A Novel Adaptive Pattern Recognition Model with Sparse Associative Memory Networks

Yang Guoqing Chen Songcan Lu Jun Liu Chuan
Nanjing Aeronautical Institute
Nanjing , 210016 , People's Republic of China

[Abstract] N–tuple adaptive pattern recognition systerm— WISARD, invented by I.Aleksander, can be considered as a logic neural network using ordinary RAMs and has found extensive applications. It does, however, have weaknesses in large dimensional or non–deterministic pattern recognition problems since n–tuple size is confined under the lower value ($n<8$) due to the consideration of the cost.

Sparse distributed memory—SDM, proposed by Kanerva, is an associative memory model in a vast data space and a generalisation of Hopfield neural networks, its distributed access principle is very efficient to store correlated patterns in the same classifier.

On the basis of WISARD and SDM models, we suggest a novel two–layered adaptive system in which n–tuple recognition principle in WISARD is still followed and RAMs are replaced with SDM networks. In addition to all properties of WISARD, it can select n–tuple size in larger range without the limitation of the cost. Experiments on the Chinese character recognition have shown the feasibility of the model.

Key Words: Sparse distributed memory pattern recognition
neural network Chinese character recognition

1. Introduction

(1) WISARD System

WISARD system is the single–layered adaptive logic networks (Fig.1) which envolve essentially from neural modelling and use a combinational logic function as a simple neuron–like cell [1][2]. The system contains a hundred of functions implemented with ordinary RAM array. Each function needs one RAM unit which has 1 bit datum length and n bit address inputs, called a n–tuple, as the ones to each function. These logic functions are set up by input training patterns. One net, called a discriminator, is required for each class. Each function samples n points in random manner from a binary input pattern which can be the television resolution picture in the frame store. All n–tuple subpatterns to one RAM are the terms of corresponding logic function. During the cassifying phase, an unknown pattern stimulates the networks. RAMs, in read mode, are addressed by the n–tuple samples, that is, the logic functions are driven and respond with 0 or 1 according to whether the input subpatterns of input pattern have been trained. The overall network response is the arithe metical summation of all RAM outputs. The decision on the unknown pattern is made by the decision logic, usually depending upon maximum–response principles.

The salient properties of the system are non–algorithm, self–adaption, real–time and massive parallel distributed processing. It does, however, have some weaknesses which are (a) the cost of RAM nets is expotentially raised as the n–tuple size increases so that the performance can not be optimized in some problems since the n–tuple size is limited under certain value, $n<8$ usually. (b) Given n, the more the training patterns, the more the locations set up, the 'saturation' of RAMs will finally be incured. (c) Without considering the correlation between training patterns or n–tuple groups. Because of these, the system is not very efficient to cope with large dimensional or non–deterministic pattern data recognition problems.

(2) Sparse Distributed Memory Model

Consider a conventional random access memory with very large address. For $n=1000$, n address length, 2^n possible addresses are larger than the number of atoms in the known universe. Obviously, there is no way of associating all, or even a relatively small fraction, of these addresses with physical storage locations. How can one construct an associative memory using these large address? Kanerva's answer is as follows: pick at random m addresses to be associated with physical storage locations (m might be a million to a billion). Because m is small compared with 2^n, these randomly chosen addresses represent a set of storage locations that is sparsely distributed in the address space [3].

A schematic picture of the functioning of the SDM as autoassociative memory is shown in Fig.2.[4]. M addresses associated with storage locations are contained in the matrix A. The input address comes in at the top as vector **a**. All addresses in A close to the input address are selected. If we view these n–bit addresses as points in an n–dimensional address space, the selected addresses will lie within a (hyper) sphere of hamming radius d_H centered at the input address which is quite unlikely to point to any one of the m randomly chosen points. These addresses have their select bit set to 1 in the vector S to select locations in C; all others in S are 0_s . The data are written into the selected locations.The procedure is a little more complicated than for a conventional computer. Instead of just replacing the old contents of a storage location with new data,the new data is added to the previous contents. Thus each of the storage locations in the SDM is actually a set of n counters (N.B. suppose that data word is the same size as the address length here). The reason is that there possibly are two or more data vectors written into any given storage location if the spheres chosen by two input addresses overlap. In read operation, the contents of the selected locations are added together to give n–sums in the field **h**. Finally, these sums are thresholded to yield output data. Note that this is a statistical reconstruction of the original data word. The output data should be the same as the original data as long as not too many other words have been written into memory.

