Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2009, Article ID 680212, 14 pages
doi:10.1155/2009/680212

Research Article

Classification of Cancer Recurrence with Alpha-Beta BAM

María Elena Acevedo, Marco Antonio Acevedo, and Federico Felipe

Department of Communications and Electronic Engineering, Superior School of Mechanical and Electrical Engineering, Av. IPN s/n, Col. Lindavista, 07738 Mexico City, Mexico

Correspondence should be addressed to María Elena Acevedo, eacevedo@ipn.mx

Received 3 June 2009; Accepted 30 July 2009

Recommended by Carlo Cattani

Bidirectional Associative Memories (BAMs) based on first model proposed by Kosko do not have perfect recall of training set, and their algorithm must iterate until it reaches a stable state. In this work, we use the model of Alpha-Beta BAM to classify automatically cancer recurrence in female patients with a previous breast cancer surgery. Alpha-Beta BAM presents perfect recall of all the training patterns and it has a one-shot algorithm; these advantages make to Alpha-Beta BAM a suitable tool for classification. We use data from Haberman database, and leave-one-out algorithm was applied to analyze the performance of our model as classifier. We obtain a percentage of classification of 99.98%.

1. Introduction

Breast cancer is a preponderant disease in the world and it is death cause of women. The women who have suffered from breast cancer and have overcome it have the risk to suffer a relapse; therefore women have to be monitored after the tumor has been extracted.

The prediction of recurrent cancer in women with previous surgery has high monetary and social costs; as a result, many researchers working in the Artificial Intelligent (AI) topic have been attracted to this problem and they have used many AI tools among others for breast cancer prediction. Some of these works are described as follows.

Many methods of AI have shown better results than the obtained by the experimental methods; for example, in 1997 Burke et al. [1] compared the accuracy of TNM staging system with the accuracy of a multilayer backpropagation Artificial Neural Network (ANN) for predicting the 5-year survival of patients with breast carcinoma. ANN increased the prediction capacity in 10% obtaining the final result of 54%. They used the following parameters: tumor size, number of positive regional lymph nodes, and distant metastasis.

Domingos [2] used a breast cancer database from UCI repository for classifying survival of patients using the unification of two widely used empirical approaches: rule induction and instance-based learning.

In 2000, Boros et al. [3] used the Logical Analysis of Data method to predict the nature of the tumor: malignant or benign. Breast Cancer (Wisconsin) database was used. The classification capacity was 97.2%. This database was used by Street and Kim [4] who combined several classifiers to create a high-scale classifier. Also, it was used by Wang and Witten [5]; they presented a general modeling method for optimal probability prediction over future observations and they obtained the 96.7% of classification.

K. Huang et al. [6] construct a classifier with the Minimax Probability Machine (MPM), which provides a worst-case bound on the probability of misclassification of future data points based on reliable estimates of means and covariance matrices of the classes from the training data points. They used the same database utilized by Domingos. The classification capacity was of 82.5%.

In other types of breast cancer diagnosis, C.-L. Huang et al. [7] employed the Support Vector Machine method to predict a breast tumor from the information of five DNA viruses.

In the last two decades, the impact of breast cancer in Mexico has increased [8]. Every year 3500 women die due to breast cancer, becoming the first death cause and the second frequent type of tumor. Therefore, we applied Associative Models to classify recurrence cancer.

The area of Associative Memories, as a relevant part of Computing Sciences, has acquired great importance and dynamism in the activity developed by international research teams, specifically those who research topics related with theory and applications of pattern recognition and image processing. Classification is a specific homework of pattern recognition because its main goal is to recognize some features of patterns and put these patterns into the corresponding class.

Associative Memories have been developed, at the same time with Neural Networks, from the first model of artificial neuron [9] to neural networks models based on modern concepts such as mathematical morphologic [10] getting through the important works of pioneers in neural networks perceptron-based [11–13].

In 1982 Hopfield presents his associative memory; this model is inspired in physical concepts and has as particularity an iterative algorithm [14]. This work has great relevance because Hopfield proved that interactions of simple processing elements similar to neurons give rise to collective computational properties, such as memory stability.

However, Hopfield model has two disadvantages: firstly, associative memory shows a low recall capacity, $0{,}15n$, where n is the dimensions of stored patterns; secondly, Hopfield memory is autoassociative, which means that it is not able to associate different patterns.

In 1988, Kosko [15] developed a heteroassociative memory from two Hopfield memories to overcome the second disadvantage of Hopfield model. Bidirectional Associative Memory (BAM) is based in an iterative algorithm the same as Hopfield. Many later models were based on this algorithm and they replaced the original learning rule with an exponential rule [16–18]; other models used a multiple training method and dummy addition [19] to achieve more pairs of patterns to be stable states and, at the same time, they eliminated spurious states. Lineal programming techniques [20], gradient descent method [21, 22], genetic algorithms [23], and delayed BAMs [24, 25] had been used with the same purpose. There are many other models which are not based on Kosko, so that they are not iterative and have not stability problems: Morphologic [26] and Feedforward [27] BAM. All these models have appeared to overcome the low-capacity recall problem showed by the first BAM; however, none of them have could recover all training patterns. Besides, these models require the patterns to have certain conditions such as Hamming distance, orthogonality, lineal independence, and lineal programming solutions, among others.

