WCNN96, Proceedings of the 1996 World Congress on Neural Networks, September 1996.

Reformulating Distributed Associative Memories for Image Classification

Fabio Ancona, Stefano Rovetta, and Rodolfo Zunino

Dept. of Biophysical and Electronic Engineering (DIBE), University of Genova
Via all'Opera Pia 11a,16145 Genova, Italy — e-mail(s): {ancona, ste, zunino}@dibe.unige.it

Abstract

The theoretical model of Distributed Associative Memories (DAMs) is reformulated by simple algebraic derivations that make the memory device practically applicable to high-dimensional, visual data processing. In particular, the analysis shows that the weight of both *the computational cost for retrieval and the physical memory occupation can be reduced from N^4 to N^2, where N is the data dimensionality. The presented research extends previous results on the model implementation on parallel machinery, whose ultimate efficiency is now characterized from a theoretical point of view, too.*

1. INTRODUCTION

Associative Memories (AMs) provide a connectionist storage system with content addressability and robustness [1]. AMs are typically used for high-dimensional data-processing tasks, for which data dimensionality hinders a direct application of classical Neural Networks. Application domains for AMs include robust image recognition and understanding.

This paper reconsiders Kohonen's classical Distributed Associative Memories (DAMs) [2]. Previous research has shown the effectiveness of DAMs for image recognition purposes [3,4], even though the original model has a limited practical impact, due to the physical sizes of the data structures involved. From this standpoint, the paper extends to a higher degree of formalism and a wider general validity a study presented in [5]. That research mainly addressed implementation features on parallel machinery to stress overall performance.

With respect to those preliminary results, the paper generalizes a reformulation of the basic model that reduces *both* physical memory occupation and recall computational timing. In particular, the analysis shows from a theoretical point of view that the relative weights of these quantities are reduced from N^4 to N^2, N being the image size. As an obvious confirmation of early results, the new model maintains those specific features that make memory implementation on parallel hardware very efficient. Therefore, in the paper, the robustness, efficiency and performance of a parallel implementation are also discussed, as they make the method applicable to real-time domains. An abstract model of the parallel implementation also makes it possible to derive a theoretical expression of the resulting efficiency of the supporting architecture; experimental results point out the validity of theoretical predictions by matching predicted values with remarkable accuracy.

2. THEORETICAL FRAMEWORK FOR ASSOCIATIVE MEMORY

2.1 - Kohonen's basic model of DAMs

An associative device couples each stored datum with related information (*key*), which is used for both storage and retrieval. The present context assumes an auto-associative schema, in which a datum and a key coincide; autoassociativity has been proved to contribute to increasing the noise insensitivity of classification schemata for quite different memory models [5-7]. This section summarizes for the reader's convenience the basics of the DAM model.

Although the original model (DAMs) relates to biological issues, both writing and reading can be expressed in terms of matrix multiplications. For simplicity and without loss of generality, square images holding N x N pixels are assumed. In the following, $\mathbf{p}[N^2]$ denotes a vector holding a stored image (pixel rows are arranged sequentially). If I images are stored in the memory, $\mathbf{S}[N^2 \cdot I]$ denotes the "data" matrix whose columns contain all vectors $\mathbf{p}$. Finally, $\mathbf{M}$ indicates the actual memory device.

Training - The analytical expression [2] for a memory matrix, $\mathbf{M}$, holding a data set, $\mathbf{S}$, is given by:

$$\mathbf{M} = \mathbf{S}(\mathbf{S}^t\,\mathbf{S})^{-1}\,\mathbf{S}^t \qquad [\, N^2 \cdot N^2 \,] \tag{1}$$

where the numbers in brackets indicate the resulting dimension of the memory matrix (proportional to N^4).