2.Two–Layered Adaptive Pattern Recognition System—TAPRS

On the basis of models both WISARD and SDM, we propose a new two–layered adaptive pattern recognition system whose architecture is shown in Fig.3. The first layer, called sparse state matrix A, consists of k sparse RAMs, denoted by $A_i, i=1,\cdots,k$. Each

RAM has n bit data length, m locations. N inputs connect at random or regularly n points from input pattern.Locations are set up before training with m states in each A_i which are sparsely distributed in 2^n address state space, $m << 2^n$, so that n can be chosen in larger range. The size of A is $k * m * n$ bits. The outputs of A construct a $k * m * g$ dimension binary vector S. The second layer contains matrix C, SUM and **d** vectors. The matrix C, addressed by S, consists of g columns, each of which represents one discriminator, classifier, similar to the one in WISARD. The size of C is $k * m * b * g$. During the training, each classifier is trained individually. Each location,b bit length, of specified classifier operates in counter mode and records the occurence frequencies of each state of all n–tuple subpatterns in the training pattern set. In classification phase, all classifier work simultaneously in read mode. When an unknown pattern inputs, the contents of all selected locations in each classifier are accumulated to form the summation vector SUM. The decision logic thresholds the SUM and decides the classifier to which the unknown pattern belongs.

If $m = 2^n$, $d_H = 0$, the TAPRS will degrade into the WISARD system.

3.Experiment Results

The key parameters of the TAPRS are n–tuple size, the number, m, of physical locations in A_i and hamming radius d_H. The experiments with the parameter optimization are carried out in virtue of the multi–font Chinese character recognition. Thirty printed Chinese characters are normalized into 24×24 dot matrics. Each character has four samples with different font types. Fig.4 shows the samples of the Chinese character '啊'.

(A)Performance versus n

N is the size of n–tuple to be mapped to input addresses of A_i from input pattern. In Fig.5, dot and solid curves represent the performance versus n in WISARD and TAPRS respectively. $N = 6$ is the optimum value in the WISARD, whereas $n = 14$ and $n = 6$ are the optimum values in the TAPRS. We can see that the performance at $n = 14$ is better than that at $n = 6$. Because of the merits of sparse RAMs, the system cost increases linearly as n increases so that n can be chosen in larger range, while the system cost in WISARD increases expotentially as n increases so that n can only be chosen under 8 usually.

(B)Performance versus m

On a given n–tuple, the optimum m is defined as the minimum number of the locations in A when the best performance is obtained. The selection of m is related to n–tuple size, hamming distance and pattern data properties. For the experiment undrer taken in this paper, given $n = 14$, the curve of performance versus m is shown in Fig.6. When $m < 64$, the correct recognition rate is raised linearly as m increases, while for $m > 64$, the rate remains almost costant. Hence, $m = 64$ is the best choice in the experiment.

Taken $d_H = 5$, a group of performance curves vesus n and m are shown in Fig.7. The other curves are not listed here.

4.Conclusions

In addition to the properties of the WISARD, including self–adaption, versatility, non–algorithm, massive parallel distributed processing, the TAPRS is better than the WISARD in the aspects of large–dimensional or non–deterministic pattern data recognition problems. The main reasons are:

(1)The sparse RAMs are used for both address and data array so that n–tuple size can be selected in larger range without the limitation of the cost since the number, m, of physical locations is increased linearly as n–tuple size.

(2)Distributed access approach to sparse RAM is used so that pattern correlation is considered. Given training set, as long as the hamming radius is appropriately chosen, the system can obtain enough large generalisation set which has the best overlaps with pattern data set related to the training set and the minimum overlaps with others.

References

1. I.Aleksander et al. : WISARD a Radical Step Forward in Image Recognition Sensor Review, July, 1984, pp120–124.
2. I.Aleksander et al. : A Guide to Pattern Recognition Using Random Access Memories, IEEE Journal, Comp.Digit.Tech,Vol.No.1, 1979.
3. P.Kanerva : Sparse Distributed Memory, Cambridge, MA : MIT Press, 1988.
4. D.J.Keeler : Comparision between Kanerva's SDM and Hopfield–Type Neural Networks, Cognitive Science 12, 1988, pp229–329.