The bidirectional associative memory model used in this work is based on Alpha-Beta Associative Memories [28]; it is not an iterative process and does not have stability problems. Alpha-Beta BAM recall capacity is maximum: $2^{\min(n,m)}$, where n and m are the dimensions of input and output patterns, respectively. This model always shows perfect recall without any condition. Alpha-Beta BAM perfect recall has mathematical bases [29]. It has been demonstrated that this model has a complexity of $\mathrm{O}(n^2)$ (see Section 2.4). Its main application is pattern recognition and it has been applied as translator [30] and fingerprints identifier [31].

Because Alpha-Beta BAM shows perfect recall, it is used as a classifier in this work. We used Haberman database, which contains data from cancer recurrence patients, because it has been included in several works to prove other classification methods such as Support Vector Machines (SVMs) combined with Cholesky Factorization [32], Distance Geometry [33], Bagging technique [34], Model-Averaging with Discrete Bayesian Network [35], in-group and out-group concept [36], and ARTMAP fuzzy neuronal networks [37]. Alpha-Beta BAM pretends to surpass the previous results, doing the observation that none of the aforementioned works have used associative models for classifying.

In Section 2 we present basic concepts of associative models along with the description of Alpha-Beta associative memories and Alpha-Beta BAM and its complexity. Experiments and results are showed in Section 3 along with the analysis of our proposal with leave-one-out method.

2. Alpha-Beta Bidirectional Associative Memories

In this section Alpha-Beta Bidirectional Associative Memory is presented. However, since it is based on the Alpha-Beta autoassociative memories, a summary of this model will be given before presenting our model of BAM.

2.1. Basic Concepts

Basic concepts about associative memories were established three decades ago in [38–40]; nonetheless here we use the concepts, results, and notation introduced in [28]. An associative memory $\mathbf{M}$ is a system that relates input patterns and outputs patterns, as follows: $\mathbf{x} \rightarrow \mathbf{M} \rightarrow \mathbf{y}$ with $\mathbf{x}$ and $\mathbf{y}$ being 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, known as the fundamental set of associations.

If μ is an index, the fundamental set is represented as $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \ldots, p\}$ with p being the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns. If it holds that $x^\mu = y^\mu$, for all $\mu \in \{1, 2, \ldots, p\}$, $\mathbf{M}$ is *autoassociative*; otherwise it is *heteroassociative*; in this case it is possible to establish that $\exists \mu \in \{1, 2, \ldots, p\}$ for which $x^\mu \neq y^\mu$. A distorted version of a pattern x^k to be recuperated will be denoted as $\tilde{x}^k$.

Table 1: Alpha operator. $\alpha : A \times A \rightarrow B$.

x	y	$\alpha(x, y)$
0	0	1
0	1	0
1	0	2
1	1	1

Table 2: Beta operator. $\beta : B \times A \rightarrow A$.

x	y	$\alpha(x, y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

If when feeding a distorted version of x^{ϖ} with $\overline{\varpi} = \{1, 2, \ldots, p\}$ to an associative memory **M**, it happens that the output corresponds exactly to the associated pattern y$\boldsymbol{\varpi}$, we say that recuperation is perfect.

2.2. Alpha-Beta Associative Memories

Among the variety of associative memory models described in the scientific literature, there are two models that, because of their relevance, it is important to emphasize morphological associative memories which were introduced by Ritter et al. [39] and Alpha-Beta associative memories. Because of their excellent characteristics, which allow them to be superior in many aspects to other models for associative memories, morphological associative memories served as starter point for the creation and development of the Alpha-Beta associative memory.

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 heart of the mathematical tools used in the Alpha-Beta model is two binary operators designed specifically for these memories. These operators are defined as follows: first, we define the sets $A = \{0, 1\}$ and $B = \{0, 1, 2\}$, and then the operators α and β are defined in Tables 1 and 2, respectively:

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 memories.

Below are shown some characteristics of Alpha-Beta autoassociative memories.

(1) The fundamental set takes the form $\{(x^{\mu}, x^{\mu}) \mid \mu = 1, 2, \ldots, p\}$.

(2) Both input and output fundamental patterns are of the same dimension, denoted by n.

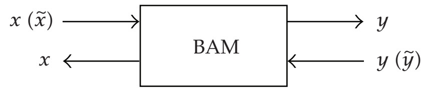

Figure 1: General scheme of a Bidirectional Associative Memory.

