Handwritten Digit Classification Based on Alpha-Beta Associative Model

Luis Octavio López-Leyva, Cornelio Yáñez-Márquez,
Rolando Flores-Carapia, and Oscar Camacho-Nieto

IPN Centro de Investigación en Computación
Juan de Dios Bátiz s/n esq. Miguel Othón de Mendizábal
Unidad Profesional Adolfo López Mateos
Del. Gustavo A. Madero, México, D.F., México
loctavioll@gmail.com, cyanez@cic.ipn.mx,
rfcarapia@yahoo.com, oscarc@cic.ipn.mx

Abstract. In this paper we present a new model appropriate for pattern recognition tasks. This new model, called $\alpha\beta$ Associative Model, arises when taking theoretical elements from the $\alpha\beta$ associative memories, and they are merged with several new mathematical transforms. When applied to handwritten digits recognition, namely in the MNIST database, the $\alpha\beta$ Associative Model exhibits competitive results against some of the most widely known algorithms currently available in scientific literature.

Keywords: Handwritten digits classification, Alpha-Beta associative model, MNIST database.

1 Introduction

Handwritten digits recognition is an important research area. Several techniques have been used to improve the recognition performance: contours [1], bootstrapping [2], neural networks [3], Singular Value Decomposition (SVD) [4], k-Nearest Neighbor (k-NN), Learning Vector Quantization (LVQ), Support Vector Classifiers (SVC) with Radial Base Function (RBF) kernel [5], Multi-Layer Perceptron (MLP) and LeNet [6], and holographic associative memories [7], among others.

On the other hand, $\alpha\beta$ associative memories have been successfully used in recent years to tackle some pattern recognition problems [8]. In this paper, a new handwritten digits recognition model is introduced, taking theoretical elements from the $\alpha\beta$ associative memories.

The rest of the paper is organized as follows: in section 2 the $\alpha\beta$ associative memories are briefly described. The new model is proposed in section 3, section 4 presents some experimental results, and section 5 includes some conclusions and future work.

2 $\alpha\beta$ Associative Memories

In this section we use the concepts, results and notation introduced in [8]. An associative memory **M** is a system that relates input patterns, and outputs

patterns, as follows: $\mathbf{x} \longrightarrow \boxed{\mathbf{M}} \longrightarrow \mathbf{y}$ with $\mathbf{x}$ and $\mathbf{y}$ the input and output pattern vectors, respectively. Each input vector forms an association with a corresponding output vector. For k integer and positive, the corresponding association will be denoted as (x^k, y^k). Associative memory $\mathbf{M}$ is represented by a matrix whose ij-th component is m_{ij}. Memory $\mathbf{M}$ is generated from an *a priori* finite set of known associations, referred to as the fundamental set of associations. If μ is an index, the fundamental set, whose cardinality is p, will be represented as: $\{ (x^\mu, y^\mu) \mid \mu = 1, 2, \ldots, p \}$. A distorted version of a pattern x^k to be recalled will be denoted as $\tilde{x}^k$. If when feeding a distorted version of x^ϖ with $\varpi = \{1, 2, \ldots, p\}$ to an associative memory $\mathbf{M}$, it happens that the output corresponds exactly to the associated pattern y^ϖ, we say that recall is correct.

The $\alpha\beta$ associative memories are of two kinds and are able to operate in two different modes. The operator α is useful at the learning phase, and the operator β is the basis for the pattern recall phase. The properties within the algebraic operators α and β, allow the $\alpha\beta$ memories to exhibit similar characteristics to the ones inherent to the binary version of the morphological associative memories, which were introduced by Ritter *et al.* [9], in the sense of: learning capacity, type and amount of noise against which the memory is robust, and the sufficient conditions for perfect recall.

2.1   The α and β Operators

The heart of the mathematical tools used in the Alpha-Beta model, are two binary operators designed specifically for these memories. We define the sets $A = \{0, 1\}$ and $B = \{0, 1, 2\}$, then the operators α and β are defined in tabular form:

Table 1. Definition of the Alpha and Beta operators

$\alpha : A \times A \to B$		$\beta : B \times A \to A$	
$x \; y$	$\alpha(x, y)$	$x \; y$	$\beta(x, y)$
0 0	1	0 0	0
0 1	0	0 1	0
1 0	2	1 0	0
1 1	1	1 1	1
		2 0	1
		2 1	1

The sets A and B, the α and β operators, along with the usual $\wedge$ (minimum) and $\vee$ (maximum) operators, form the algebraic system $(A, B, \alpha, \beta, \wedge, \vee)$ which is the mathematical basis for the Alpha-Beta associative models.

