Searching Algorithms of the Optimal Radix of Exponential Bidirectional Associative Memory *

Chua-Chin Wang and Jyh-Ping Lee
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424

Abstract

Due to the limitation of the dynamic range of VLSI circuits in the implementation of the exponential bidirectional associative memory (eBAM), the optimal radix must be found in order to get the maximal dimension for the eBAM. It is also necessary to update the radix when new pattern pairs are to be stored. An optimal updating algorithm for the radix is also present. Besides, a pattern pair training algorithm is proposed to ensure every trained pair stored in the eBAM will be recalled when an input pair is located in its basin with radius r.

1 Introduction

After Kosko [5] proposed the *bidirectional associative memory* (BAM), many researchers threw efforts on improving its intrinsic poor capacity and implementing the BAM with hardware circuits. However, most of their work can not provide significant improvement for the capacity of the BAM [7]. Chiueh and Goodman [1] proposed an exponential Hopfield associative memory motivated by the MOS transistor's exponential drain current dependence on the gate voltage in the subthreshold region such that the VLSI implementation of an exponential function is feasible. Chiueh also proposed an exponential correlation associative memory (ECAM) which is an autocorrelator utilizing the mentioned exponential function of VLSI circuits to enlarge the correlation between stored pattern pairs. Based upon the concept of Chiueh's exponential Hopfield associative memory, Jeng *et al.* proposed one kind of exponential BAM [4]. However, the energy function proposed in [4] can not guarantee that every stored pattern pair will have a local minimum on the energy surface. Moreover, there is no capacity analysis given in [4].

Since Chiueh [2], Glasser [3], and Mead [6] all pointed out the dynamic range of the VLSI exponential circuits operating in the subthreshold region is approximately fixed, this property leads two important limitations in the implementation of an eBAM. The first limitation is the dimension, i.e., the length of a stored pattern of the stored patterns might be limited. The second limitation is that the minimal radix of the exponential circuit must be estimated in order to get the maximal dimension of stored patterns. These limitations raise the problem of how to find the optimal radix of the eBAM when certain pattern pairs are given, and the problem of how to update the radix when new pattern pairs are to be stored.

In this paper, we again adopt the weighted learning method [8] to ensure the optimal radix is found and updated. In addition, a pattern pair training algorithm is also present to train the eBAM with pattern pairs so that every trained pair are guaranteed to be recalled when an input pair is resided in its own basin.

2 Framework of Exponential BAM

Before proceeding the discussion of the searching algorithm, we have to restate the framework of the eBAM neural network as a background knowledge [7].

2.1 Evolution equations

Suppose we are given M training sample pairs, which are $\{(A_1, B_1), (A_2, B_2), ..., (A_M, B_M)\}$, where $A_i = (a_{i1}, a_{i2}, ..., a_{in})$, $B_i = (b_{i1}, b_{i2}, ..., b_{ip})$. Let X_i and Y_i be the bipolar mode of the training pattern

*This research was partially supported by National Science Council under grant NSC 82-0113-E-110-092-T.

pairs, A_i and B_i, respectively. That is, $X_i \in \{-1,1\}^n$ and $Y_i \in \{-1,1\}^p$. Instead of using Kosko's approach [5], we use the following evolution equations in the recall process of the eBAM :

$$y_k = \begin{cases} 1, & \text{if } \sum_{i=1}^{M} y_{ik} b^{X_i \cdot X} \geq 0 \\ -1, & \text{if } \sum_{i=1}^{M} y_{ik} b^{X_i \cdot X} < 0 \end{cases}, \quad x_k = \begin{cases} 1, & \text{if } \sum_{i=1}^{M} x_{ik} b^{Y_i \cdot Y} \geq 0 \\ -1, & \text{if } \sum_{i=1}^{M} x_{ik} b^{Y_i \cdot Y} < 0 \end{cases} \tag{1}$$

where b is a positive number, $b > 1$, "$\cdot$" represents the inner product operator, x_k and x_{ik} are the kth bits of X and the X_i, respectively, and y_k and y_{ik} are for Y and the Y_i, respectively. The reasons for using an exponential scheme are to enlarge the attraction radius of every stored pattern pair and to augment the desired pattern in the recall reverberation process.

We adopt the SNR approach to compute the capacity of the exponential BAM,

$$SNR_{eBAM} = \frac{2^{n-1} b^4}{2(M-1)(1+b^{-4})^{n-1}} \tag{2}$$

where n is assumed to the $\min(n,p)$ without any loss of generality.