(3) The memory is a square matrix, for both modes, $\mathbf{V}$ and $\mathbf{\Lambda}$. If $x^\mu \in A^n$, then

$$\begin{aligned} v_{ij} &= \bigvee_{\mu=1}^{p} \alpha\left(x_i^\mu, x_j^\mu\right), \\ \lambda_{ij} &= \bigwedge_{\mu=1}^{p} \alpha\left(x_i^\mu, x_j^\mu\right), \end{aligned} \tag{2.1}$$

and according to $\alpha : A \times A \to B$, we have that v_{ij} and $\lambda_{ij} \in B$, for all $i \in \{1, 2, \ldots, n\}$ and for all $j \in \{1, 2, \ldots, n\}$.

In recall phase, when a pattern x^μ is presented to memories $\mathbf{V}$ and $\mathbf{\Lambda}$, the ith components of recalled patterns are

$$\begin{aligned} \left(\mathbf{V}\Delta_\beta x^\omega\right)_i &= \bigwedge_{j=1}^{n} \beta\left(v_{ij}, x_j^\omega\right), \\ \left(\mathbf{\Lambda}\nabla_\beta x^\omega\right)_i &= \bigvee_{j=1}^{n} \beta\left(\lambda_{ij}, x_j^\omega\right). \end{aligned} \tag{2.2}$$

2.3. Alpha-Beta BAM

Generally, any bidirectional associative memory model appearing in current scientific literature could be draw as Figure 1 shows.

General BAM is a "black box" operating in the next way: given a pattern x, associated pattern y is obtained, and given the pattern y, associated pattern x is recalled. Besides, if we assume that $\tilde{x}$ and $\tilde{y}$ are noisy versions of x and y, respectively, it is expected that BAM could recover all corresponding free noise patterns x and y.

The model used in this paper has been named Alpha-Beta BAM since Alpha-Beta associative memories, both max and min, play a central role in the model design. However, before going into detail over the processing of an Alpha-Beta BAM, we will define the following.

In this work we will assume that Alpha-Beta associative memories have a fundamental set denoted by $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \ldots, p\}$ $x^\mu \in A^n$ and $y^\mu \in A^m$, with $A = \{0, 1\}$, $n \in \mathbf{Z}^+$, $p \in \mathbf{Z}^+$, $m \in \mathbf{Z}^+$, and $1 < p \leq \min(2^n, 2^m)$. Also, it holds that all input patterns are different; M that is $x^\mu = x^\xi$ if and only if $\mu = \xi$. If for all $\mu \in \{1, 2, \ldots, p\}$ it holds that $x^\mu = y^\mu$, the Alpha-Beta memory will be *autoassociative*; if on the contrary, the former affirmation is negative, that is, $\exists \mu \in \{1, 2, \ldots, p\}$ for which it holds that $x^\mu \neq y^\mu$, then the Alpha-Beta memory will be *heteroassociative*.

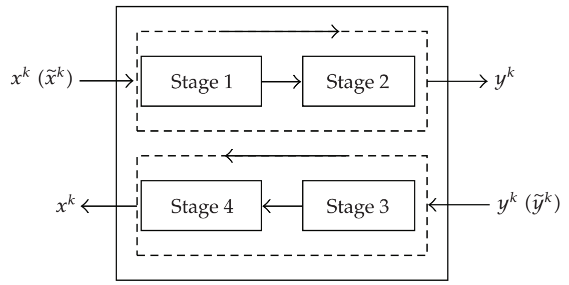

Figure 2: Alpha-Beta BAM model scheme.

Definition 2.1 (One-Hot). Let the set A be $A = \{0, 1\}$ and $p \in \mathbf{Z}^+$, $p > 1$, $k \in \mathbf{Z}^+$, such that $1 \leq k \leq p$. The kth one-hot vector of p bits is defined as vector $h^k \in A^p$ for which it holds that the kth component is $h_k^k = 1$ and the set of the components are $h_j^k = 0$, for all $j \neq k$, $1 \leq j \leq p$.

Remark 2.2. In this definition, the value $p = 1$ is excluded since a one-hot vector of dimension 1, given its essence, has no reason to be.

Definition 2.3 (Zero-Hot). Let the set A be $A = \{0, 1\}$ and $p \in \mathbf{Z}^+$, $p > 1$, $k \in \mathbf{Z}^+$, such that $1 \leq k \leq p$. The kth zero-hot vector of p bits is defined as vector $\overline{h}^k \in A^p$ for which it holds that the kth component is $h_k^k = 0$ and the set of the components are $h_j^k = 1$, $\forall j \neq k$, $1 \leq j \leq p$.

Remark 2.4. In this definition, the value $p = 1$ is excluded since a zero-hot vector of dimension 1, given its essence, has no reason to be.

