Guide to pattern recognition using random-access memories

I. Aleksander and T.J. Stonham

Indexing terms: *Pattern recognition, Random-access storage*

Abstract: About 12 years of work with a specific type of learning pattern-recognition system are reviewed. The principles and characteristics of the scheme, which is based on random-access-memory implementation, are discussed in some detail. Methods of improving performance and cost-optimising pattern recognisers are presented, together with case studies in a variety of fields including the recognition of alphanumerics, chemical data and faults in digital circuit boards.

1 Introduction

This paper contains no new material on the use of random-access memories (r.a.m.s) pattern recognition. It is designed to bring together and put forward in a simple way the fundamentals of the subject, which, to date, have been published in a scattered variety of articles and papers. The scheme is centered around r.a.m.s, but this is largely a conceptual crutch, becuase embodiments of resulting pattern recognisers might turn out to be software for conventional processors, specialised microprocessor systems or straight forward read-only memory (r.o.m.) programmable logic array (p.l.a.) hardware.

Implementation is not the central issue of this paper: the concept of learning, however, is. The first Section of the paper has been purposely divorced from any form of implementation to cover the pure rationale for using a learning system. R.A.M.s are introduced as simple learning machines in themselves with rather problematic characteristics which may be overcome by the use of simple networks of r.a.m.s. The effect of network parameters on pattern-recognition properties is salient to the subsequent Sections of the paper. Simple examples are used to illustrate the principles involved.

The latter part of this paper concentrates on a survey of optimisation methods and techniques that have been used in practice to improve performance and a set of case studies that show the sort of recognition performance that can be achieved using this type of system.

2 Learning pattern recogniser

Given a relatively modest matrix of 100 binary points (say, 10 x 10), the number of possible patterns that can be represented on it is large: 2^{100} or, roughly, 10^{33}.

A pattern recogniser is a device which when connected to this matrix will signal (say with a 1, as opposed to a 0) at some output the presence of a prespecified subset of the 2^{100} patterns. The problem arises from the fact that the specification for this subset could be very vague and insufficient for the design of a system tailored to the job. For example, such a specification may be the set of all patterns that look like an A. Or maybe, it could be the set of all versions of a prototype A distorted by up to 5% noise (i.e. with no more than 20 binary points altered).

The latter implies

$$\binom{100}{5} + \binom{100}{4} + \binom{100}{3} + \binom{100}{2} + \binom{100}{1} + 1$$

or approximately 79×10^6 possible patterns.

One could easily design a system to cater for the latter case by forming the bit-by-bit comparsion of an unknown pattern with the prototype, and, if more than 5 bits disagreed, a 0 would be output, with a 1 output for other cases.

However, it would be almost impossible to do this for the former case because not all distortions of A_s which still look like an A may be forseen by the designer.

It is for this reason that one appeals to learning systems, which generally operate in the following way.

Let U be the total (universal) set of all patterns that could occur on the binary matrix. Say that there are several sets of patterns that are to be distinguished. Let us label them $A, B, C \ldots$ The situation may be represented as a set diagram, as in Fig. 1.

A sample of each pattern class is shown* to the learning system, together with its class label. These samples are called training sets, say T_A, T_B, T_C for the example in Fig. 1.

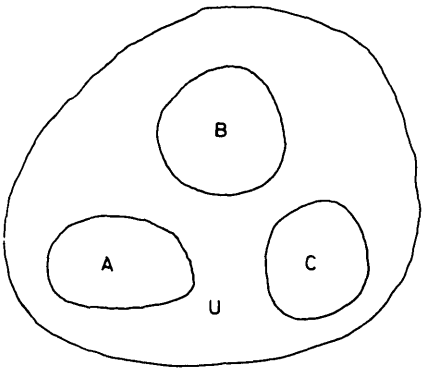


Fig. 1 *Pattern sets*

*This will later be called the process of training, and much of this paper will deal with the matter

Paper T283 C, first received 17th July and in revised form 23rd October 1978

Prof. Aleksander and Dr. Stonham are with the Department of Electrical Engineering & Electronics, Brunel University, Kingston Lane, Uxbridge, Middx. UB8 3PH, England

0140-1334/79/010029 + 12 $01·50/0

Much of this paper is concerned with the way in which the learning system interprets these training patterns and attempts to classify patterns other than those in the training set. The fact that a system classifies patterns other than those in the training set is called generalisation. The generalisation sets related to T_A, T_B and T_C are labelled G_A, G_B and G_C, a typical situation being shown in Fig. 2. Several points and definitions arise from this diagram:

(*a*) First, one assumes that only the patterns in A, B and C will require classification. The rest of the space in U represents meaningless spurious patterns. Then, we see that (with the given generalisation) all the patterns in B will be identified as belonging to class B, and the recognition of this class is perfect. This too applies to all patterns in A.

(*b*) The area of G_B which lies outside B merely indicates that some spurious patterns which in fact do not belong to B would be classed as Bs. This does not matter, however, because we have said above that only patterns within A, B and C are used as input to the system.

(*c*) The fact that G_A encroaches into area C (subset K) indicates that those patterns in C which fall in the K area will be classified incorrectly as As; this is called a misclassification.

(*d*) The fact that there are patterns in C which fall outside G_C and in no other classification area indicates that the system will treat them as unclassifiable; this is called a rejection.

(*e*) In general, a rejection is thought of more favourably than a misclassification. It is equivalent to the system saying 'I don't know' rather than making the wrong decision. In the recognition of postcodes, for example, a rejected envelope could be salvaged and classified by a human, whereas a misclassification could cause it to be sent to the wrong address.

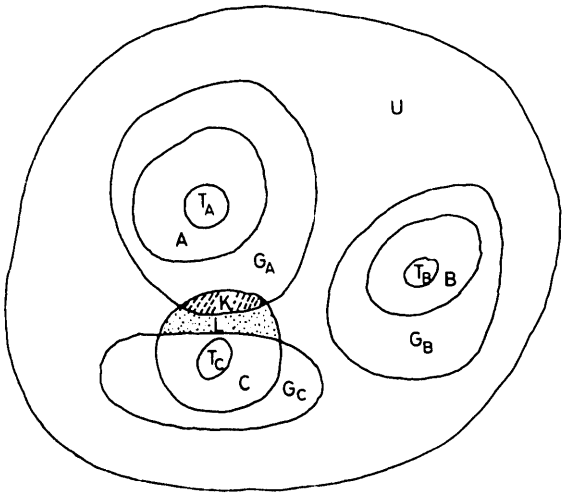


Fig. 2 *Generalisation sets and errors*

3 Random-access memory as a learning pattern recogniser

The essential elements of a ramdom-access-memory are shown in Fig. 3 (this is not a complete circuit diagram, it is just intended to aid the description that follows).

This operates as follows. There are 2^n possible distinct address patterns as the n address terminals. Each such pattern selects (addresses) a two-state (0/1) device (flip-flop), causing the value of its state to appear at the data-out terminal. The above is known as the read or use mode.

Another mode, the write or teach mode, is entered by energising the write-enable terminal. In this mode, the contents of the addressed flip-flop may be charged to a value determined by the logical value of the data-in terminal. That is, if the data-in terminal is at 0 when the write-enable terminal is energised, whichever flip-flop is addressed by the input at the time will enter state 0, whatever the value of its previous state.