2.2 Dynamic Range Limitation

Chiueh [2], Glasser [3], and Mead [6] pointed out that a transistor operating in the subthreshold region working as an exponentiation circuit has a dynamic range of approximate 10^5 to 10^7. In the realization of eBAM by VLSI circuit, hence, the dimension, n, and subsequently the amount of stored pattern pairs, M, must also be limited. Assume the dynamic range D to be $D = b^n$. When the dimension of the eBAM, n, increases, then b will decrease. The capacity of the eBAM, M, is approximately exponentially proportional to the dimension of the eBAM according to Eqn.(2). Therefore, the radix b must be optimally small.

2.3 Optimal Radix Searching & Updating Algorithms

2.3.1 Optimal radix searching algorithm

Considering the dynamic range limitation, we have to study an approach to find the minimal radix, b, in order to increase the dimension and the capacity of the eBAM given certain pattern pairs to be stored. A modification of Wang's weighted learning algorithm [8] leads to a optimal radix searching algorithm.

According to Eqn.(1), the following two equalities should hold.

$$F_{x_{jk}} = x_{jk} \sum_{i=1}^{M} x_{ik} \cdot b^{Y_i \cdot Y_j} > 0, \quad F_{y_{jk}} = y_{jk} \sum_{i=1}^{M} y_{ik} \cdot b^{X_i \cdot X_j} > 0 \tag{3}$$

The above two equalities can also be deemed as the sufficient conditions for a pattern pair, (X_j, Y_j), to reside in a stable state. In order to find an optimal radix for the eBAM, a cost function is defined, which is the function of the radix, b.

$$J(b) = -\sum_{j=1}^{M} \sum_{k=1}^{n} F_{x_{jk}} \cdot H(F_{x_{jk}}) - \sum_{j=1}^{M} \sum_{k=1}^{p} F_{y_{jk}} \cdot H(F_{y_{jk}}) \tag{4}$$

where $H(t) = 0$ if $t > 0$ and $H(t) = 1$ if $t \leq 0$. This cost function has several properties to be a good measure of the radix.

1). If all of the F_x and F_y are greater than 0, then $J(b) = 0$. That is the optimal b is found.

2). If any of $F_{x_{jk}}$ or $F_{y_{jk}}$ is smaller than 0, then $J(b)$ must be greater than 0.

3). The mentioned two properties imply a gradient descent approach is suitable to be used in order to reach $J(b) = 0$.

Hence, the algorithm of searching the optimal radix is summarized as

$$\begin{aligned}
b(0) &= 1.0 \\
b(t+1) &= b(t) - q \cdot \frac{\partial J(b)}{\partial b} \qquad (5)\\
\frac{\partial J(b)}{\partial b} &= -\sum_{j=1}^{M}\sum_{k=1}^{n}\sum_{i=1}^{M} x_{jk}x_{ik} \cdot \left(\sum_{s=1}^{p} y_{is}y_{js}\right) \cdot b^{Y_i \cdot Y_j - 1} \cdot H(F_{x_{jk}}) \\
&\quad -\sum_{j=1}^{M}\sum_{k=1}^{p}\sum_{i=1}^{M} y_{jk}y_{ik} \cdot \left(\sum_{s=1}^{n} x_{is}x_{js}\right) \cdot b^{X_i \cdot X_j - 1} \cdot H(F_{y_{jk}}) \\
&= -\sum_{j=1}^{M}\sum_{k=1}^{n}\sum_{i=1}^{M} x_{jk}x_{ik} \cdot (Y_i \cdot Y_j) \cdot b^{Y_i \cdot Y_j - 1} \cdot H(F_{x_{jk}}) \\
&\quad -\sum_{j=1}^{M}\sum_{k=1}^{p}\sum_{i=1}^{M} y_{jk}y_{ik} \cdot (X_i \cdot X_j) \cdot b^{X_i \cdot X_j - 1} \cdot H(F_{y_{jk}}) \qquad (6)
\end{aligned}$$

2.3.2 Optimal radix updating algorithm

In Section 2.3.2, we only discuss how to find an optimal b right before pattern pairs being stored in the eBAM. If there are more pattern pairs to be stored after some pattern pairs have been installed, the original radix won't be an optimal radix any more. Hence, there are two methods to solve this problem. First, all of the pattern pairs, including stored pairs and pairs to be stored, are taken into the algorithm in Section 2.3.2 to develop a new b. This method has a big overhead. That is, all of the stored pattern pairs have to be extracted from the eBAM before searching another optimal b. Hence, we propose a method to update the radix only basing upon a given additional pattern pair.