The ij-th entry of the matrix $y \oplus x^t$ is: $[y \oplus x^t]_{ij} = \alpha(y_i, x_j)$. If we consider the fundamental set of patterns: $\{ (x^\mu, y^\mu) \mid \mu = 1, 2, \ldots, p \}$ then, the ij-th entry of the matrix $y^\mu \oplus (x^\mu)^t$ is: $\left[y^\mu \oplus (x^\mu)^t \right]_{ij} = \alpha(y_i^\mu, x_j^\mu)$.

2.2   The Learning and Recalling Phases

Because of there are two kinds of $\alpha\beta$ associative memories, $\vee$ and $\wedge$, and if we consider that each one of the kinds is able to operate in two different modes, heteroassociative and autoassociative, we have four different available choices. Due to expediency, we will only discuss here the $\alpha\beta$ autoassociative memories of kind $\vee$. Then, the fundamental set takes the form $\{(x^\mu, x^\mu) \,|\, \mu = 1, 2, \ldots, p\}$. Also, the input and output patterns have the same dimension n, and the memory is a square matrix $\mathbf{V} = [v_{ij}]_{n\times n}$.

Learning Phase
For each $\mu = 1, 2, \ldots, p$, and from each couple (x^μ, x^μ) build the matrix $\left[x^\mu \oplus (x^\mu)^t\right]_{n\times n}$.

Apply the binary $\vee$ operator to the matrices obtained in step 1 to get $\mathbf{V}$ as follows: $\mathbf{V} = \bigvee_{\mu=1}^{p} \left[x^\mu \oplus (x^\mu)^t\right]$.

The ij-th entry is given as: $v_{ij} = \bigvee_{\mu=1}^{p} \alpha\left(x_i^\mu, x_j^\mu\right)$, where $v_{ij} \in B, \forall i \in \{1, 2, \ldots, n\}, \forall j \in \{1, 2, \ldots, n\}$.

Recalling Phase
A pattern x^ϖ, with $\omega \in \{1, 2, ..., p\}$ is presented to the $\alpha\beta$ autoassociative memory of kind $\vee$ and the following operation is done: $\mathbf{V}\Delta_\beta x^\varpi$. The result is a column vector of dimension n, with i-th component given as:

$$(\mathbf{V}\Delta_\beta x^\varpi)_i = \bigwedge_{j=1}^{n} \beta\left(v_{ij}, x_j^\varpi\right)$$

$$(\mathbf{V}\Delta_\beta x^\varpi)_i = \bigwedge_{j=1}^{n} \beta\left\{\left[\bigvee_{j=1}^{n} \alpha\left(x_i^\mu, x_j^\mu\right)\right], x_j^\varpi\right\}$$

3   Our Proposal

Some definitions are necessary before the algorithm of the $\alpha\beta$ Associative Model can be described. Below are presented 6 definitions, which are then used to characterize the two phases of the new model algorithm.

3.1   Definitions

Definition 1. *Let* $A = \{0, 1\}$ *be the binary set, and* $\mathbf{x}$ *and* $\mathbf{y}$ *be two vectors* $\mathbf{x} \in A^n$, $\mathbf{y} \in A^n$, *where* n *is the dimension of vectors* $\mathbf{x}$ *and* $\mathbf{y}$, *and* $n \in \mathbb{Z}^+$. *We define* k_i *as follows:*

$$k_i = \sum_{j=1}^{i} \beta\left(y_j, y_j\right) \ \text{for } i \in \{1, 2, \ldots, n\}$$

Definition 2. *Let* $A = \{0, 1\}$ *be the binary set, and* **x** *and* **y** *be two vectors* $\mathbf{x} \in A^n$, $\mathbf{y} \in A^n$, *where* $n \in \mathbb{Z}^+$ *is the dimension of vectors* **x** *and* **y**, *and* k_n *is a number obtained through definition 1. We define the Restriction of vector* **x** *with respect to vector* **y** *as a new vector* $z \in A^{n-k_n}$, *denoted as* $x|_y$, *through the following algorithm:*

```
for (int i = 1; i < n-k(n); i++) {
  int c = 0;
  while (y[i + c] == 1) {
    c++;
  }
  z[i] = x[i + c];
}
```

Remark 1. The last component of z is z_{n-k_n}.