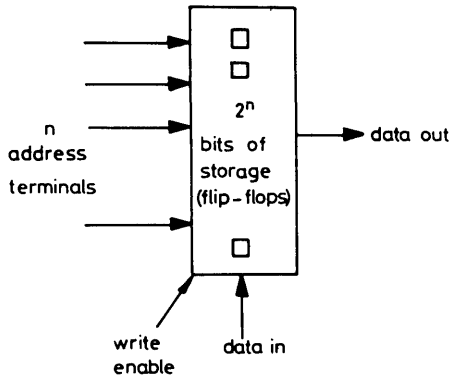


Fig. 3 *Random-access memory*

A r.a.m. alone can act as a rather trivial pattern recogniser in the following way. Assume that the pattern is applied at the n binary inputs. The recogniser will recognise one class of pattern by outputting a 1 when a member of this class is present at the input.

Assume that all the flip-flops are initially at zero. That is, no pattern is identified. Say, now, that all the all-1s pattern and the n patterns with only one 0 are taught to the r.a.m. by presenting them in turn to the address terminals with the write-enable wire energised and a 1 at the data-input terminal. During a subsequent 'use' phase, the r.a.m. will recognise the $n + 1$ patterns it was taught during the previous 'teach' phase, but no others. Hence, it acts as a learning pattern recogniser, but one with no generalisation. It is because of this latter point that the scheme was labelled as being 'trivial'. It will now be shown that very simple networks of r.a.m.s are not trivial in the same way.

4 Some simple networks and their generalisation properties

In Fig. 4*a*, a simple arrangement is shown where a 3 x 3 binary matrix is connected to three r.a.m.s. The 'data-in' terminals of the r.a.m.s are connected together and an AND gate receives the 'data-output' terminals. In this way, the r.a.m.s are taught to respond with 1s for the patterns in the training set, and only those patterns causing all three r.a.m.s to output 1s would be classified in the same way as the training set.

It is assumed that all stores are at 0 before training commences. A specific training set T_A is shown in Fig. 4*b*, and it is represented as a subset of the universal set U in Fig. 4*c*. Note that U contains $2^{3\times3} = 512$ patterns. The generalisation set G_A may be computed as follows.

Looking at the way the r.a.m.s are connected, one notes that during the entire training session each r.a.m. 'sees' precisely two addressing subpatterns. The set G_A is then clearly made up of all combinations of such addressing subpatterns. In general, the size of G_A (denoted by G_A) is given by

$$G_A = k_1 \times k_2 \times k_3 \ldots k_e$$
$$= \prod_e k_j$$

where k_j is the number of patterns seen during training by jth r.a.m. In this example

$$G_A = 2 \times 2 \times 2 = 8$$

Subtracting from this number those patterns in T_A, G_A must contain another 5 combinations of 'seen' subpatterns. These are shown in Fig. 4*d*, the set diagram being shown in Fig. 4*e*.

Consider now what would happen if the AND gate were replaced by an OR gate, the rest of the system remaining exactly the same. In this case, a pattern will be classified in G_A as long as one of the r.a.m.s sees an input seen during training. One can show that G_A is now much larger and, in fact, contains 296 patterns. This large number includes patterns as diverse as those shown in Fig. 5*a*. Clearly, one gets a feel for the fact that the more restricted decision at the output (AND) causes G_A to include patterns that are more directly related, that is, similar to the training set.

There is another way of affecting G_A by means of network changes. This is shown in Fig. 5*b*. It is assumed that the training set is the same as in Fig. 4*b*. This time, however, we note that the first and third r.a.m.s see two subpatterns, whereas the second sees only one. Hence, the total number of patterns in G_A is

$$G_A = 2 \times 2 \times 1 = 4$$

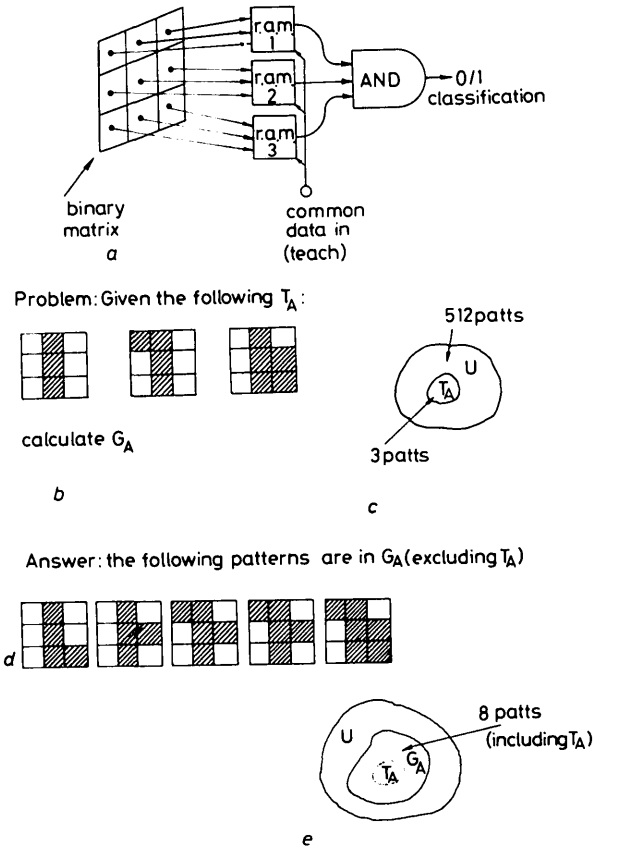


Fig. 4 *Example of training and generalisation in a simple single-layer net*

a 3 X 3 binary matrix
b Specific training set
c Subset of *U*
d 'Seen' subpatterns
e Set diagram

Hence, outside the training set there is just one pattern in G_A, and this is shown in Fig. 5*c*. This connection has, in fact, picked out a very important common feature of the training set: the centre vertical bar.

The networks discussed here have a particular name: single-layer nets, or s.l.n.s. They always have the following physical characteristics:

(i) Given a binary matrix of R points, an s.l.n. contains R/n n-input r.a.m.s connected (often at random) so that each r.a.m. input (i.e. address terminal) 'sees' one, and only one, matrix point.

(ii) A simple decision such as AND, OR, MAJORITY or GREATER-THAN-X, is made at the output of the r.a.m.s.

From the examples in this Section of this paper one learns that:

(*a*) Generalisation is affected first by the diversity of the patterns in the training set; that is, the more diverse the patterns in the training set, the greater will be the number of subpatterns seen by each r.a.m., and hence, the greater will be the generalisation set.

(*b*) Secondly, the output decision, based mainly on the value of a threshold placed on the number of r.a.m.s that respond with a 1, strongly affects the size of the generalisation set. The smallest set is achieved with an AND decision (i.e. threshold between a R/n and $R/n - 1$), and the greatest with an OR decision (i.e. a threshold between 0 and 1).

(*c*) Thirdly, the connection of r.a.m.s to common features in the training set reduces the generalisation set.

5 Multicategory pattern recognition

A single-layer net tends to divide a universal set into two categories: those patterns in G_A and those that are not.

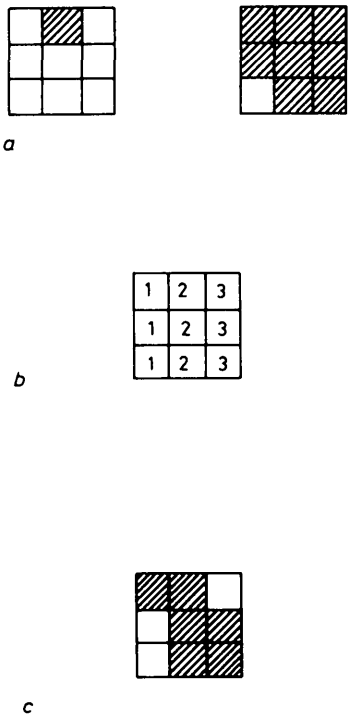


Fig. 5 *Generalisation for an OR gate at output of net in Fig. 4*

a Examples
b Reconnection pattern; this differs from Fig. 4*a* in the sense that 1s are connected to r.a.m. 1 etc.
c Reduced generalisation set

Clearly, most pattern-recognition problems would require classification into more categories (for example 26 for the letters of the alphabet). That is, one needs to approach the situation shown in Figs. 1 and 2. Particularly important is the 'don't-know' category.