Assume there are already M pattern pairs stored in an eBAM with a radix b_0 which make all of the pairs stay at their respective table state. Then, we also have the following equation according to Eqn.(4),

$$J_0(b) = 0, \quad \forall\, b \geq b_0 \qquad (7)$$

where $J_0(b)$ is the cost function of the eBAM with only the original M pairs of patterns. If there are more pairs to be encoded in this eBAM and every pair is still in its own stable state, a sufficient condition to satisfy this demand is the new radix must be larger than the original radix according to Eqn.(2). Assume (X_h, Y_h) is the $(M+1)$th pair to be stored in the eBAM which has already stored M pattern pairs. Then the overall cost function of the eBAM after the $(M+1)$th pair is stored is formulated as

$$J'(b) = -\sum_{j=1}^{M+1}\sum_{k=1}^{n} F'_{x_{jk}} \cdot H(F'_{x_{jk}}) - \sum_{j=1}^{M+1}\sum_{k=1}^{p} F'_{y_{jk}} \cdot H(F'_{y_{jk}}) \qquad (8)$$

where

$$F'_{x_{jk}} = x_{jk} \sum_{i=1}^{M+1} x_{ik} \cdot b^{Y_i \cdot Y_j}, \quad F'_{y_{jk}} = y_{jk} \sum_{i=1}^{M+1} y_{ik} \cdot b^{X_i \cdot X_j} \qquad (9)$$

If the $H(\cdot)$ terms are 0 given $b = b_0$, then the original b_0 is good enough to store and recall the additional pair. Thus, the algorithm stops. On the other hand, the $H(\cdot)$ terms are 1, which can be crossed out from the new cost function for the sake of clarity. Hence, the new cost function can be rewritten as

$$\begin{aligned}
J'(b) &= -\sum_{j=1}^{M+1}\sum_{k=1}^{n} F'_{x_{jk}} - \sum_{j=1}^{M+1}\sum_{k=1}^{p} F'_{y_{jk}} \\
&= -\sum_{k=1}^{n}\left(\sum_{j=1}^{M} F'_{x_{jk}} + F'_{x_{hk}}\right) - \sum_{k=1}^{p}\left(\sum_{j=1}^{M} F'_{y_{jk}} + F'_{y_{hk}}\right)
\end{aligned}$$

$$
\begin{aligned}
&= -\sum_{k=1}^{n}\left(\sum_{j=1}^{M} F_{x_{jk}} + b^p + 2\sum_{i=1}^{M} x_{ik}x_{hk}\cdot b^{Y_h\cdot Y_i}\right) - \sum_{k=1}^{p}\left(\sum_{j=1}^{M} F_{y_{jk}} + b^n + 2\sum_{i=1}^{M} y_{ik}y_{hk}\cdot b^{X_h\cdot X_i}\right) \\
&= \left\{-\sum_{k=1}^{n}\sum_{j=1}^{M} F_{x_{jk}} - \sum_{k=1}^{p}\sum_{j=1}^{M} F_{y_{jk}}\right\} \\
&\quad + \left\{-n\cdot b^p - 2\sum_{k=1}^{n}\sum_{i=1}^{M} x_{hk}\cdot x_{ik}\cdot b^{Y_h\cdot Y_i} - p\cdot b^n - 2\sum_{k=1}^{p}\sum_{i=1}^{M} y_{hk}\cdot y_{ik}\cdot b^{X_h\cdot X_i}\right\} \\
&= J_0(b) + J_\triangle(b) \qquad (10)
\end{aligned}
$$

Since we have assumed all of the pattern pairs in the eBAM before the installation of the extra pair are recallable, then $J_0(b)$ is 0. Besides, any new radix b must satisfy the condition given in Eqn.(7) in order to make all of the $M+1$ pairs stay in stable states. Thus, the $J_\triangle(b)$ can be deemed as the legitimate new cost function of the eBAM.

$$
J_\triangle(b) = -n\cdot b^p - 2\sum_{k=1}^{n}\sum_{i=1}^{M} x_{hk}\cdot x_{ik}\cdot b^{Y_h\cdot Y_i} - p\cdot b^n - 2\sum_{k=1}^{p}\sum_{i=1}^{M} y_{hk}\cdot y_{ik}\cdot b^{X_h\cdot X_i} \qquad (11)
$$

Again, a gradient descent approach is used to find the optimal b for the ebAM.