Definition 2.5 (Expansion vectorial transform). Let the set A be $A = \{0, 1\}$ and $n \in \mathbf{Z}^+$, $y\ m \in \mathbf{Z}^+$. Given two arbitrary vectors $x \in A^n$ and $e \in A^m$, the expansion vectorial transform of order m, $\tau^e : A^n \to A^{n+m}$, is defined as $\tau^e(x, e) = \mathbf{X} \in A^{n+m}$, a vector whose components are $X_i = x_i$ for $1 \leq i \leq n$ and $X_i = e_i$ for $n + 1 \leq i \leq n + m$.

Definition 2.6 (Contraction vectorial transform). Let the set A be $A = \{0, 1\}$ and $n \in \mathbf{Z}^+$, $y\ m \in \mathbf{Z}^+$ such that $1 \leq m < n$. Given one arbitrary vector $\mathbf{X} \in A^{n+m}$, the contraction vectorial transform of order m, $\tau^c : A^{n+m} \to A^m$, is defined as $\tau^c(\mathbf{X}, m) = \mathbf{c} \in A^m$, a vector whose components are $c_i = X_{i+n}$ for $1 \leq i < m$.

In both directions, the model is made up by two stages, as shown in Figure 2.

For simplicity, the first will describe the process necessary in one direction, in order to later present the complementary direction which will give bidirectionality to the model (see Figure 3).

The function of Stage 2 is to offer a y^k as output ($k = 1, \ldots, p$) given an x^k as input.

Now we assume that as input to Stage 2 we have one element of a set of p orthonormal vectors. Recall that the *Linear Associator* has perfect recall when it works with orthonormal vectors. In this work we use a variation of the *Linear Associator* in order to obtain y^k, parting from a *one-hot* vector h^k in its kth coordinate.

For the construction of the modified Linear Associator, its learning phase is skipped and a matrix $\mathbf{M}$ representing the memory is built. Each column in this matrix corresponds to

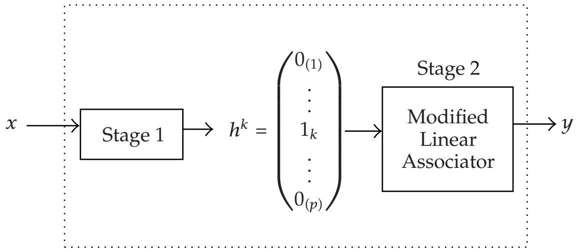

Figure 3: Schematics of the process done in the direction from x to y. Here, only Stage 1 and Stage 2 are shown. Notice that $h_k^k = 1$, $v_i^k = 0$ $\forall i \neq k$, $1 \le i \le p$, $1 \le k \le p$.

each output pattern y^μ. In this way, when matrix $\mathbf{M}$ is operated with a one-hot vector h^k, the corresponding h^k will always be recalled.

The task of Stage 1 is the following: given an x^k or a noisy version of it ($\tilde{x}^k$), the *one-hot* vector h^k must be obtained without ambiguity and with no condition. In its learning phase, Stage 1 has the following algorithm.

(1) For $1 \le \ k \le p$ do expansion: $\mathbf{X}^k = \tau^e(x^k, h^k)$.

(2) For $1 \le i \le n$ and $1 \le j \le n$: $v_{ij} = \vee_{\mu=1}^{p} \alpha(X_i^\mu, X_j^\mu)$.

(3) For $1 \le k \le p$ do expansion: $\overline{\mathbf{X}}^k = \tau^e(x^k, \overline{h}^k)$.

(4) For $1 \le i \le n$ and $1 \le j \le n$,

$$\lambda_{ij} = \bigwedge_{\mu=1}^{p} \alpha\left(\overline{X}_i^\mu, \overline{X}_j^\mu\right). \tag{2.3}$$

(5) Create modified Linear Associator:

$$\mathbf{LAy} = \begin{bmatrix} y_1^1 & y_1^2 & \cdots & y_1^p \\ y_2^1 & y_2^2 & \cdots & y_2^p \\ \vdots & \vdots & \cdots & \vdots \\ y_n^1 & y_n^2 & \cdots & y_n^p \end{bmatrix}. \tag{2.4}$$

Recall phase is described through the following algorithm.

(1) Present, at the input to Stage 1, a vector from the fundamental set $x^\mu \in A^n$, for some index $\mu \in \{1, \ldots, p\}$.

(2) Build vector: $\mathbf{u} = \sum_{i=1}^{p} h^i$.

(3) Do expansion: $\mathbf{F} = \tau^e(x^\mu, u) \in A^{n+p}$.

(4) Obtain vector: $\mathbf{R} = \mathbf{V}\Delta_\beta \mathbf{F} \in A^{n+p}$.

(5) Do contraction: $\mathbf{r} = \tau^c(\mathbf{R}, n) \in A^p$.