Definition 3. *Let* $A = \{0, 1\}$ *be the binary set, and* **x**, **y**, *and* **z** *be three vectors* $\mathbf{x} \in A^n$, $\mathbf{y} \in A^n$, $\mathbf{z} \in A^{n-k_n}$, *where* $n \in \mathbb{Z}^+$ *is the dimension of vectors* **x** *and* **y**. *We define the Pseudo-Inverse Restriction of vector* **z** *with respect to vector* **y** *as the operation obtained by the following algorithm:*

```
for (int i = 1; i < n; i++) {
  int c = 0;
  if (y[i] == 0) {
    x[i] = z[i + c];
  } else {
    c++;
    x[i] = 1;
  }
}
```

Definition 4. *Let* $A = \{0, 1\}$ *be the binary set, and* **x** *and* **y** *be two vectors* $\mathbf{x} \in A^n$, $\mathbf{y} \in A^n$, *where* n *is the dimension of vectors* **x** *and* **y**, *and* $n \in \mathbb{Z}^+$. *The* $\tau-$*Transform of* **x** *with respect to* **y** *gives as result a vector with dimension* n, *whose components are calculated as follows:*

$$[\tau(x, y)]_i = x_i - \beta(x_i, y_i)$$

Definition 5. *Let* $A = \{0, 1\}$ *be the binary set, and* **x** *and* **y** *be two vectors* $\mathbf{x} \in A^n$, $\mathbf{y} \in A^n$, *where* n *is the dimension of vectors* **x** *and* **y**, *and* $n \in \mathbb{Z}^+$. *The* $\theta-$*Transform of* **x** *with respect to* **y** *is a vector of dimension* n *given by:*

$$\theta(x, y) = \tau(x, y) + \tau(y, x)$$

Definition 6. *Let* $A = \{0, 1\}$ *be the binary set, let* **x** *be a vector* $\mathbf{x} \in A^n$ *of dimension* $n \in \mathbb{Z}^+$, *and* $p \in \mathbb{Z}^+$, $1 < p \leq 2^n$ *is the cardinality of the fundamental*

set of an $\alpha\beta$ Model. The vector $\mathbf{S}$ is made up of n binary components, which are calculated from the fundamental set as:

$$
S_i =
\begin{cases}
\displaystyle\bigwedge_{k=1}^{p/2} \beta\left(x_i^{2k-1}, x_i^{2k}\right) & \text{if } p \text{ is even} \\[4ex]
\beta\left[\displaystyle\bigwedge_{k=1}^{(p-1)/2} \beta\left(x_i^{2k-1}, x_i^{2k}\right), x_i^p\right] & \text{if } p \text{ is odd}
\end{cases}
$$

3.2 The $\alpha\beta$ Associative Model Algorithm

With these definitions, we are now capable of describing the algorithm.

Learning Phase

1. Calculate the vector $\mathbf{S}$ from the fundamental set, according to definition 6.
2. Obtain the restricted version of all the fundamental patterns $\mathbf{x}^\mu|_{\mathbf{S}}$ and assemble the restricted fundamental set $\{(\mathbf{x}^\mu|_{\mathbf{S}}, \mathbf{x}^\mu|_{\mathbf{S}}) | \mu = 1, 2, \ldots, p\}$.

