The algorithm outlined here can be understood as a search in a classifying tree with erroneous data. In contrast to the only error free solution (full search), the algorithm reduces search time significantly, tolerating some errors in the result. A systematic statistical analysis of the algorithm is left to further research, but testing with some hundered images showed promising results.

5 Implementation and Results

The simple associative memory was tested extensively with different forms of coding with and without regularisation. Some results are given in previous papers [3,4]. The scale space approach with sequential search, mentioned here in the following benefits very much from the speed of the simple recognition system[1]

The real implementation of the scale search process uses a central filtered representation of the input image thus needing only one (time consuming) filtering operation per association. A result in case of a relatively difficult recognition task is given in Fig. 2.

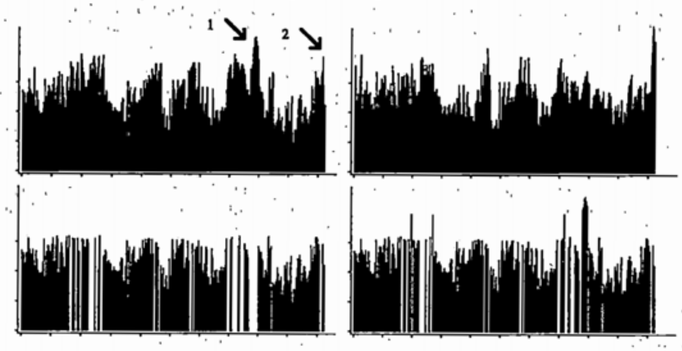

Figure 2: *A typical result of the search process in scale space: the left pictures show the sumhistogram and the inhibitory histogram at the start of the search process, the right ones at its successfull end. All histograms display correlation against pattern number. The memory contained 512 patterns of 128 square pixel grey valued camera images with laboratory scenes. The input image (corresponding to stored images at arrow 2) was shifted by 15 pixels down and 13 left and associated. Note that the unshifted image was nearly satisfactory matched to some "false" patterns (see arrow 1), but the matching procedure inhibits these solutions in the search process and took the right pattern.*

References

[1] G.-J. Giefing, H. Janßen, and H. A. Mallot. A saccadic camera movement system for object recognition. this volume, 1991.

[2] D. Hubel and Wiesel. Functional architecture of macaque visual cortex. *Proc. R. Soc. London B*, 198:1 – 59, 1977.

[3] H. Janßen and J. Kopecz. Image representation in hypercolumnar structure my means of associative memory. In *Close-Range Photogrammetry meets Machine Vision*, pages 1050 – 1057. ISPRS, SPIE, 1990.

[4] H. Janßen, J. Kopecz, and H. A. Mallot. Erkennung natürlicher Bilder mit Hilfe diskreter parametrischer Bildrepräsentationen und Assoziativspeichern. In *Mustererkennung 1990, Informatik Fachberichte 254*, pages 232 – 233. DAGM, Springer-Verlag, 1990.

[5] H. A. Mallot, W. von Seelen, and F. Giannakopoulos. Neural mapping and space-variant image processing. *Neural Networks*, 3:245 – 263, 1990.

[6] G. Palm. On associative memory. *Biol. Cybern. 36, 19-31*, 1987.

[7] M. Porat and Y. Y. Zeevi. The generalized gabor scheme of image representation in biological and machine vision. *IEEE PAMI*, 10:452 – 468, 1988.

[8] D. J. Willshaw, O. P. Buneman, and H. C. Longuet-Higgins. Non-holographic associative memory. *Nature*, Vol. 222:960–962, 1969.

[1] With a pattern vector length of 3136 bit and a memory size of 3136 patterns (1.2 MBytes) the recognition time on a SUN-4 is about 1/10 second.

Artificial Neural Networks
T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (Editors)

DESIGN IMPROVEMENTS IN ASSOCIATIVE MEMORIES FOR CEREBELLAR MODEL ARTICULATION CONTROLLERS (CMAC)

P.C. Edgar An[1], W. Thomas Miller III[1], P.C. Parks[2]

[1] Electrical and Computer Engineering Department, University of New Hampshire, Durham, NH 03824, USA.

[2] Applied and Computational Mathematics Group, Royal Military College of Science, Shrivenham, Swindon, SN6 8LA, UK.