If $\mathbf{r}$ is one-hot vector, it is assured that $k = \mu$, then $y^\mu = \mathbf{LAy} \cdot \mathbf{r}$. STOP.

Else:

(6) For $1 \leq i \leq\ p : w_i = u_i - 1$.

(7) Do expansion: $\mathbf{G} = \tau^e(x^\mu, w) \in A^{n+p}$.

(8) Obtain a vector: $\mathbf{S} = \mathbf{\Lambda}\nabla_\beta \mathbf{G} \in A^{n+p}$.

(9) Do contraction: $\mathbf{s} = \tau^c(\mathbf{S}^\mu, n) \in A^p$.

(10) If $\mathbf{s}$ is zero-hot vector, then it is assured that $k = \mu$, $y^\mu = \mathbf{LAy} \cdot \overline{\mathbf{s}}$, where $\overline{\mathbf{s}}$ is the negated vector of $\mathbf{s}$. STOP. Else:

(11) Do operation $\mathbf{r} \wedge \overline{\mathbf{s}}$, where $\wedge$ is the symbol of the logical AND operator, so $y^\mu = \mathbf{LAy} \cdot (\mathbf{r} \wedge \overline{\mathbf{s}})$. STOP.

The process in the contrary direction, which is presenting pattern $y^k\ (k = 1, \ldots, p)$ as input to the Alpha/Beta BAM and obtaining its corresponding x^k, is very similar to the one described above. The task of Stage 3 is to obtain a one-hot vector h^k given a y^k. Stage 4 is a modified Linear Associator built in similar fashion to the one in Stage 2.

2.4. The Alpha-Beta BAM Algorithm Complexity

An algorithm is a finite set of precise instructions for the realization of a calculation or to solve a problem [41]. In general, it is accepted that an algorithm provides a satisfactory solution when it produces a correct answer and is efficient. One measure of efficiency is the time required by the computer in order to solve a problem using a given algorithm. A second measure of efficiency is the amount of memory required to implement the algorithm when the input data are of a given size.

The analysis of the time required to solve a problem of a particular size implies finding the *time complexity* of the algorithm. The analysis of the memory needed by the computer implies finding the *space complexity* of the algorithm.

Space Complexity

In order to store the px patterns, a matrix is needed. This matrix will have dimensions p x$(n+p)$. Input patterns and the added vectors, both *one-hot* and *zero-hot*, are stored in the same matrix. Since $x \in \{0,1\}$, then this values can be represented by character variables, taking 1 byte each. The total amount of bytes will be Bytes_x $= p(n+p)$.

A matrix is needed to store the py patterns. This matrix will have dimensions $p \cdot (m+p)$. Output patterns and the added vectors, both *one-hot* and *zero-hot*, are stored in the same matrix. Since $y \in \{0,1\}$, then this values can be represented by character variables, taking 1 byte each. The total amount of bytes will be Bytes_y $= p(m+p)$.

During the learning phase, 4 matrices are needed: two for the Alpha-Beta autoassociative memories of type max, Vx and Vy, and two more for the Alpha-Beta autoassociative memories of type min, Λx y Λy. Vx and Λx have dimensions of $(n+p)$x$(n+p)$, while Vy and Λy have dimensions $(m+p)$x$(m+p)$. Given that these matrices hold only positive integer numbers, then the values of their components can be represented with character variables of 1 byte of size. The total amount of bytes will be Bytes_VxΛx $= 2(n+p)^2$ and Bytes_VyΛy $= 2(m+p)^2$.

A vector is used to hold the recalled *one-hot* vector, whose dimension is p. Since the components of any one-hot vector take the values of 0 and 1, these values can be represented by character variables, occupying 1 byte each. The total amount of bytes will be Bytes_vr $= p$.

The total amount of bytes required to implement an Alpha-Beta BAM is

$$\begin{aligned}&\text{Total} = \text{Bytes\_x} + \text{Bytes\_y} + \text{Bytes\_Vx}\Lambda\text{x} + \text{Bytes\_Vy}\Lambda\text{y} + \text{Bytes\_vr}\\&\text{Total} = p(n + m + 2p) + 2\left[(n + p)^2 + (m + p)^2\right] + p.\end{aligned} \tag{2.5}$$

Time Complexity

The time complexity of an algorithm can be expressed in terms of the number of operations used by the algorithm when the input has a particular size. The operations used to measure time complexity can be integer compare, integer addition, integer division, variable assignation, logical comparison, or any other elemental operation.

The following is defined: EO: elemental operation; n_pares: number of associated pairs of patterns; n: dimension of the patterns plus the addition of the *one-hot* or *zero-hot* vectors.