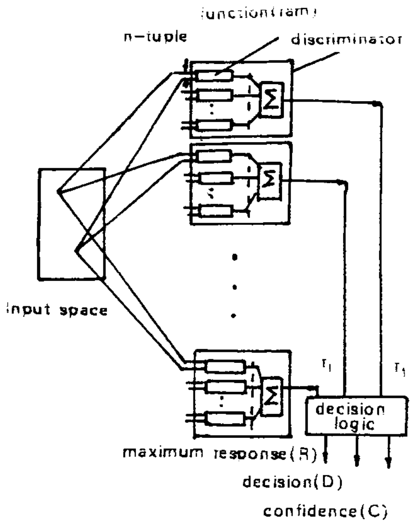

Fig.1 Single-layered adaptive pattern recognition system.

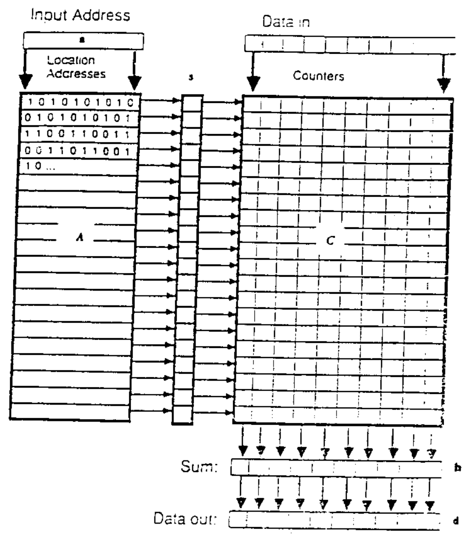

Fig.2 Sparse distributed memory model for autoassociation.

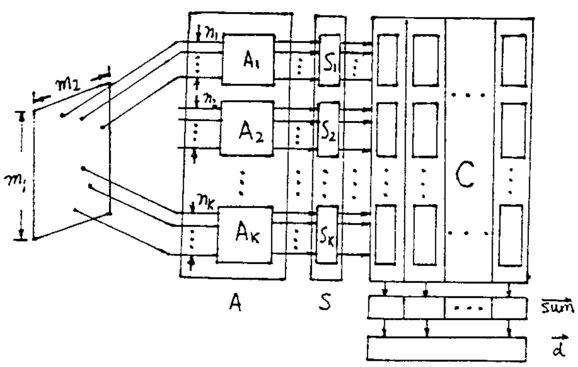

Fig.3 New two-layered adaptive pattern recognition system.

啊 啊 啊 啊

Fig.4 Samples of machine-printed chinese character '啊'.

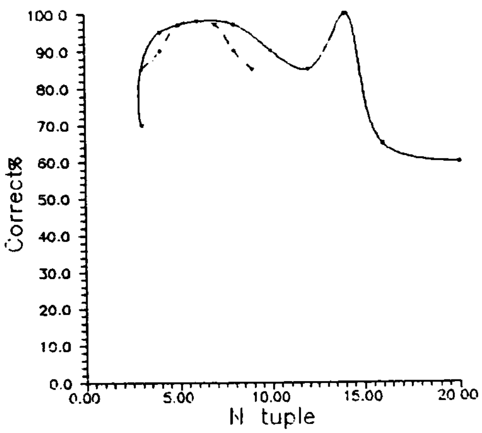

Fig.5 Recognition performance versus n for TAPRS.

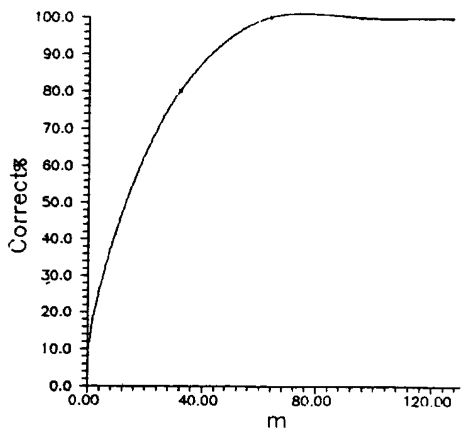

Fig.6 Recognition performance versus m for TAPRS.

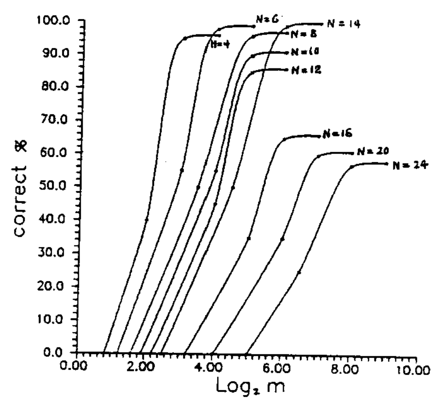

Fig.7 Recognition performance versus n,m for TAPRS, $d_H = 5$.