

The additional information provided by error models has been shown to provide an information preserving encoding that makes layering possible, and an analytic approximation has been developed to allow such networks to be simulated. These features allow the network to perform well in situations where the input pattern has associated measurement error. The simplifying assumptions made about the likely distributions of these errors may yet prove to be application specific. It has been argued that network flexibility requires automatic node generation and a new algorithm has been suggested which appears to perform satisfactorily. The resulting network requires serial data input and the resulting classification is independent of the order of data presentation.

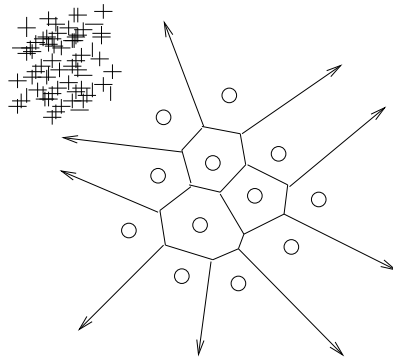
We can say that in some respects we have been successful in that a two layered network has been demonstrated to be capable of performing a specific vision task. However there is clearly more work to be done in developing suitable invariant representations for use as input patterns. Pattern classification networks can only learn to recognise the information they are given and do not have the ability to alter the representation of their input data. Networks can be made to learn invariant features but if suitable invariants are not made explicit in the input data then they will learn nothing of generic value. To make object recognition viable, networks will also need to be tailored to the demands of these invariant representations.

References

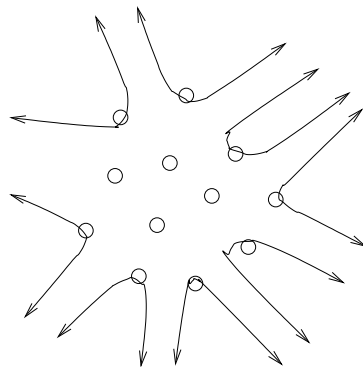
- Ashby. F. G. and Perrin. N. A. (1988). Towards a Unified Theory of Similarity and Recognition. *Psychological Review* , 95, No.1 ,124-150.
- Carpenter. G.A. and Grossberg. S. (1987). A Massively Parallel Architecture for a Self-Organising Neural Pattern Recognition Machine. *Computer Vision, Graphics, and Image Processing* , 37, 54-115.
- Kohonen. T. (1988). The "Neural" Phonetic Typewriter. *IEEE Computer* , (March), 11,22.
- Kohonen. T. (1984). *Self Organisation and Auto-Associative Memory*. New York: Springer-Verlag.
- Lippman. R. P. (1987). An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine* , (April), 4-22.
- Porrill. J. et al. (1987). Tina : A 3d vision system for pick and place. *Proceedings of the Alvey Vision Conference (AVC87)*, 65-72.
- Thacker. N. A. and Mayhew. J. E. W. (1988). Preliminary Studies in the use of Neural Networks for Pattern Classification. AIVRU Sheffield University, internal memo 35.



a/ Familiar patterns are encoded at many nodes

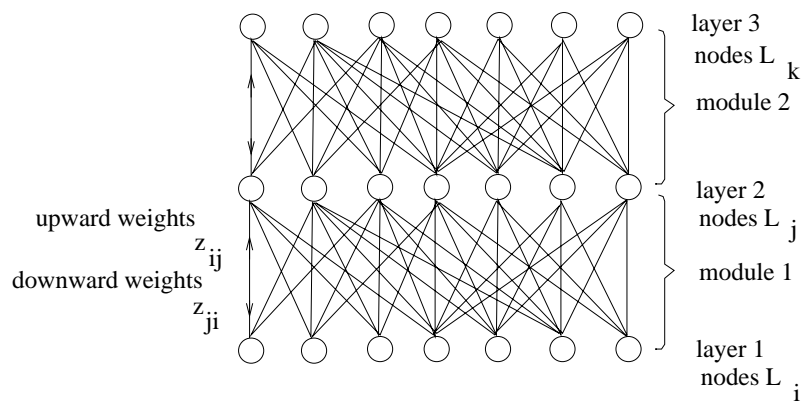


b/ Unfamiliar patterns are encoded at one node



c/ Boundaries defining input positions which would result in new node generation.