The recalling phase algorithm will be analyzed, since this is the portion of the whole algorithm that requires a greater number of elemental operations.

```
Recalling Phase
       u = 0; (1)
      while(u<n_pares) (2)
              i = 0; (3)
    while(i<n) (4)
                      j = 0; (5)
      while(j<n) (6)
       if(y[u][i]==0 && y[u][j]==0) (7)
                                                        t=1; (8)
       else if(y[u][i]==0 && y[u][j]==1) (9a)
                                                        t=0;
       else if(y[u][i]==1 && y[u][j]==0) (9b)
                                                        t=2;
       else              t=1;
       if(u==0) (10)
         Vy[i][j]=t; (11)
       else
         if(Vy[i][j]<t) (12)
            Vy[i][j]=t; (13)
       j++; (14)
i++; (15)
        u++; (16)
```

(1) 1 EO, assignation

(2) n_pares EO, comparison

(3) n_pares EO, assignation

(4) n_pares*n EO, comparison

(5) n_pares*n EO, assignation

(6) n_pares*n*n EO, comparison

(7a) n_pares*n*n EO, comparison: y[u][i]==0

(7b) n_pares*n*n EO, relational operation AND: &&

(7c) n_pares*n*n EO, comparison: y[u][j]==0

(8) There is allways an allocation to variable t, n_pares*n*n EO

(9) Both if sentences (a and b) have the same probability of being executed, n_pares*n*$(n/2)$

(10) n_pares*n*n EO, comparison

(11) This allocation is done only once, 1 EO

(12) (n_pares*n*n)-1 EO, comparison

(13) Allocation has half probability of being run, n_pares*n*$(n/2)$

(14) n_pares*n*n EO, increment

(15) n_pares*n EO, increment

(16) n_pares EO, increment

The total number of EOs is Total = 1 + n_pares$(3 + 3n + 9n^2)$.

From the total of EOs obtained, n_pares is fixed with value 50, resulting in a function only dependant on the size of the patterns: $f(n) = 1 + 50(3 + 3n + 9n^2)$.

In order to analyze the feasibility of the algorithm we need to understand how fast the mentioned function grows as the value of n rises. Therefore, the Big-O notation [33], shown below, will be used.

Let f and g be functions from a set of integer or real numbers to a set of real numbers. It is said that $f(x)$ is $O(g(x))$ if there exist two constants C and k such that

$$|f(x)| \leq C\ |g(x)| \quad \text{when } x > k. \tag{2.6}$$

The number of elemental operations obtained from our algorithm was

$$f(n) = 1 + 50\left(3 + 3n + 9n^2\right). \tag{2.7}$$

A function $g(x)$ and constants C and k must be found, such that the inequality holds. We propose $50(3n^2 + 3n^2 + 9n^2) = 150n^2 + 150n^2 + 450n^2 = 750n^2$.

Then if $g(n) = n^2$, $C = 750$ and $k = 1$, we have that

$$|f(n)| \leq 750\ |g(n)| \quad \text{when } n > 1, \text{therefore } O\left(n^2\right). \tag{2.8}$$

3. Experiments and Results

The database used in this work for Alpha-Beta BAM performance analysis as classifier was proposed by Heberman and it is available in [42]. This database has 306 instances with 3 attributes, which are (1) age of patient at time of operation, (2) patient's year of operation, and (3) number of positive axillary nodes detected. Database has a survival status (class attribute): (1) the patient survived 5 years or longer and (2) the patient died within five year.

Table 3: Results of the classifications with different methods using Haberman database.

Method	Classification (%)	Reference
SVM-Bagging	74	[26]
Model-Averaging	90	[27]
In-group out-group	77.17	[28]
Fuzzy ARTMAP	75	[29]
Alpha-Beta BAM	99.98	This work

The number of instances was reduced at 287 due to some records appeared as duplicated or in some cases records were associated with a same class. From the 287 records, 209 belonged to class 1 and the 78 remainder belonged to class 2.

Implementation of Alpha-Beta BAM was accomplished on a Sony VAIO laptop with Centrino Duo processor and language programming was Visual C++ 6.0.

Leave-one-out method [43] was used to carry out the performance analysis of Alpha-Beta BAM classification. This method operates as follows: a sample is removed from the total set of samples and these 286 samples are used as the fundamental set; therefore, we used the samples to create the BAM. Once Alpha-Beta BAM learnt, we proceeded to classify the 286 samples along with the removed sample, and this means that we presented to the BAM every sample belonging to fundamental set as well as the removed sample.

The process was repeated 287 times, which corresponds to the number of records. Alpha-Beta BAM had the following behavior: in 278 times, Alpha-Beta BAM classified in perfect way the excluded sample and in the 9 remainder probes it did not achieve to classify correctly. Here, it must be emphasized that incorrect classification appears just with the excluded sample, because in all probes belonging to fundamental set, Alpha-Beta BAM shows perfect recall. Therefore, in 278 times the classification percentage was of 100% and 99.65% in the remainder. Calculating the average of classification from the 287 probes, we observed that Alpha-Beta BAM classification was of 99.98%.