Recall Phase

1. Let $\tilde{\mathbf{x}}$ be a unknown input vector to the $\alpha\beta$ Model.
2. Obtain the restricted input pattern $\tilde{\mathbf{x}}|_{\mathbf{S}}$ from the input vector $\tilde{\mathbf{x}}$ and the vector $\mathbf{S}$, according to definition 2.
3. Obtain the $\tau-$Transform of the restricted input pattern $\tilde{\mathbf{x}}|_{\mathbf{S}}$ with respect to each of the restricted fundamental patterns, $\tau\left(\tilde{\mathbf{x}}|_{\mathbf{S}}, \mathbf{x}^\mu|_{\mathbf{S}}\right) \forall \mu$, according to definition 4.
4. Obtain the $\tau-$Transform of each of the restricted fundamental patterns with respect to the restricted input pattern $\tilde{\mathbf{x}}|_{\mathbf{S}}$, $\tau\left(\mathbf{x}^\mu|_{\mathbf{S}}, \tilde{\mathbf{x}}|_{\mathbf{S}}\right) \forall \mu$, according to definition 4.
5. Obtain the $\theta-$Transform of the restricted input pattern $\tilde{\mathbf{x}}|_{\mathbf{S}}$ with respect to each of the restricted fundamental patterns, $\theta\left(\tilde{\mathbf{x}}|_{\mathbf{S}}, \mathbf{x}^\mu|_{\mathbf{S}}\right) \forall \mu$, according to definition 5.
6. Find ω such that $\theta\left(\tilde{\mathbf{x}}|_{\mathbf{S}}, \mathbf{x}^\omega|_{\mathbf{S}}\right) = \displaystyle\bigwedge_{\mu=1}^{p} \theta\left(\tilde{\mathbf{x}}|_{\mathbf{S}}, \mathbf{x}^\mu|_{\mathbf{S}}\right), \omega \in \{1, 2, \ldots, n\}$.
7. The restricted fundamental vector corresponding to the restricted input pattern $\tilde{\mathbf{x}}|_{\mathbf{S}}$ is $\mathbf{x}^\omega|_{\mathbf{S}}$.
8. Apply the inverse restriction function to $\mathbf{x}^\omega|_{\mathbf{S}}$, according to definition 3, in order to obtain the final result.

4   Experimental Results

We used the MNIST database of handwritten digits to test the Alpha-Beta Associative Model. This database has a training set of 60,000 examples, and a test set of 10,000 examples. The MNIST database is made up by gray-level images
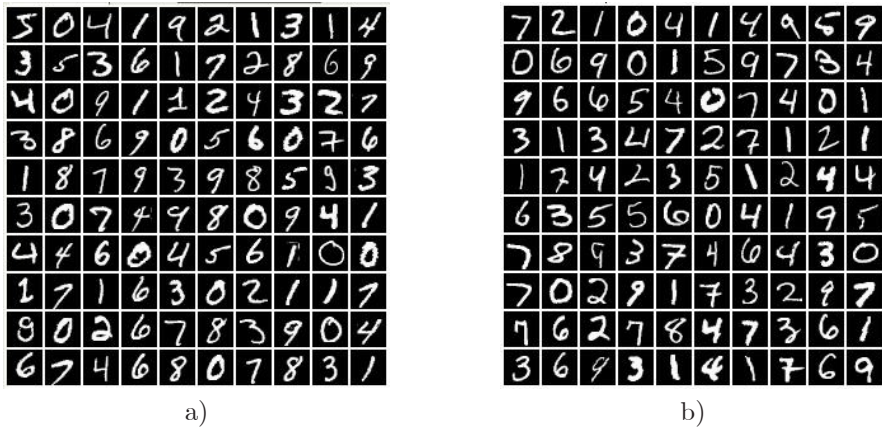
Fig. 1. MNIST database a) first 100 handwritten digits of training set, b) first 100 handwritten digits used for testing

and was constructed from NIST's Special Database 3 and Special Database 1, which contain images of handwritten digits [10].

The algorithm of the $\alpha\beta$ Associative Model was programmed in Borland's C++ Builder 6.0. In the learning phase we use the 60,000 handwritten digits of the training set, and a subset of the testing set for the recalling phase. An example of the patterns used is shown in figure 1.

The result obtained was 99% of handwritten digits correctly recognized. When compared to the results presented in references [11] - [16], and shown in table 2 (along with the proposed algorithm), we can see that the $\alpha\beta$ Associative Model

Table 2. Comparison between various algorithms

Algorithm [Reference]	Year	Preprocessing	Error
k-NN, Euclidean (L2) [11]	1998	none	5.0%
k-NN, Euclidean (L2) [11]	1998	deskewing	2.4%
2-layer NN, 300 HU [11]	1998	deskewing	1.6%
2-layer NN, 800 HU, Cross-Entropy [12]	2003	none	1.6%
2 layer MLP (CE) [12]	2003	none	1.6%
SVM [13]	2002	affine	1.4%
k-NN, Tangent Distance [11]	1998	affine & thick	1.1%
2 layer MLP (CE) [12]	2003	affine	1.1%
k-NN, Tangent Distance [11]	1998	16x16 pixels	1.1%
NN, RBM + NCA training [14]	2007	none	1.0%
$\alpha\beta$ Associative Model [This paper]	**2008**	**none**	**1.0%**
Convolutional net LeNet-5 [11]	1998	none	0.95%
2-layer NN, 800 HU, MSE [12]	2003	elastic distortions	0.9%
large conv. net, random features [15]	2007	none	0.89%
Convolutional net LeNet-5 [11]	1998	huge distortions	0.85%
large conv. net, unsup pretraining [16]	2006	elastic distortions	0.39%