This is achieved by using several s.l.n.s. In theory, one could use $\log_2 C$ s.l.n.s for a C-class problem. For example, two nets (Fig. 6*a*) could be used to recognise, say, three classes A, B, C, as shown in Fig. 6*b*. This requires that for training set T_A, only s.l.n. α is taught to output a 1, for T_B both α and β are taught to output a 1, while for T_C only β is taught to output a 1. When neither net responds, this is taken to be a 'reject' or 'don't-know' response.

There is a major difficulty with this kind of scheme: the nets have to be trained to respond in the same way to patterns that are clearly in different classes. That is, it is likely that the patterns on which the s.l.n.s are trained have a great diversity, leading to an inappropriate generalisation.

Consider the example in which A is the set of three vertical patterns, of which Fig. 6*c* is an example, B is the set of three horizontal patterns of which Fig. 6*d* is an example and C is the set of three diagonal patterns (or negative slope) shown in Fig. 6*e*. If the nets are connected as in Fig. 4, it may be seen that if α is trained on A and B and β on B and C, all the patterns in A and C will give a B response.

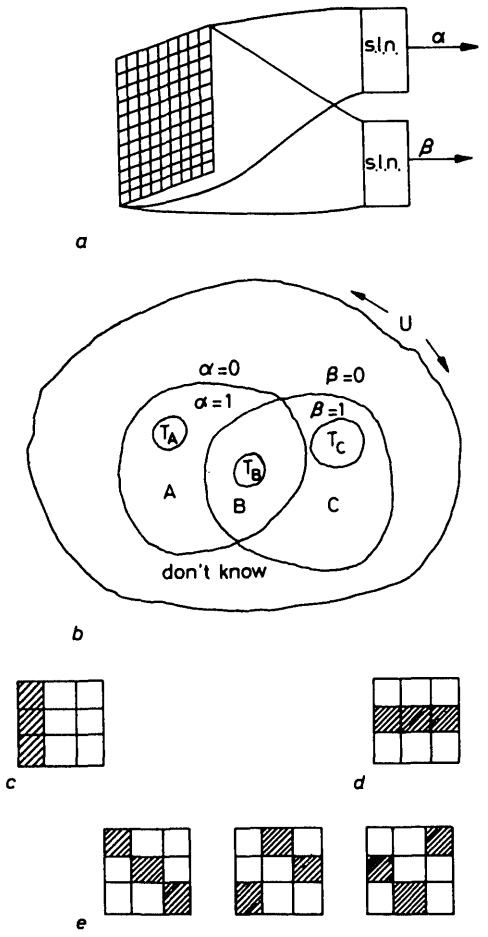


Fig. 6 *Multinet systems*

a Two nets for a 2-class problem
b Coding of net output
c Example of vertical pattern
d Example of horizontal pattern
e Diagonal pattern

One imagines that the problem could be solved by altering the conditions and the training. For example, one could let β be connected as in Fig. 5 and the horizontals be A, the verticals C with the diagonal as B. The intent here is to tailor the matrix connections to the features of the patterns. However, the scheme does not work because one notes that, because the diagonals are used to teach both nets, α cannot distinguish between diagonals and verticals, whereas β cannot distinguish between diagonals and horizontals. Hence, any pattern will give the B response.

A more usual way of dealing with the multicategory problem is to assign one modified s.l.n. per class. The modification consists in not having an output decision circuit, but leaving the output decision to an overall circuit called a maximum-response detector. The system is shown in Fig. 7*a* for a 3-class problem. The maximum-response detector assigns the classification to that discriminator which has the strongest response. The resulting generalisation sets are shown in the diagram of Fig. 7*b*. All multiple responses (in the sense that two or more s.l.n.s respond the same way) are classed as 'don't know'. What is left of the individual s.l.n.s is now called a discriminator.

The major advantage of this scheme is that a discriminator will be trained as a set of patterns which are in some sense similar. Also, one can tailor the discriminator connections to suit an individual class of pattern. Returning to the example of the horizontal, vertical and diagonal lines, one would require three discriminators. Say that the first H is to distinguish horizontals and has its r.a.m.s horizontally connected, the second V has the r.a.m.s. vertically connected, and the third D has them diagonally connected (as in Fig. 6*e*). For each step in the training procedure (assuming training on all nine patterns), only the appropriate discriminator is taught. Consequently, for that discriminator, one r.a.m. will be taught to output a 1 for the all-1 address, and the other two output a 1 for the all 0 address. Consequently, when any pattern in a training set is presented to the system, all three r.a.m.s will fire in the appropriate discriminator, and none in the other two. This provides perfect recognition of the training set. Now, consider a pattern which is a distortion of a vertical training pattern, say

0 1 0
1 1 0
0 1 0

Clearly, two r.a.m.s in the V discriminator will fire and none in the others, still giving the correct decision. Indeed, all single-bit distortions of the training sets will be correctly classified.

A further point of interest in this kind of system of discriminators is that one can obtain control over the generalisation by insisting that there will be a minimum difference (threshold) between the responses of the two maximum discriminators, for the pattern not to be rejected. Figs. 7*c* and *d* show the resulting generalisation sets for the same situation as in Fig. 7*b*, except that in 7*c* the threshold is 2 and in 7*d* 3. (Normally the threshold is 1 as in Fig. 6*b*.

In summary, therefore, the C-discriminator system for a C-class problem has the following advantages:

(*a*) The discriminators are not required to accept a greater diversity of patterns than contained in a single class. This avoids exaggerated generalisation.

(*b*) Connections can, in certain cases, be tailored to

suit individual classes.

(*c*) Predetermination of output-decision thresholds (fixed-output decision) is avoided.

(*d*) Differential thresholds between discriminators may be used to control the 'reject' class size.