In Table 3 there can be observed results comparisons of some classification methods such as SVM-Bagging, Model-Averaging, in-group/out-group method, fuzzy ARTMAP neural network, and Alpha-Beta BAM. Methods presented in [24, 25] do not show classification results and they just indicate that their algorithms are used to accelerate the method performance.

Alpha-Beta BAM exceeds the other methods by a 9.98% and none of these algorithms use an associative model.

We must mention that Haberman database has records very similar to each other, and this feature could complicate the performance of some BAMs, due to the restriction respecting to the data characteristics, for example, Hamming distance or orthogonality. However, Alpha-Beta BAM does not present these kinds of data limitations and we had proved it with the obtained results.

4. Conclusions

The use of bidirectional associative memories as classifiers using Haberman database has not been reported before. In this work we use the model of Alpha-Beta BAM to classify cancer recurrence.

Our model present perfect recall of the fundamental set in contrast with Kosko-based models or morphological BAM; this feature makes Alpha-Beta BAM the suitable tool for pattern recognition and, particularly, for classification.

We compared our results with the following methods: SVM-Bagging, Model-Averaging, in-group/out-group method, and fuzzy ARTMAP neural network, and we found that Alpha-Beta BAM is the best classifier when Haberman database was used, because the classification percentage was of 99.98% and exceeds the other methods by a 9.98%.

With these results we can prove that Alpha-Beta BAM not just has perfect recall but also can recall the most of records not belonging to training patterns.

Even though patterns are very similar to each other, Alpha-Beta BAM was able to recall many of the data, so that it could perform as a great classifier. Most of Kosko-based BAMs have low recalling when patterns show features as Hamming distance, orthogonality and linear independence; however, Alpha-Beta BAM does not impose any restriction in the nature of data.

The next step in our research is to test Alpha-Beta BAM as classifier using other databases as Breast Cancer (Wisconsin) and Breast Cancer (Yugoslavia) and with standard databases as Iris Plant or MNIST; therefore we can obtain the general performance of our model. However, we have to take into account the "no free lunch" theorem which asserts that any algorithm could be the best in one type of problems but it can be the worst in other types of problems. In our case, our results showed that Alpha-Beta BAM is the best classifier when Haberman database was used.

Acknowledgments

The authors would like to thank the Instituto Politécnico Nacional (COFAA and SIP) and SNI for their economical support to develop this work.

References

[1] H. B. Burke, P. H. Goodman, D. B. Rosen, et al., "Artificial neural networks improve the accuracy of cancer survival prediction," *Cancer*, vol. 79, no. 4, pp. 857–862, 1997.

[2] P. Domingos, "Unifying instance-based and rule-based induction," *Machine Learning*, vol. 24, no. 2, pp. 141–168, 1996.

[3] E. Boros, P. Hammer, and T. Ibaraki, "An implementation of logical analysis of data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 2, pp. 292–306, 2000.

[4] W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," in *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)*, pp. 377–382, ACM, San Francisco, Calif, USA, August 2001.

[5] Y. Wang and I. H Witten, "Modeling for optimal probability prediction," in *Proceedings of the 9th International Conference on Machine Learning (ICML '02)*, pp. 650–657, July 2002.

[6] K. Huang, H. Yang, and I. King, "Biased minimax probability machine for medical diagnosis," in *Proceedings of the 8th International Symposium on Artificial Intelligence and Mathematics (AIM '04)*, Fort Lauderdale, Fla, USA, January 2004.

[7] C.-L. Huang, H.-C. Liao, and M.-C. Chen, "Prediction model building and feature selection with support vector machines in breast cancer diagnosis," *Expert Systems with Applications*, vol. 34, no. 1, pp. 578–587, 2008.

[8] O. López-Ríos, E. C. Lazcano-Ponce, V. Tovar-Guzmán, and M. Hernández-Avila, "La epidemia de cáncer de mama en México. Consecuencia de la transición demográfica?" *Salud Publica de Mexico*, vol. 39, no. 4, pp. 259–265, 1997.

[9] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[10] G. X. Ritter and P. Sussner, "An introduction to morphological neural networks," in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 4, pp. 709–717, 1996.

[11] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[12] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.

[13] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[14] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, no. 8, pp. 2554–2558, 1982.

[15] B. Kosko, "Bidirectional associative memories," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 49–60, 1988.

[16] Y. Jeng, C. C Yeh, and T. D. Chiveh, "Exponential bidirectional associative memories," *Electronics Letters*, vol. 26, no. 11, pp. 717–718, 1990.

[17] W.-J. Wang and D.-L. Lee, "Modified exponential bidirectional associative memories," *Electronics Letters*, vol. 28, no. 9, pp. 888–890, 1992.