Abstract

A number of recent improvements to the design of associative memories for CMAC systems are described. These are (i) an improved scheme for allocating C weights to a given input vector in R^N, (ii) design of receptive field shapes within the hypercube associated with an individual weight (including some experimental evaluations of these shapes), (iii) matching the field shapes to the hypercube itself using the concept of "superspheres", (iv) speeding up the convergence of the weight training procedure.

1. INTRODUCTION

The "Cerebellar Model Articulation Controller" (CMAC) was introduced by J.S. Albus [1] in the early 1970s. At the heart of the CMAC system is an associative memory which "learns", in a training procedure, the non-linear function $y = f(s_1, s_2, \ldots, s_N)$, where $\underline{s} = (s_1, s_2, \ldots, s_N)$ is the input or "stimulus" vector in R^N. In the control application $y = y_t$ is usually the scalar output at discrete time t of a non-linear difference equation of a system in which $s_1, s_2, \ldots, s_N$ are in fact $y_{t-1}, y_{t-2}, \ldots y_{t-\ell}, u_{t-1}, u_{t-2}, \ldots u_{t-m}$, where $\ell+m = N$ and the sequence $\{u_t\}$ is the input to the system. The memory thus "learns the system dynamics". This is achieved by calculating y as the average sum of a fixed number of C "weights" $x_{(i)}$ corresponding to the given input vector $\underline{s}$. We can represent this choice by ordering the weights in a 1xp vector as $\underline{x}^T = (x_1, x_2, \ldots, x_p)$ and considering a 1xp binary "associative vector" $\underline{a}^T$ consisting of 1s in the positions corresponding to chosen x_i's and zeros elsewhere. The desired property of this $\underline{s} \rightarrow \underline{a}$ mapping (the "receptive field center placement problem") may be summed up in the following statement:

> "For input vectors $\underline{s}_k$ lying within a certain neighbourhood of a chosen input vector $\underline{s}_j$ the Hamming distance between the corresponding association vectors $\underline{a}_k$ and $\underline{a}_j$ should be linearly proportional to the Euclidean distance between $\underline{s}_k$ and $\underline{s}_j$".

2. THE RECEPTIVE CENTER PLACEMENT PROBLEM

The receptive field centers in the standard CMAC procedure developed by Albus [1] are distributed on the grid of each of C hyperplanes with receptive field width of C. Each of two consecutive hyperplanes are relatively offset by one quantization unit in each dimension (fig 1).

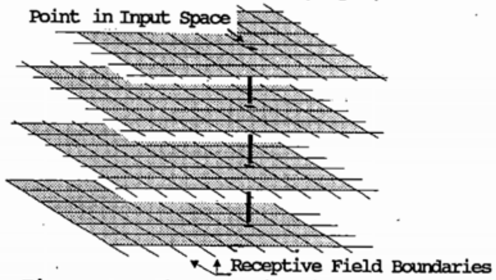

Figure 1 The organization of centres for the two-dimensional CMAC with C of 4. The dark square in each plane indicates the activated cell. (N=2 C=4)

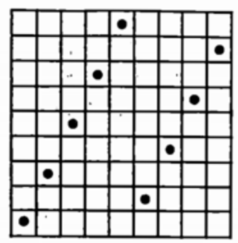

Figure 2 The solid dots show the centers in the two-dimensional hypercube (square in this case). The generalisation level was 9, and D was <1,2>. (N=2, C=9)

We have developed a rule which provides a more uniform distribution of receptive fields while preserving desirable features of the standard CMAC. To explain the rule, we adopt a modular arithmetic expression to define the center co-ordinates. The co-ordinates of the mth center inside the hypercube is defined as [m·d1%C, m·d2%C,···m·dN%C], where % is the modulo operator, and m ranges from 0 to C-1. It is easily seen that the first center is always fixed at the origin [0,0··,0]. To be precise, a displacement vector D is defined as <d1,d2,··dN>, where d1 is set to 1. The center distribution is completely described by the vector D. D is a unity vector for the standard CMAC. The rule R1 below, provides a better offsetting scheme for the receptive cells:

$$\alpha \cdot d_i \neq \gamma \cdot C, \quad \text{where} \quad 0 \le i < C, \quad 0 < d_i < C/2, \quad d_i, \alpha, \gamma \in I$$