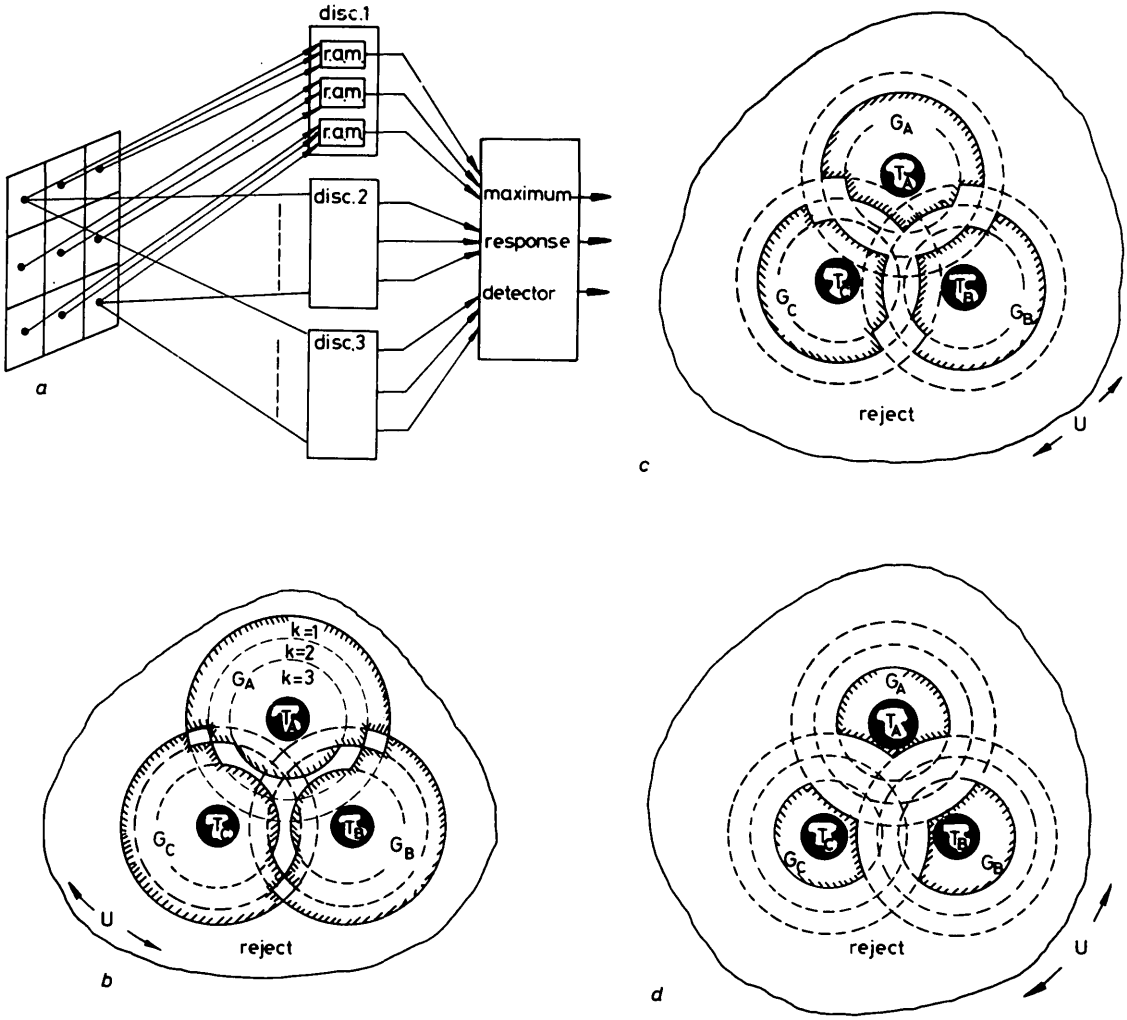


Fig. 7 *Multicategory problem*

a 3-category system
b Strongest-response discrimination
c Threshold of 2
d Threshold of 3

6 Optimisation of s.l.n. classifiers

6.1 Introduction

To suppress the extensive and sometimes inappropriate generalisation which occurs when the C-class multicategory problem is implemented with the minimum number ($\log_2 C$) of s.l.n.s practical systems invariably comprise C discriminators for the C-class problem. The designer is therefore able to optimise each discriminator to respond in the appropriate manner to only one data category. Nevertheless, the intrinsic flexibility of the technique is so great as to preclude exhaustive assessment of all system parameters and a heuristic approach must, in some cases, be adopted.

In this Section, the optimisable parameters of the system are considered, and techniques for improving the overall recognition are discussed.

6.2 Training data

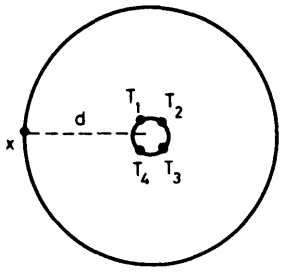
The training data on which the discriminators are set up determines the generalisation properties of the system, as has been shown in previous Sections. The response to an unknown pattern can be assessed[1] by measuring its Hamming distance from all the members of the training set, and hence its most probable classification category can be obtained. In practical pattern recognition, the nature of the training data in terms of Hamming distance is of paramount importance to the overall performance.[2] Fig. 8*a* shows the area of response of patterns to a set trained on set T, where

$$T = \{T_1, T_2, T_3, T_4, \ldots, T_n\}$$

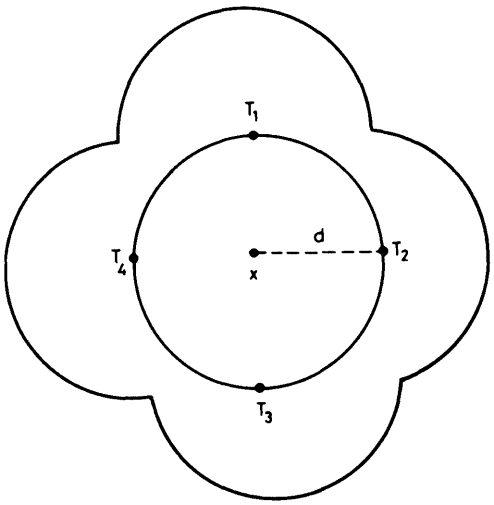
in which unknown patterns will obtain an equal or higher response than pattern X at Hamming distance d from T. In this case, the training set T is well defined in terms of recurrent pattern features and the Hamming distances between training patterns is low.

The same test/training pattern Hamming-distance relationship can prevail for a less-well-defined pattern class

as shown in Fig. 8*b*. *T* now represents a much more diverse class of patterns, although the mean Hamming distance from *X* still remains at *d*. It can be seen that the comparable area of response is now much larger; i.e. the discriminator has a much larger generalisation area which may induce high responses from patterns which should not be associated with *T*.



a



b

Fig. 8 *Hamming distance*

a Generalisation area of a tightly clustered training set *T* at a distance *d* from test pattern *X*
b Generalisation area of a diverse training set at a distance *d* from test pattern *X*

6.2.1 Optimum training set: An exploration of the Hamming-distance relationships between training patterns will determine the action required to optimise the training sets. In the case where the patterns cluster, a procedure that ensures that the training set is representative can be used. An s.l.n. is trained on an arbitrary member of a class of patterns (a reference set), and the responses of all members obtained. The pattern with the lowest response is detected and incorporated into the training set, and the system retrained. By repeating the procedure, a reliable training set can be compiled.

This procedure serves two purposes:

(*a*) The training set thus obtained is fully representative of the reference set and, hence, the data category as a whole

(*b*) The response behaviour can provide evidence of recurrent *n*-tuple features and, hence, a pattern in the data.

The latter can be ascertained by observing the lowest response after each training increment. If, after training on *n* patterns (less the total number of patterns in the reference set), the lowest response is in fact the maximum possible from the discriminator, further training will not contribute any additional information on the pattern class. The patterns remaining in the reference set and not included in the training set are being recognised by generalisation, as shown diagrammatically in Fig. 9.

The tightness of the clustering at the training patterns can be obtained by observing the rate at which bits within the discriminator are being set, and full specification of the reference set occurs when this rate becomes zero.