is competitive against other algorithms which present the best classification results on the MNIST database of handwritten digits. It is noteworthy to mention that, while not exhibiting the best result, the presented model does not need to perform any preprocessing besides binarization, unlike many of the other algorithms.

5   Conclusions

We have applied the $\alpha\beta$ Associative Model to handwritten digits recognition, using the MNIST database. This database was created and published by Yann LeCun.

The experimental results obtained in this work, shown that the $\alpha\beta$ Associative Model is competitive against other algorithms which present the best classification results on the MNIST database of handwritten digits; namely, the algorithms reported by Yann LeCun, who apply a preprocessing phase, before carrying on the recognizing task. Besides, most of these algorithms are more computationally expensive than the $\alpha\beta$ Associative Model.

We recommend to others researchers who are interested in this model, the application of the Alpha-Beta Associative Model on other public databases (from UCI, for example).

References

1. Cheng, D., Yan, H.: Recognition of handwritten digits based on contour information. Pattern Recognition 31(3), 235–255 (1998)
2. Saradhi, V.V., Murty, M.N.: Bootstrapping for efficient handwritten digit recognition. Pattern Recognition 34(5), 1047–1056 (2001)
3. Zhang, B., Fu, M., Yan, H.: Nonlinear neural network model of mixture of local principal component analysis: application to handwritten digits recognition. Pattern Recognition 34(2), 203–214 (2001)
4. Savas, B., Eldén, L.: Handwritten digit classification using higher order singular value decomposition. Pattern Recognition 40(3), 993–1003 (2007)
5. Liu, C.-L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: Benchmarking of state-of-the-art techniques. Pattern Recognition 36(10), 2271–2285 (2003)
6. Suen, C.Y., Tan, J.: Analysis of errors of handwritten digits made by a multitude of classifiers. Pattern Recognition Letters 26(3), 369–379 (2005)
7. Mayora-Ibarra, O., Curatelli, F.: Handwritten digit recognition by means of a holographic associative memory. Expert Systems with Applications 15(3-4), 399–403 (1998)

8. Yáñez-Márquez, C.: Memorias Asociativas Basadas en Relaciones de Orden y Operadores Binarios (in Spanish). Ph.D. Thesis. National Polytechnics Institute, Computers Research Center, Mexico (2002)
9. Ritter, G.X., Sussner, P., Diaz-de-Leon, J.L.: Morphological Associative Memories. IEEE Transactions on Neural Networks 9, 281–293 (1998)
10. Yann LeCun, Courant Institute, NYU Corinna Cortes, Google Labs, New York The MNIST database of handwritten digits, `http://yann.lecun.com/exdb/mnist/`
11. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
12. Simard, P.Y., Steinkraus, D., Platt, J.: Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis. In: International Conference on Document Analysis and Recognition (ICDAR), pp. 958–962. IEEE Computer Society, Los Alamitos (2003)
13. Decoste, D., Scholkopf, B.: Training Invariant Support Vector Machines. Machine Learning 46, 161–190 (2002)
14. Salakhutdinov, R.R., Hinton, G.E.: Learning a Nonlinear Embedding by Preserving Class Neighbourhood Structure. In: AI and Statistics, Puerto Rico (2007)
15. Ranzato, M.A., Huang, F.J., Boureau, Y.L., LeCun, Y.: Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition. In: Proc. Computer Vision and Pattern Recognition Conference (CVPR 2007). IEEE Press, Los Alamitos (2007)
16. Ranzato, M.A., Poultney, C., Chopra, S., LeCun, Y.: Efficient Learning of Sparse Representations with an Energy-Based Model. In: Platt, J., et al. (eds.) Advances in Neural Information Processing Systems (NIPS 2006). MIT Press, Cambridge (2006)