All di's which satisfy R1 are potential candidates for the components in D. The range of di is limited up to C/2 due to the mirror image symmetry. If there are more choices for di than N, di should be chosen in such a way to maximize the deviation among all chosen di's. If there are fewer choices for di than N, di should be chosen so that the di's are selected with roughly equal frequencies. For example, for a CMAC with C of 9 and N of 2, possible di are {1,2,4}. A vector D of <1,2> satisfies all the constraints. If this is chosen, the co-ordinates for the centers are [0,0], [1,2], [2,4], and so forth. The arrangement of these centers is shown in Fig. 2.

Parks and Militzer [2] generated a set of vectors D for many combinations of C and N by using an exhaustive search. The evaluation was based on maximising the minimum distance between two nearest cell-center neighbours. Reference [4] gives tables of the optimised vectors <d1, d2,...dN> for 2≤N≤10 and 2≤C≤50.

3. EXPERIMENTAL EVALUATION OF RECEPTIVE FIELD SHAPES

The uniformity of receptive field coverage can best be evaluated in terms of function approximation. The space-spanning function of interest in this paper is a constant output. One way to evaluate the uniformity of coverage is to set all the weights equal to 1, and to evaluate the network output at many random points in the hypercube of side C in the input space.

Experiments were done on a CMAC with C from 2 to 50 both with N=3 and N=10 input spaces. Linearly tapered radial basis and square basis receptive field functions were used with R1 center placement, (B and C in Figs 3 & 4). These functions were then compared with the modified standard CMAC (diagonal center placement and the linearly tapered field shape with the square basis contour), (A in Figs 3 & 4). The network outputs were evaluated at 1000 random inputs inside the hypercube, and the ratio of the standard deviation to the mean of these responses was computed. The results are plotted in Figs 3 & 4.

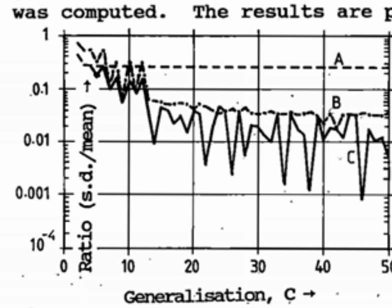

Figure 3 The ratio of the deviation to the mean of the total summed field strengths. (N=3, C variable)

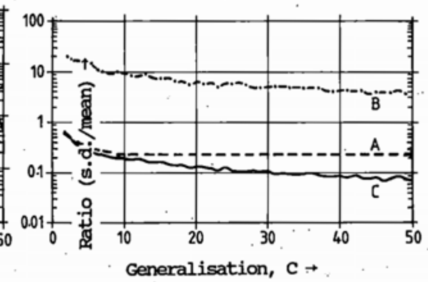

Figure 4 The ratio of the deviation to the mean of the total summed field strengths. (N=10, C variable)

We conclude that the R1 center placement produces a better fit without introducing any penalty in computation. The square basis function (C in Figs 3 & 4) is best suited to CMAC due to the hypercubic nature of the building block, unless a more sophisticated field shape such as that suggested in Section 4 is introduced.

4. USE OF SUPERSPHERES

A problem which arises with the receptive field shapes described in section 3 above is that discontinuities in partial derivative of the stored function f can arise. A natural scheme to match receptive field shapes to the CxCx..xC N dimensional hypercube which forms the boundaries of a receptive association cell is to use a "supersphere" [3].

The superspherical contours may be combined with a cosine field shape to give a field shape $z(\xi_1, \xi_2, \ldots, \xi_N)$ defined by the transcendental equation

$$|\xi_1/\tfrac{1}{2}C|^{1+1/\sqrt{z}} + \ldots + |\xi_N/\tfrac{1}{2}C|^{1+1/\sqrt{z}} = \{(1/\pi)\cos^{-1}(2z-1)\}^{1+1/\sqrt{z}}$$

Here $(\xi_1, \xi_2, \ldots, \xi_N)$ is a local coordinate system based on an origin at the center of the CXCX...XC N-dimensional hypercube. The shape of z for the case N = 2 is sketched in Fig 5.