This technique has been successfully employed on patterns derived from mass spectra to ascertain the existence of recurrent features in the data set.[3]

6.2.2 Subcategories: The diversity of patterns as typified by *T* in Fig. 8*b* is unlikely to occur in real data, which can be recognised by the human interpreter, but distinct subcategories can often be detected within a class (e.g. upper- and lower-case versions of a letter). The assignment of separate discriminations to each subcategory can quite often reduce the overall errors by restricting the generalisation between the subcategories within a single net. The populations of the subcategories must be taken into account, and it may be economic to ignore low-populated subcategories. This can be illustrated by considering the separation of printed characters and, in particular, *B*s and 8s. Fig. 10 shows the generalisation area for training sets

$$8 = (8_1, 8_2, 8_3, 8_4 \ldots)$$

$$B = (B_1, B_2, B_3, B_4)$$

The inclusion of the character ⊟, as obtained from a 7-segment display, would create a problem. It can represent an 8 or *B* in the training sets for either character, and would therefore destroy a large proportion of the discrimination between those alphanumerics. The problem can be dealt with either by ignoring the recalcitrant character with respect to training (clearly it would most probably be rejected during classification) or by creating a discriminator to detect it together with its close variants. The latter course of action would preserve dichotomy

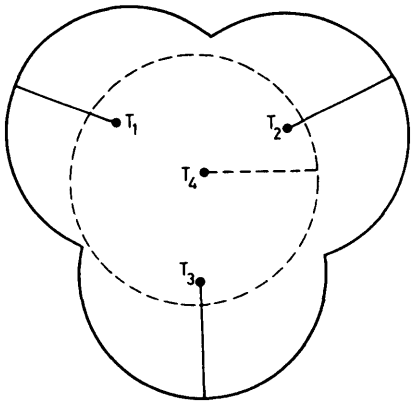


Fig. 9 *Generalisation completely specified by T_1, T_2 and T_3. Pattern T_4 does not contribute any further information*

between 8s and Bs, although 8/B and B/B separations would not be high.

6.2.3 Protection against errors in training data: A consideration of rogue training patterns has culminated in a method whereby the recognition performance of a classification can be protected against corruption arising from errors in the training data. By an extension of the example in Fig. 10, the inclusion (by accident or otherwise) of a member of 8 in B, would cause the separability of character 8 and B to be almost totally destroyed. One can overcome this by the use of a memory element threshold[4] t, whereby the pattern n-tuples sampled by the memory elements within a discriminator must occur t times over the complete training data before the location is set. A threshold of 1 would provide protection against one error pattern by suppressing all the n-tuple samples over and above those occurring within the valid training data. This prevents the area of generalisation encompassing another category.

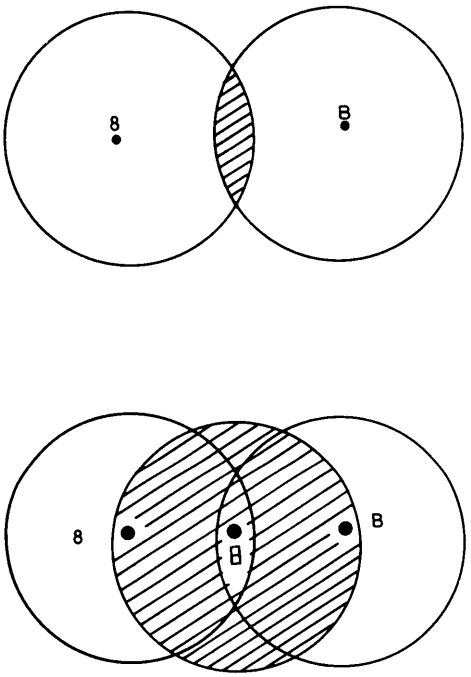


Fig. 10 *Error area is increased by the inclusion of pattern* B *in either or both training set*

Error area is shown by shading

6.2.4 Size of n-tuple samples: The size of n-tuple samples affects the behaviour of an s.l.n. system as does the number of training patterns in applications where an optimum training set cannot be specified. (This is generally the case in optical pattern recognition where noise is present). It has been found empirically that for a given size of training set there is an optimum value for the size of the pattern samples which will give maximum performance; smaller samples causing overgeneralisation and larger samples undergeneralisation. This behaviour is also manifest if the sampling size is kept constant, and the number of patterns in the training sets is varied (small training sets having an effect similar to small n-tuple samples). The extent to which optimisation of the n-tuple size is both desirable and cost effective depends again on the nature of the data to be classified, and is proportional to its diversity. N-tuple behaviour is summerised in Fig. 11 for recognition of hand-written characters (a diverse data form) and machine printed characters (a relatively constrained data form). By combining these performances with the cost characteristics in terms of bits of storage against n for a 1-to-1 mapping (Fig. 12), the return on n-tuple size optimisation can be assessed.

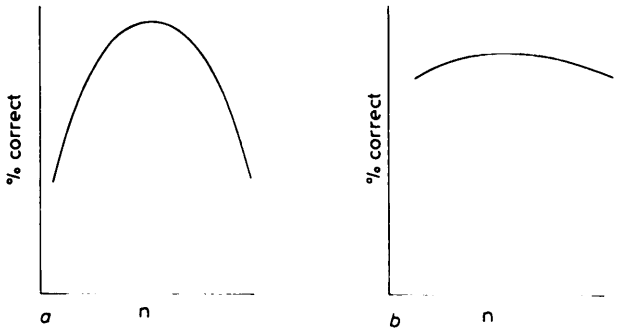


Fig. 11 *Performance behaviour with varying n-tuple size*

a Handwriting
b Machine-printed characters

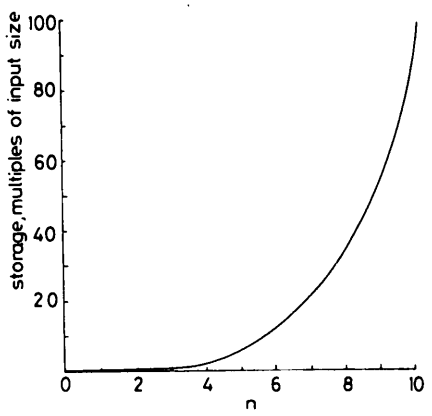


Fig. 12 *Cost with increasing n in terms of storage*

6.2.5 Input connection mapping: The input mapping determines the sampling and defines the locations of the matrix. These are combined to form a subpattern which is subsequently interpreted as an address within the s.l.n.s. There is a vast number of possible connections available (for a 16 x 16-bit pattern matrix the number of 256! or approximately. 10^{1000}). An exhaustive assessment is therefore impossible, and a 'hill-climbing' approach must be adopted, although this gives no guarantee that the final mapping is the most suited. The classification performance as a function of the mapping approximates to a normal distribution, with the majority of mappings giving average performance. There is a small but finite probability of a specific map giving a very much better (or worse) performance, but its detection is governed by chance. Fig. 13 gives experimental results of a classification distribution pertaining to medical data, for a range of maps.

In an unoptimised problem, a random map is chosen, because sampling points distributed throughout the pattern matrix are more likely to detect global features than an ordered map, which in a single-layer system is only sensitive to local features.

In optimising a mapping, an initial check would be carried out on a set of maps, perhaps up to 10, to ensure that an unduly poor map had not been selected. The best map would then be selected and the connections modified according to the outcome of the following investigations:

(a) Removal of invariant points on input matrix

By observing the training data sets, any points on the input patterns which are invariant over the whole training set, and are always either 1 or always 0, can be disconnected.[6]

An invariant point in a subpattern reduces the effective storage of its associated memory element by half (2^n to 2^{n-1}). If the invariance within the training data is typical of the data as a whole, this area of store is never addressed, and if information occurs in the hitherto invariant location during operation, all corresponding elements in each discriminator have identical outputs. Consequently, no classification information is forthcoming from the pattern point and its reconnection cannot degrade the performance and will probably contribute in a positive manner to the classification.