Recall **-** Memory reading simply involves the multiplication of the addressing vector (row-scanned image), **k**, by the memory matrix, **M**, thus yielding the recall vector, **r**:

$$\mathbf{M}\,\mathbf{k} = \mathbf{r} \qquad [\,N^2 \cdot N^2\,] \cdot [\,N^2\,] = [\,N^2\,] \tag{2}$$

Analysis **-** As already pointed out in [5], memory size represents a crucial issue of Kohonen's model because, especially for normal-size images, physical memory occupation is proportional to N^4. The computational load (time) of the reading process is $T_C^{(\text{Kohonen})} = N^4 \cdot (T^* + T^+) - N^2 \cdot T^+$, where T^* and T^+ represent the times needed for one floating-point multiplication and sum, respectively. For example, an application involving images with 128x128-pixel images yields a memory with 256M elements. It should be stressed that these values depend on the actual data dimensionality and not on the number of stored prototypes.

2.2 - Reformulation to a Practical Model

The issues defined in the previous Section make the implementation of Kohonen's algorithm difficult or impossible on most types of machinery. However, the theoretical model allows a reformulation of the matrix **M**, which is split into two partial matrixes, $\mathbf{M_1}$ and $\mathbf{M_2}$. The analysis shows that this procedure preserves the basic memory functioning.

Training **-** The two matrixes, $\mathbf{M_1}$ and $\mathbf{M_2}$, are computed (off line) during training, as follows:

$$\mathbf{M}_2 = \mathbf{S}^t \qquad [\,I \cdot N^2\,] \tag{3}$$

$$\mathbf{M}_1 = \mathbf{S}\,(\mathbf{S}^t\,\mathbf{S})^{-1} \qquad [\,N^2 \cdot I\,] \tag{4}$$

*Recall***-** Memory reading is split into two subprocesses: first, multiplying the key vector, **k**, by $\mathbf{M_2}$ gives the intermediate vector, **v**; secondly, the product of **v** by $\mathbf{M_1}$ yields the actual recall:

$$\text{a)}\quad \mathbf{M}_2\,\mathbf{k} = \mathbf{v} \qquad [\,I \cdot N^2\,]\cdot[\,N^2\,] = [\,I\,] \tag{5}$$

$$\text{b)}\quad \mathbf{M}_1\,\mathbf{v} = \mathbf{r} \qquad [\,N^2 \cdot I\,]\cdot[\,I\,] = [\,N^2\,] \tag{6}$$

Proof of memory functioning **-** The verification of the consistency of the reformulated model is a straightforward consequence of simple marix-algebra properties:

$$\mathbf{r}^{(\text{new})} = \mathbf{M}_1(\mathbf{M}_2\,\mathbf{k}) = \mathbf{S}(\mathbf{S}^t\,\mathbf{S})^{-1}\,(\mathbf{S}^t\,\mathbf{k}) = [\mathbf{S}(\mathbf{S}^t\,\mathbf{S})^{-1}\,\mathbf{S}^t]\,\mathbf{k} \tag{7}$$

$$\mathbf{r}^{(\text{Kohonen})} = \mathbf{M}\,\mathbf{k} = [\mathbf{S}(\mathbf{S}^t\,\mathbf{S})^{-1}\,\mathbf{S}^t]\,\mathbf{k} \tag{8}$$

The equivalence of expressions (7) and (8) shows that the recall vectors for the two models coincide, thus proving algebraic consistency.

Analysis **-** As far as physical memory occupation is concerned, $\mathbf{M}_1$ and $\mathbf{M}_2$ have an identical number of elements that is equal to $I \cdot N^2$, hence the total number of stored physical elements is $2 \cdot I \cdot N^2$ elements. This makes the total memory occupation depend on the number of stored images, and might appear as a theoretical drawback in terms of generality. On the other hand, the advantage over the original model (requiring N^4 elements) is notable, if one (reasonably) assumes I to be much smaller than the number of pixels in an image. For example, if $I < 1000$, we obtain a saving in memory of at least one order of magnitude, which represents a good threshold for the method's usefulness.