5. SPEEDING UP CONVERGENCE OF THE LEARNING ALGORITHM

Here it will be simply recorded that Parks and Militzer [4] have done much work on speeding up the learning algorithm - devised originally by Albus [1] as a "Hebbian learning" process.

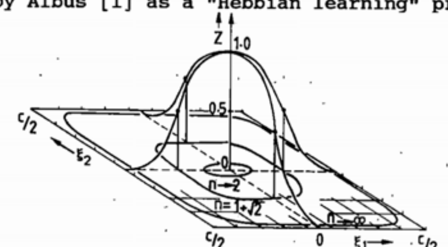

Figure 5 The receptive field shape $z(\xi_1, \xi_2)$ described in Section 4, (N=2, C=9).

6. REFERENCES

[1] Albus, J.S. (1975) "A new approach to manipulator control: the cerebellar model articulation controller (CMAC)" Trans. ASME, Series G, J. Dynamic Systems, Measurement and Control, 97 220-233.

[2] Parks, P.C. and Militzer, J. (1991) "Improved Allocation of Weights for Associative Memory Storage in Learning Control Systems", preprint. Proc. IFAC Symposium on Design Methods of Control Systems, Zürich, September 1991.

[3] Gardner M. (1977) "Mathematical Carnival", Chapter 18, Vintage Books, Random House, New York.

[4] Parks P.C. and Militzer J. (1989) "Convergence properties of associative memory storage for learning control systems and their enhancement" Proc. IFAC Symposium on Adaptive Systems in Control and Signal Processing, Glasgow, April 1989, Proc. 2 565-572, Pergamon Press, Oxford.

Artificial Neural Networks
T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (Editors)

Implementing a "Sense of Time" via Entropy in Associative Memories

Larry M. Manevitz

Department of Mathematics and Computer Science
University of Haifa
Haifa, Israel

Abstract

In this paper we show how noise in a neural network can be naturally used to store certain kinds of temporal information. The mathematical model developed here is an adaptation of the static memory model (Sparse Distributed Memory) of Kanerva, assuming a Poisson distribution of noise. As an illustration certain kinds of Pavlovian learning can be accounted for by this mechanism.

Introduction

Consider the paradigm Pavlovian experiment; a bell is rung and then the animal is fed. After several repetitions of this it is observed that the animal salivates (in anticipation of being fed) even in the absence of food. Some experiments have been done testing how large a variance in time between bell and food is tolerable in the learning process. However, suppose the time is kept constant, say at five minutes. Then the animal also learns. as a by-product of the other learning, the time period five minutes. That is, it is expected that the salivation would start only near the five minute period.

Assuming this is the case, what is a good model for the method of storage of this temporal information? Since the learning is presumably done by some mechanism of reinforcement, and since it is not possible to know in advance which items will be the time-dependent ones, nor which are the relevant time periods, it seems unlikely that the mechanism can consist as some sort of "time-stamping", e.g. as in [BM].

Moreover, using introspection, it seems that there is some "sense of time" both in current consciousness and in certain memories. How can this be naturally stored? The "naturality" implies that the information should not need to be heavily pre-processed. One method is by pointer chains; which is suggested by Kanerva [K]. Here the temporal information is stored as a function of the number of elements in the chain. However, this is very inefficient since this would require enormous replication of information. (This is related to the "Frame Problem" [Br].)

Some sort of "dynamic" memory where the contents changed with time would be a convenient mechanism; then the rate of change can serve as the implicit storage of the temporal data.

In Kanerva's model [K] (or Albus [A] or Marr [Mr]) memory is stored and retrieved probabilistically by an averaging process over many memory cells. In this situation a very simple additional assumption, which seems quite natural physiologically, can account for time varying memory. That is, if there are sufficiently many cells averaged and it can be assumed that they have different decay times, then the memory at a location can be thought of as a dynamic one which changes as some of the cells decay.

Such a differentiation seems physiologically plausible, since one would expect that in any collection, e.g. of brain cells, there would be connections of different qualities amongst the dendrites, but that the *distribution* of quality amongst the dendrites should be a constant. It would seem to make evolutionary sense

During part of this research the author was a NASA-STANFORD-ASEE Faculty Fellow at NASA Ames Research Center.