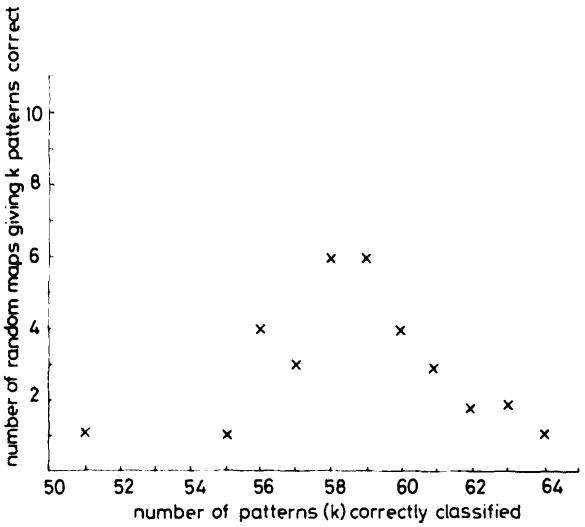


Fig. 13 *Distribution of responses with 36 differing maps*
Medical data

(b) Common n-tuple samples

An extension of the invariant-point problem in the training data can result in common n-tuple samples throughout the training data. This gives rise to common memory-element contents within corresponding elements in different discriminators, the outputs of which will always, therefore, be identical, and no contribution to the classification of the data will be made by those elements.

The probability of common n-tuples occurring throughout a complete training set is low; therefore, a major improvement for all classes in a multicategory classification is unlikely. If, however, reference is made to actual classification results, this method can be used to improve the classification between categories of low separability by removing common n-tuples between them.

It is possible (although unlikely) that the margin between the responses of highly separable groups will be reduced, although the classification will not be corrupted. The overall effect is therefore that the classification and confidence is improved throughout the data set as a whole.

(c) Feature masking

Feature masking within the elements of a discriminator is comparable at the subpatterns (n-tuple) level to the effect of rogue patterns in training sets at the pattern level. It occurs when the same subpattern presents itself in the training set of two or more classes.

The reduction of feature masking does not involve any direct modification to the connection mapping, but it can increase the usefulness of subpattern samples by the judicial modification of the store contents after training. Information on the frequency of occurrence of subpatterns is required; therefore, the optimisation is more suited to software systems in the first instance.

The technique can best be described by reference to an example. Let corresponding memory elements in different discriminators sample a set of subpatterns during training such that

$$A = (a_1 a_2 a_3 \ldots a_n) = B = (b_1 b_2 b_3 \ldots b_n)$$

It is probable that some of the addresses will be the same, e.g.

$$a_1 = b_1 \quad \text{and} \quad a_3 = b_3$$

Masking degrades the performance significantly if the frequencies of occurrence of a and b differ significantly, implying that the sample is a strong feature of one class and a weak feature of the other. As the memory elements are not sensitive to the frequency of occurrence of the subpatterns, over and above the initial occurrences, the corresponding elements in the discriminators are unable to differentiate, and have identical outputs. By locating the memory element in the discriminator for which the feature is weak and resetting the appropriate bit after training, the strong feature can contribute to the correct classification of these patterns at the expense of enhancing an incorrect response to patterns of other categories where the feature is weak. This approach involves a trade off, but it can be utilised to increase the confidence of decisions between data categories of low separability.

6.3 Coding of physical data

Most pattern recognisers have to be interfaced with the real world, and the method of transduction and coding influence the overall performance, In optical pattern recognition, a suitable threshold must be chosen and applied to the grey scale to detect image boundaries.

The problem of coding numerical data is more critical. The discriminators require binary information, and a simple binary coding of a numerical value might seem appropriate. The code is interpreted by the s.l.n.s as a pattern, and from the knowledge of their Hamming-distance behaviour, it is imperative that the coding reflects similarities in measurements. A direct binary code has discontinuities in its Hamming distances between incremented numbers, thereby creating disproportionate changes in the pattern. A reflexive Gray code eliminates this specific problem, but widely differing quantities can now be close in Hamming distances. The most suitable coding is a 1-in-n code, which overcomes all the Hamming-distance problems. It is, however, very inefficient and can require a large amount of input space.

The properties of these codes are summarised in Fig. 14.

Binary and Gray codes will distort the generalisation characteristics of an s.l.n. with respect to properties of the data in the physical world.

Another approach to coding has been to measure binary properties of the data to be recognised, and this appears to be a suitable technique in some applications. In mass-spectral recognition, the binary patterns represent the

presence or absence of peaks pertaining to specific masses, and a similar approach could be made to the encoding of sound according to the presence or absence of frequency components.

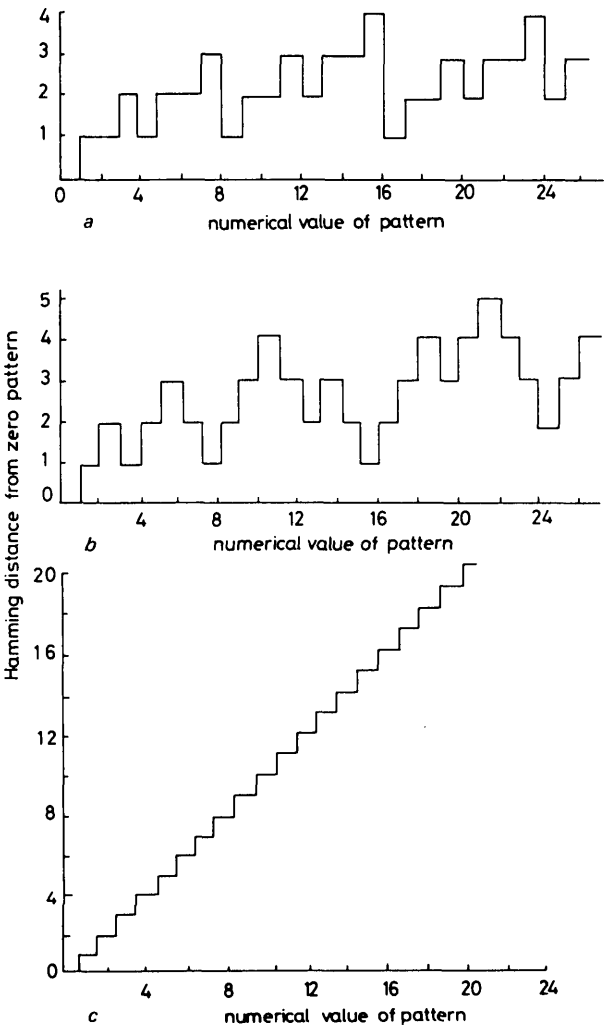


Fig. 14 *Hamming distance properties of codes*

a Binary code
b Gray code
c 1-in-1 code

6.4 *Data transformations and multilevel networks*

Nonlinearities in the coding of physical data can distort the area of generalisation of a training set. One would normally expect similar patterns in the real world to be transformed into binary patterns closely related in terms of Hamming distance. However, advantage can be taken of distorted areas of generalisation to improve the classification. For example, the Hamming distance between acceptable binary versions of the characters *C* and *O* can be very low, and the generalisation areas of these characters overlap. If high confidence requirements are imposed on the classifier, a large number of *C*s and *O*s can be rejected which would otherwise be misclassified. A second layer processor can then be employed, wherein the mapping only considers the potential areas of difference between the characters, i.e. the right-hand side of the pattern matrix. The relative separations in Hamming distance are thereby increased, and there is a greater probability of correct classification at the output of the second layer. This technique, whereby high levels of rejection are induced (to reduce the output error rate to a minimum)[7] in the first layer and nonlinear transforms used to enable a classification to be made in the second layer, is termed 'buck-passing'.[8] Note that the transform must be nonlinear. A linear transformation is tantamount to using a different random connection mapping, and it is then most unlikely that the accuracy of the classification will be improved.