Figure 1: The figure shows the development of new node generation boundaries for the case where there is no error on the weights z_{ij} . The error on the input pattern is represented as a smearing of the input distribution.



node variables

I = input value (some function of the pattern to be classified)

A = current activity value determined by the action of STM equations

T = back-projected template of expected signal

S = reconstruction of noise free component of input signal

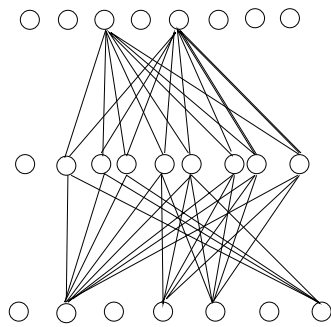
D = dot product between I and upward weights

P = probability that node would have the maximum D given error model

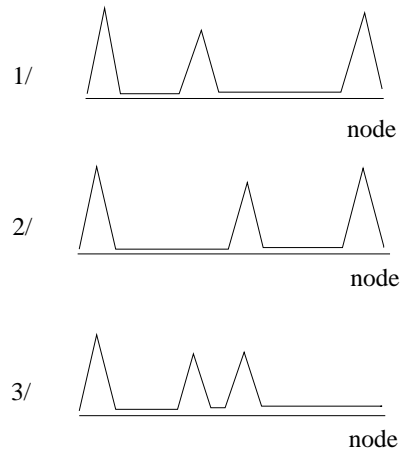
O = output value

Figure 2: Network architecture and relevant variables computed for each node.

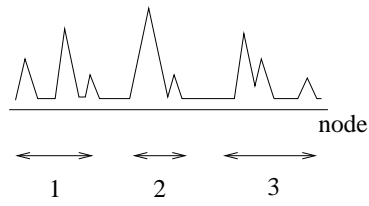
a) sparsely connected network



b) patterns are presented serially to the first layer.



c) composite pattern of output from the second layer used to construct input to the last layer.



d) final layer responds to the full set of patterns presented to the first layer.



Diagram 1

Figure 3: Envisaged use of a general purpose learning classification system.