[18] S. Chen, H. Gao, and W. Yan, "Improved exponential bidirectional associative memory," *Electronics Letters*, vol. 33, no. 3, pp. 223–224, 1997.

[19] Y. F. Wang, J. B. Cruz Jr., and J. H. Mulligan Jr., "Two coding strategies for bidirectional associative memory," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 81–92, 1990.

[20] Y. F. Wang, J. B. Cruz Jr., and J. H. Mulligan Jr., "Guaranteed recall of all training pairs for bidirectional associative memory," *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 559–567, 1991.

[21] R. Perfetti, "Optimal gradient descent learning for bidirectional associative memories," *Electronics Letters*, vol. 29, no. 17, pp. 1556–1557, 1993.

[22] G. Zheng, S. N. Givigi, and W. Zheng, "A new strategy for designing bidirectional associative memories," in *Advances in Neural Networks*, vol. 3496 of *Lecture Notes in Computer Science*, pp. 398–403, Springer, Berlin, Germany, 2005.

[23] D. Shen and J. B. Cruz Jr., "Encoding strategy for maximum noise tolerance bidirectional associative memory," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 293–300, 2005.

[24] S. Arik, "Global asymptotic stability analysis of bidirectional associative memory neural networks with time delays," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 580–586, 2005.

[25] J. H. Park, "Robust stability of bidirectional associative memory neural networks with time delays," *Physics Letters A*, vol. 349, no. 6, pp. 494–499, 2006.

[26] G. X. Ritter, J. L. Diaz-de-Leon, and P. Sussner, "Morphological bidirectional associative memories," *Neural Networks*, vol. 12, no. 6, pp. 851–867, 1999.

[27] Y. Wu and D. A. Pados, "A feedforward bidirectional associative memory," *IEEE Transactions on Neural Networks*, vol. 11, no. 4, pp. 859–866, 2000.

[28] C. Yáñez-Márquez, *Associative memories based on order relations and binary operators*, Ph.D. thesis, Center for Computing Research, Mexico City, Mexico, 2002.

[29] M. E. Acevedo-Mosqueda, C. Yáñez-Márquez, and I. López-Yáñez, "Alpha-beta bidirectional associative memories: theory and applications," *Neural Processing Letters*, vol. 26, no. 1, pp. 1–40, 2007.

[30] M. E. Acevedo-Mosqueda, C. Yáñez-Márquez, and I. López-Yáñez, "Alpha-beta bidirectional associative memories based translator," *International Journal of Computer Science and Network Security*, vol. 6, no. 5A, pp. 190–194, 2006.

[31] M. E. Acevedo-Mosqueda, C. Yáñez-Márquez, and I. López-Yáñez, "Alpha-beta bidirectional associative memories," *International Journal of Computational Intelligence Research*, vol. 3, no. 1, pp. 105–110, 2007.

[32] D. DeCoste, "Anytime query-tuned kernel machines via cholesky factorization," in *Proceedings of the SIAM International Conference on Data Mining (SDM '03)*, 2003.

[33] D. DeCoste, "Anytime interval-value outputs for kernel machines: fast support vector machine classification via distance geometry," in *Proceedings of the International Conference on Machine Learning (ICML '02)*, 2002.

[34] Y. Zhang and W. N. Street, "Bagging with adaptive costs," in *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM '05)*, pp. 825–828, Houston, Tex, USA, November 2005.

[35] D. Dash and G. F. Cooper, *Model-Averaging with Discrete Bayesian Network Classifiers*, Cambridge, UK.

[36] A. Thammano and J. Moolwong, "Classification algorithm based on human social behavior," in *Proceedings of the 7th IEEE International Conference on Computer and Information Technology (CIT '07)*, pp. 105–109, Aizuwakamatsu, Japan, October 2007.

[37] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 698–713, 1992.

[38] T. Kohonen, "Correlation matrix memories," *IEEE Transactions on Computers*, vol. 21, no. 4, pp. 353–359, 1972.

[39] G. X. Ritter, P. Sussner, and J. L. Diaz-de-Leon, "Morphological associative memories," *IEEE Transactions on Neural Networks*, vol. 9, pp. 281–293, 1998.

[40] C. Yáñez-Márquez and J. L. Díaz de León, "Memorias asociativas basadas en relaciones de orden y operaciones binarias," *Computación y Sistemas*, vol. 6, no. 4, pp. 300–311, 2003.

[41] K. Rosen, *Discrete Mathematics and Its Applications*, McGraw-Hill, Estados Unidos, Brazil, 1999.

[42] A. Asuncion and D. J. Newman, "UCI machine learning repository," University of California, School of Information and Computer Science, Irvine, Calif, USA, 2007, http://archive.ics.uci.edu/ml.

[43] A. R. Webb, *Statistical Pattern Recognition*, John Wiley & Sons, West Sussex, UK, 2002.