A similar conclusion can be drawn when considering the computational timings required by the reformulated model. The timings for the two individual recall steps and the total recall step are given by:

$$T_C^{(a)} = (I \cdot N^2) \cdot T^* + [I \cdot (N^2 - 1)] \cdot T^+ \tag{9}$$

$$T_C^{(b)} = (I \cdot N^2) \cdot T^* + [(I - 1) \cdot N^2] \cdot T^+ \tag{10}$$

$$T_C^{(\text{tot})} = T_C^{(a)} + T_C^{(b)} = N^2 \cdot (2 \cdot I \cdot T^* + 2 \cdot I \cdot T^+ - T^+) - T^+ \tag{11}$$

The corresponding quantity for the original model amounts to:

$$T_C^{(\text{Kohonen})} = N^4 \cdot (T^* + T^+) - N^2 \cdot T^+ \qquad (12)$$

A comparison of expressions (11) and (12) shows that the model reformulation results in a substantial saving in computational load. Indeed, the most relevant factor is proportional to $2N^2$ instead of N^4, which represent an impressive improvement for very large value of N (as is the case with image recognition).

In summary, the main conclusion from this analysis is that the new model provides a system designer with the possibility of massive reduction in complexity *and* storage requirements. In both cases, as shown in Table I, the improvement is of the order of a power of two of the image size, N.

Table I - Comparisons of complexity and storage between the original and reformulated models

	Memory occupation	*Computational Cost*
Original DAM model	$\propto N^4$	$\propto N^4$
Reformulated model	$\propto N^2$	$\propto N^2$

With respect to the research presented in [5], which dealt only with preliminary issues on parallel implementations, the above derivations confirm the overall validity of the model reformulation to a wider extent. Quite interestingly, the analysis shows that the model reformulation does not require the tradeoff between memory occupation and computational cost that is usually brought about when porting theoretical models to practical applications.

3. THEORETICAL ANALYSIS OF PARALLEL IMPLEMENTATION

The possibility of efficient implementation is crucial for an associative model's effectiveness in practical applications, especially when coping with real-time domains. The research here presented maintains the basic approach adopted in [5,8]. Transputers still provide inexpensive and flexible support for development, but the present analysis yields a general model valid for other and possibly more effective HW machinery.

The implementation methodology follows a slice-oriented approach for the data-allocation strategy, and is supported by a hierarchical processor configuration, whose topology is arranged as a tree. The method used to download the memory recall process is independent of the tree depth and is based on a uniform distribution of matrix multiplications among processors. This strategy consists in splitting all memory matrixes into as many *slices* as the number of processors. The basic algorithm for run-time recall and subsequent image classification is detailed in [5], and entrusts each processor with the computation of partial slice-matrix products. Top-down and bottom-up communications support data dispatching and result recollection accordingly.

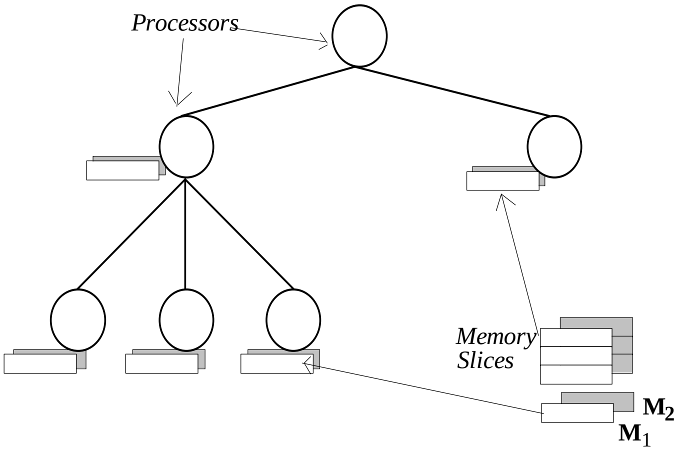

Fig.1 - Data allocation and processor organization

In the following, we derive an expression of the efficiency of the overall parallel systems for the recall process. Intermediate steps leading to the eventual expression are omitted for brevity. In the formula, M denotes the number of processors, τ_{sum}, τ_{mul}, τ_{pixel}, the elementary timings for summing a pair, floating-point multiplying a pair, and transmitting one pixels value(s), respectively, whereas $F^{(b)}$ and $F^{(d)}$ are constant, topology-dependent factors. The predicted value of efficiency, η, is:

$$\eta = \frac{2IN^2\tau_{mul} + \left(2IN^2 - I - N^2\right)\tau_{sum}}{2IN^2\tau_{mul} + (2I-1)N^2\tau_{sum} + M(M-I-1)\tau_{sum} + \left[\left(2N^2+I\right)F^{(d)} + IMF^{(b)}\right]\tau_{pixel}} \tag{13}$$

The analytical derivation (13) allows one to compare predicted values with experimental evidence both for the tests presented in [5] and other experiments run successively. Table II summarizes measured timings, obtained efficiency values and matches them against theoretical predictions.

Table II - Recall timings and efficiency values

(M = 8)	*Timings*		*Efficiency*	
	T_{seq}	T_{par}	*Measured*	*Predicted*
$I = 8$	1.777	0.411	0.54	0.575
$I = 16$	3.488	0.586	0.74	0.733

T_{seq} , T_{par}= *recall timings* for the sequential and parallel algorithm, respectively

The data shown in the table confirm the validiy of the theoretical model of the parallel process. An important property of this schema is that the system's efficiency increases with the number of stored prototypes; this confirms theory (13) which predicts:

$$\lim_{N\to\infty} \eta = 1 \tag{14}$$

The asymptotic behaviour (14) witnesses the robustness of the implementation schema; this result is a direct consequence both of the specific memory model, allowing a slice-oriented data allocation, and of the topological processor arrangement.

5. CONCLUSIONS

The present paper did not address the problem of recognition performance. This intentional omission is motivated by the fact that the properties of associative classifiers are well known [6-8], and, in the specific case of DAMs, accuracy at image classification has been already studied and analyzed in [2,5]. Conversely, the presented research aimed to clarify general properties of the basic associative model, which may enhance its implementation on practical application from a mostly general point of view.

REFERENCES

[1] Hinton G.E., Anderson J.A. (Eds.) *Parallel Models of Associative Memory - (Updated ed.)* Lawrence Erlbaum, New York, 1989.

[2] Kohonen T., Oja E., Lehtio P., "Storage and processing of information in DAM systems" in G. E. Hinton, J. A. Anderson (Eds.) *Parallel Models of Associative Memory - (Updated ed.)* Lawrence Erlbaum, NY, 1989.

[3] Wechsler H. Zimmermann J.L., "2-D Invariant Object Recognition Using Distributed Associative Memory", *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. PAMI 10-6, 1988, pp. 811-821.

[4] Polzleitner W. Wechsler H., "Selective and Focused Invariant Recognition Using Distributed Associative Memories (DAMs)", *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. PAMI 12-8, 1990, pp. 809-814.

[5] Garramone S, Pettinella M, Rebora A, Zunino R "Parallel implementation of distributed associative memories for image classification", Proc. World Congress on Neural Networks WCNN'95 - vol.I, pp.454-457

[6] Anguita D., Parodi G., Zunino R., "Associative structures for vision", *Multidimensional Systems and Signal Processing,* 1994, vol. 5, pp. 75-96.

[7] Diaspro A., Parodi G., Zunino R., "A performance analysis of an associative system for image classification", *Pattern Recognition Letters,* 1993, vol. 14, No. 11, pp. 861-868.

[8] Pagano F., Parodi G., Zunino R., "Parallel implementation of associative memories for image classification", *Parallel Computing,* 1993, Vol.19, No. 6, pp. 667-684.