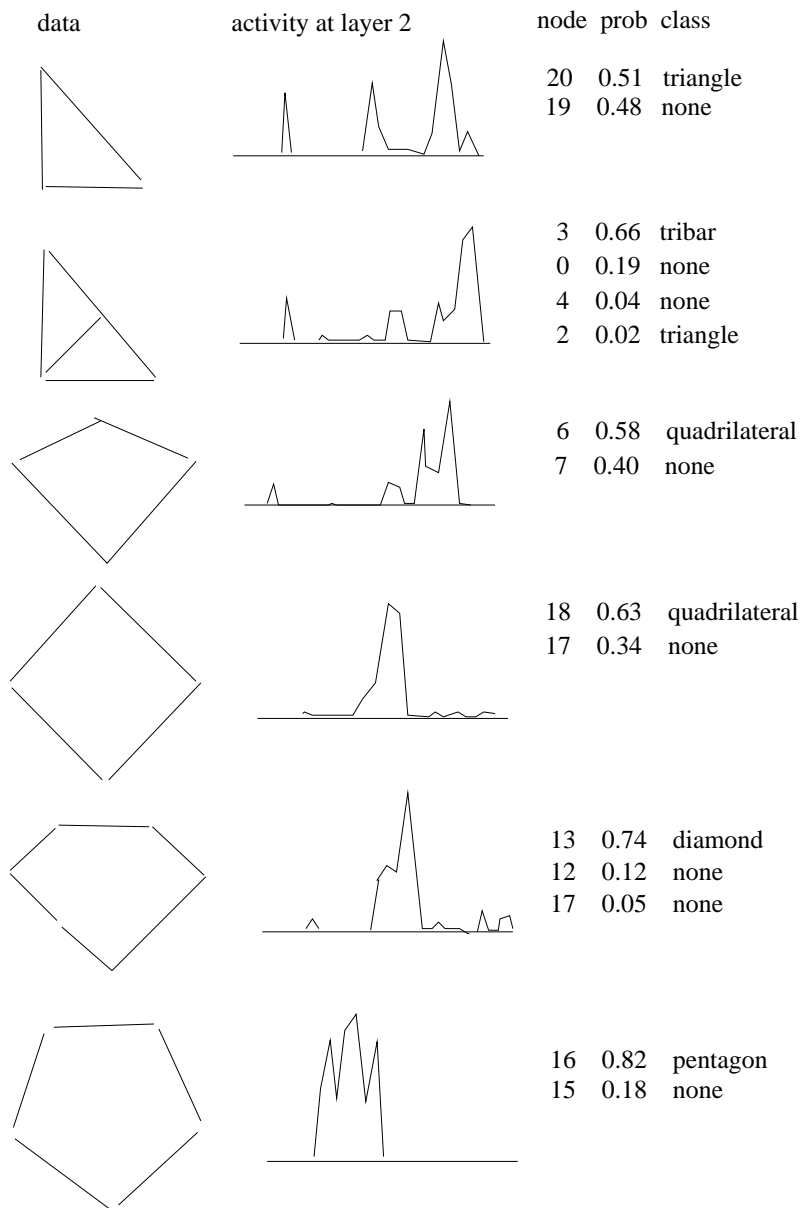


Figure 5

Figure 4: Invariant encoding of planar objects in a two layer classification network. Individual patterns corresponding to features are presented to the network for encoding at the first layer. The accumulated output response of the first layer is then used to provide input to the second layer for object classification. The three groups of peaks correspond to the response of the first layer to each of the three input histograms. The total activity of these groups is proportional to the length of each individual reference line.

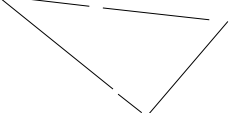
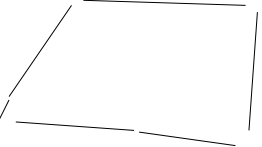
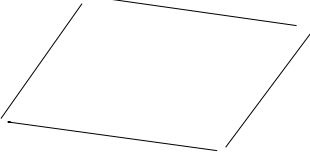
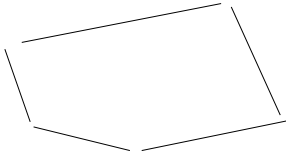
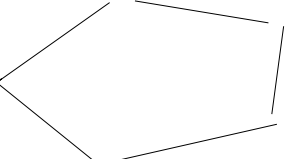
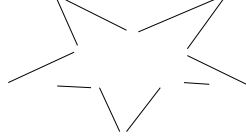
	orthographic view	node	probability	class
a)		21	0.65	triangle
		15	0.18	triangle
		10	0.06	triangle
		23	0.02	triangle
b)		65	0.65	quadrilateral
		54	0.33	quadrilateral
c)		4	0.38	square
		65	0.18	diamond
		25	0.08	square
		5	0.08	square
		49	0.08	diamond
d)		62	0.39	diamond
		61	0.39	diamond
		27	0.04	square
		57	0.04	square
		33	0.04	diamond
e)		47	0.26	pentagon
		38	0.73	pentagon
f)		45	0.27	pentagon
		3	0.14	pentagon
		55	0.13	pentagon
		38	0.10	pentagon
		47	0.09	pentagon
		31	0.09	pentagon

Figure 5: Test data and the probability responses produced at nodes in the second layer after 200 random presentations of the objects. The network copes well with problems of noise generally encountered in edge based representations of stereo data. Confusion between the classification of diamond and square objects (c) and (d) is due to the chosen resolution of the classification network. The pentagonal star (f) was not presented during training and is incorrectly classified as a pentagon due to the invariant representation chosen for input to the first layer (see text).