Buck-passing can be extended to the point whereby only a coarse classification is made in the first layer. In alphanumeric recognition examples of the categories could be:

(*a*) round characters	*C O G Q*
(*b*) double-round characters	8 *B S S*
(*c*) vertical centre	1 *I T J*
(*d*) vertical edges	*H M W N*
(*e*) – – – – –	– – – –

and a discriminator detects each of these groups of characters. The final classification can then be made in a second layer, where optimisation can be more extensive owing to the limited number of possible decisions determined by the first layer.

7 Case studies

7.1 *Introduction*

In this Section, a range of pattern-recognition problems to which s.l.n. classifiers can be applied will be examined and referenced. Two basic approaches can be adopted towards a specific data set. One can be termed recognition, in which the aim is to assist or replace a human operator who processes a data form where well defined patterns or characteristics exist. The other approach is pattern detection, where meaningful partition of a data set is sought through the use of a classifier trained on a representative subset of data. If partition accuracy exceeds that expected by chance allocation of categories, the existence of some characteristic information can be postulated and pattern recognition processing may be appropriate. A salient feature of this approach is that the processing is self-evolving and does not depend on existing theories and prejudices associated with the interpretation of the data, and therefore the opportunities of establishing new cause/effect relationships and correlations between measurements and physical/structural descriptions exist. The results quoted in what follows are those that were generally obtained after optimisation of structure and training sets. Details may be obtained from the References. Recognition results never include testing or training sets.

7.2 *Alphanumeric character recognition*

One of the original pattern-recognition problems, alphanumeric recognition performance figures with an *n*-tuple recogniser system, were first published by Bledsoe and Browning in 1959[9] using a software simulation. 80% correct classification was achieved on hand-printed characters. Subsequent *n*-tuple experiments by Ullmann[5] and Cheung[10] on similar data have produced accuracies up to 90%.

Machine-printed characters provide a more constrained data set, although the problem is nontrivial owing to the diversity of fonts and styles and the presence of noise.

An initial performance of 93% correct classification has been reported in an unoptimised single layer classification over a data set of 10000 examples from 34 character categories, and this rises to 97% after optimisation.[11] The

introduction of multilayer techniques gives a further increase to 98·2%,[4] and limited experiments to improve the second-layer discrimination show that 99% should be possible.

Fig. 15 illustrates the diversity of data encountered in the *O* class in these experiments.

7.3 Chemical-pattern recognition

Single-layer network systems were applied to the recognition of mass spectra in the early 1970s. Mass spectra are well defined data, because the presence or absence of molecular fragments can be ascertained to a high degree of accuracy. The motivation for the work was based on the ability of the experienced human spectroscopist to recognise spectra on inspection. Emulation of this behaviour within a machine was attempted. Ideally, a unique classification is required for each spectrum, but in practice a general classification according to the compound's functional group was made (e.g. 1^{ry} alcohols, aldehydes, diesters etc.). In all, up to 1000 spectra falling into 42 general classes were assessed.[12, 13] The data represented the most frequently occurring compounds in a chemical manufacturing plant, and it was found that the data could be partitioned into the 42 categories with an accuracy greater than 99%, which gave a larger saving in cost over direct storage at a time when typical prices were £1 per bit. The dramatic decrease in costs over the past eight years, together with the ability to obtain a unique identification, make direct storage more attractive.

Fig. 15 *Examples of digitised machine-printed characters*

Recent research,[14] however, has revealed a powerful facility for the interpretation of spectra of mixtures of compounds, where a data explosion in terms of possible spectra precludes direct storage. (There are 500 000 possible mixtures of any two spectra from the original base of 1000). Furthermore, the problem is more closely related to the practical environment, in which spectra of pure samples rarely occur due to contamination, and their use in manufacturing processes.

To understand the problem fully, the mass spectrum of a mixture must be considered. The mass spectrum of a pure compound is normalised such that the largest peak has an intensity of 100. The fragmentations of molecules occur in a short period of time (10^{-10} s) within a mass spectrometer, and there is no recombination of radicles. Therefore, a spectrum of a mixture can be regarded as the superposition of the spectra of the compounds. This is valid, however, only when the largest peaks in the individual spectra are equal. If the relative concentrations of components are such that this condition is not fulfilled, the level of significance of one or other of the spectra is reduced within the composite spectrum.

A binary pattern derived from the measure of the presence or absence of a peak above a threshold intensity at each integer mass value will therefore vary according to the relative concentrations of the components, and direct correlation with a library cannot be achieved.

There is a further difficulty in obtaining certified data pertaining to mixtures to train an s.l.n. pattern recogniser, which is obviated by deriving the discriminant logic functions directly from those for the pure components and thereby eliminating mixture training routines. The variation in the data owing to concentration differences is circumvented by the generalisation of the networks. Experiments to date have demonstrated that mass spectra of mixtures of two components can be classified according to the functional groups of their components. Over 80 000 classifications into 135 data categories have been made, still remaining within the confines of a manufacturer's plant. Detection of both components in the mixture can be as high as 79%, whereas the detection of at least one of the components can be 98·5%.[15] The generalisation of the networks allow variations of up to 50% in the relative concentrations of the components without adversely affecting the overall performance.

7.4 Industrial automation

The need for pattern recognition in industrial automation is a growing requirement. To this end, a videocamera has been interfaced with a pattern recognition system[15] to provide an input facility for real data and to enable recognition studies to be carried out.

Two main approaches have been made, using s.l.n.s, to the problem of industrial automation typified by the need to recognise and sort components on a conveyor belt.

The first approach is to regard the components as shapes and select a training set in much the same way as is done with alphanumerics. The problems of normalising the data are more difficult in the industrial environment, and it is clear from the previous Sections that the patterns (shapes) must be aligned in a predetermined way to minimise Hamming distances between members of the same class of patterns, and thereby achieve optimum classification. The proprocessing must be done electronically because the moving and rotating of a camera is unacceptable. Othogonal shifting can readily be achieved on a pattern matrix, but rotation is more difficult. It has been achieved by the use of 'rotated' mappings. The s.l.n. is trained on a set of shapes aligned in a predetermined manner, and the test data is input to the recogniser via a set of mappings, each being related to the mapping used in training which in effect rotate the input. A classification is made at each rotation and the final decision made with reference to all the responses.

Although it would be expected in engineering that the shapes encountered would be of precise dimensions, the generalisation of a pattern-recognition system is still required to circumvent the distortions which can occur in the preprocessing, as illustrated by the rotation of a bar by rotation mappings in Fig. 16.

A second approach to industrial automation is recognition through tracking round the edge of an object. An s.l.n. controls the position of a window, which is trained to move around the edge of a shape. The system thus possesses the ability to track around similar shapes, whereby they can be recognised.[17]

7.5 Fault diagnosis in digital systems

Encouraging results have been achieved by employing an s.l.n. to monitor the function of a digital sequential system.

Test points throughout the circuit contribute to the elements of the input pattern, and the network is trained when the system is known to be functioning correctly.

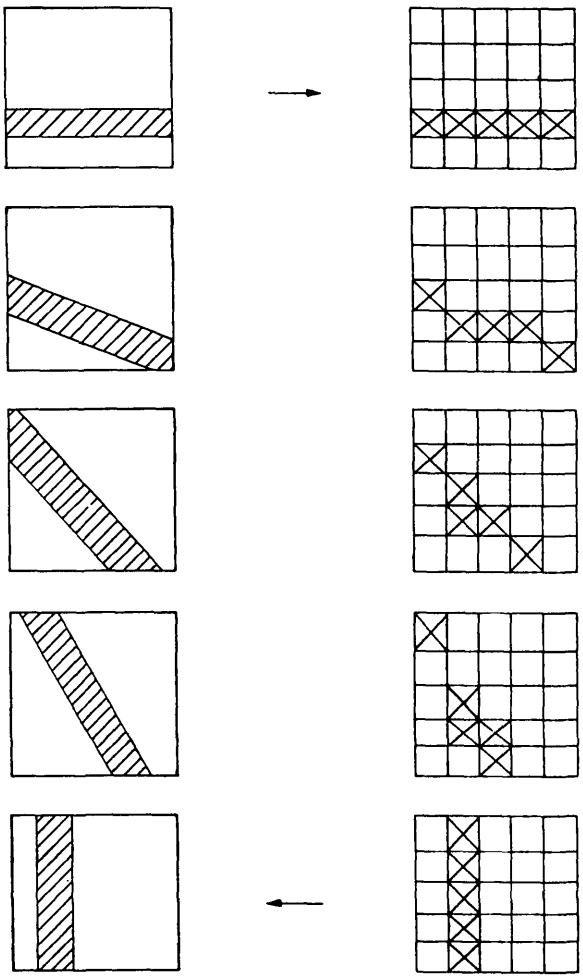


Fig. 16 *Use of rotation mappings*

The matrices on the right represent the rotation of a digitised pattern using mappings. The diagrams on the left represent the physical rotation of the object

If and when the circuit malfunctions, the outputs of some or all of the test points give rise to patterns which have not been seen during training and are outside the generalisation area of the s.l.n. and will not be recognised. A signal can then be output to signify malfunction of the circuit. Preliminary results indicating 90% detection of malfunctions have been achieved[18] to date.

7.6 *Processing of medical records*

The processing of medical records is being researched using s.l.n.s. This problem falls into the category of pattern detection in the first instance.

The aim is to endeavour to classify abdominal pain in its many forms from composite patterns representing the patient's medical history and physical examination upon admission as an emergency into hospital. Initial findings of the order of 70% correct diagnosis by the pattern recogniser are far in excess of the anticipated by random assignment of class, and suggest that the data is featured.

7.7 *Other applications*

Two further projects for which experimental results are not yet available are

(*a*) Signature verification, which would appear to be implementable by s.l.n.s. The verifier would only require one discriminator on which a threshold could be set to effect a simple acceptable/not-acceptable decision.

(*b*) Speech recognition, a traditional subject for pattern recognition which has not yet reached fruition. S.L.N. systems would be confined to small single-word vocabularies. The simplest system which has a high feasibility is an aid to speech therapy, whereby a recogniser could be set up for patient use to detect single words or syllables. Here, the accuracy requirements are not unduly high.

8 Concluding comments

This paper is intended to provide a guide to s.l.n. pattern recognisers, their behaviour and operation, and the types of problems to which they can be applied.

The technique can be used as a research tool because the recognition mechanism does not rely on any established theory of analysing of the data it is classifying. The opportunity for establishing new or hitherto unknown relationships between the measurements in the real world on which the patterns are in some way based and the overall data description exists. The technique can be used to detect patterns or recognise known established patterns.

The case studies pertain to specific research assignments, but the range of problems to which s.l.n.s can be applied is much greater. Other examples might include prediction of pharmacological activity of drugs from mass spectra, disease prediction from personal records, vehicle monitoring and in fact any data form which can be coded in a binary format can be assessed with SLNs.

Finally, the physical implementation of s.l.n.s is very flexible, varying from a conventional computer simulation, through microcomputers to hardware versions using r.a.m.s and r.o.m.s.

9 References

1 ALEKSANDER, I.: 'Microcircuit Learning Nets – Hamming distance behaviour', *Electron. Lett.*, 1970, **6**, pp. 134–135
2 STONHAM, T.J.: 'Improved Hamming distance analysis for digital learning networks', *ibid.*, 1977, **13**, pp. 155–156
3 STONHAM, T.J.: 'Automatic classification of mass spectra', *Pattern Recognition*, 1975, **17**, pp. 235–247
4 GOUTOS, G.P.: 'The classification of alpha-numeric characters with a random access memory system'. M.Sc. dissertation, University of Kent, 1978
5 ULLMAN, J.R.: 'Experiments with the *n*-tuple method of pattern recognition', *IEEE Trans.*, 1969, **C–18**, pp. 1135–1137
6 STONHAM, T.J., and ALEKSANDER, I.: 'Optimisation of digital learning networks when applied to pattern recognition of mass spectra', *Electron. Lett.*, 1974, **10**, pp. 301–303
7 STONHAM, T.J., and FAIRHURST, M.C.: 'Mechanisms for variable rejection rate in a *N*-tuple pattern classifier'. Internal Report, University of Kent, 1976
8 ALEKSANDER, I.: 'Workshop on pattern recognition'. Queen Mary College, 1975
9 BLEDSOE, W.W., and BROWNING, I.: 'Pattern recognition and reading by machine', Proceedings Eastern joint computing conference, 1959, pp. 225–232
10 CHEUNG, C.Y.: 'Some aspects of adaptive logic for pattern recognition'. Ph.D. thesis, University of Kent, 1973
11 STONHAM, T.J., and FAIRHURST, M.C.: 'A classification system for alpha-numeric characters based on learning network techniques', *Digital Processes*, 1976, **2**, pp. 321–339
12 STONHAM, T.J., and ALEKSANDER, I.: 'Automatic classification of mass spectra by means of digital learning networks', *Electron. Lett.*, 1973, **9**, pp. 391–393
13 STONHAM, T.J., ALEKSANDER, I., CAMP, M., PYKE, J., and SHAW, M.: 'Classification of mass spectra using adaptive logic networks', *Analyt. Chem.*, 1975, **47**, pp. 1817–1823
14 STONHAM, T.J., and ENAYAT, A.: 'A pattern recognition method for the interpretation of mass spectra and mixtures of compounds'. Internal Report, University of Kent, 1977
15 ENAYAT, A.: 'A pattern recognition method for the interpretation of mass spectra of mixtures of compounds'. M.Sc. Dissertation, University of Kent, 1978
16 NAPPEY, J.A.: 'JANSYS – a suite of programs for pattern recognition'. Internal Report AC/R/050, Brunel University, 1977
17 DAWSON, C.: 'Simply scene analysis using digital learning nets'. Ph.D. thesis, University of Kent, 1976
18 ALEKSANDER, I., and AL-BANDAR, Z.: 'Adaptively designed test logic for digital circuits', *Electron. Lett.*, 1977, **13**, pp. 466–467

Igor Aleksander was born in Yugoslavia, obtained his bachelor's degree in engineering at the University of the Witwatersrand in South Africa and his doctorate at the University of London, England.

He spent several years in industry before joining the faculties of the Universities of London and Kent. At present he is Professor of Electronics and deputy head of the Electrical Engineering department at Brunel University.

John Stonham graduated in electronics at the University of Kent at Canterbury in 1969. He continued postgraduate studies at Kent University and was awarded an M.Sc. degree in 1972 for research into parallel digital representation of analogue computing elements. In 1974 his research into automatic classification of mass-spectral data culminated in the award of Ph.D. and this was followed by a period of post-doctoral research supported by the UK Science Research Council. Dr. Stonham is currently a lecturer within the Department of Electrical Engineering & Electronics at Brunel University.