$$
\begin{aligned}
b(0) &= b_0 \\
b(t+1) &= b(t) - q\cdot\frac{\partial J_\triangle(b)}{\partial b} \\
\frac{\partial J_\triangle(b)}{\partial b} &= -np\cdot b^{p-1} - 2\sum_{k=1}^{n}\sum_{i=1}^{M} x_{hk}\cdot x_{ik}\cdot (Y_h\cdot Y_i)\cdot b^{Y_h\cdot Y_i - 1} \\
&\quad -np\cdot b^{n-1} - 2\sum_{k=1}^{p}\sum_{i=1}^{M} y_{hk}\cdot y_{ik}\cdot (X_h\cdot X_i)\cdot b^{X_h\cdot X_i - 1} \qquad (12)
\end{aligned}
$$

Note that the above algorithm stops when the overall cost given in Eqn.(8) is 0, i.e., $J'(b) = 0$. The final b is the updated radix for all of the $M+1$ pattern pairs.

2.3.3 Guarantee recall training algorithm

In the previous sections, we are literally based on a statistic approach to analyze the practical capacity of the eBAM. In the following, we are going to discuss an algorithm which will select and train the eBAM with pattern pairs to make sure all of the store pairs guaranteed to be recalled in their own basins, respectively. The object is to make sure every stored pattern pair guaranteed to be recalled if given an input pattern located in its basin.

Assume an eBAM is given M pattern pairs, $(X_1, Y_1), \ldots, (X_M, Y_M)$. Then, a training algorithm of the eBAM can be tabulated as follows.

Step 0. Let $b(0)$ be the initial value of the radix. $b(0)$ is computed by the searching algorithm of Section 2.3.3.

Step 1. Repeat Step 1 to 3 for $(X_1, Y_1), \ldots, (X_M, Y_M)$.

Step 2. Reverse the sign of r bits of X_g abd Y_g, respectively, to be X and Y. If $X_g\cdot X > X_i\cdot X$, and $Y_g\cdot Y > Y_i\cdot Y$, $\forall\, i, i = 1,\ldots,M$, $i \neq g$, then take (X, Y) as the input training pair. Otherwise, go to Step 1 and try another pair.

Step 3. Take (X, Y) as the input pair of the following two equations until $J(b) = 0$.

$$
J(b) = -\sum_{k=1}^{n} x_{gk}\sum_{i=1}^{M} x_{ik}\cdot b^{Y_i\cdot Y}\cdot H(x_{gk}\sum_{i=1}^{M} x_{ik}\cdot b^{Y_i\cdot Y})
$$

$$-\sum_{k=1}^{p} y_{gk} \sum_{i=1}^{M} y_{ik} \cdot b^{X_i \cdot X} \cdot H(y_{gk} \sum_{i=1}^{M} y_{ik} \cdot b^{X_i \cdot X}) \quad (13)$$

$$b(t+1) = b(t) - q \cdot \frac{\partial J(b)}{\partial b} \quad (14)$$

$$\frac{\partial J(b)}{\partial b} = -\sum_{k=1}^{n} \sum_{i=1}^{M} x_{gk} x_{ik} \cdot (Y_i \cdot Y) \cdot b^{Y_i \cdot Y_j - 1} \cdot H(x_{gk} \sum_{i=1}^{M} x_{ik} \cdot b^{Y_i \cdot Y})$$
$$-\sum_{k=1}^{p} \sum_{i=1}^{M} y_{gk} y_{ik} \cdot (X_i \cdot X) \cdot b^{X_i \cdot X_j - 1} \cdot H(y_{gk} \sum_{i=1}^{M} y_{ik} \cdot b^{X_i \cdot X}) \quad (15)$$

The final b is assigned to be $b(0)$ for the training of next input training pair. Go to Step 1 until all of the pattern pairs are trained.

Step 4. After all of the M pairs either deleted or trained, the final b is used to be the radix of the eBAM.

Note that in the above algorithm, Step 2 is to avoid any confusion of two identical pairs, and to increase the radius of the the (X_g, Y_g) pair. Because the reversion of the signs of r bits is the basin radius of the training pair. Thus, those pattern pairs trained and stored in the eBAM are guaranteed to be recalled when the input pair is located in its basin of which the radius is r. The cost function in Step 3 is different from Eqn.(4) is because in this algorithm we train one pattern pair each time.

3 Simulation Analysis

Example 1. The radix searching algorithm is aimed at looking for a minimal radix which is still able to recall all of the pairs to be stored. We apply the algorithm to store 200 pattern pairs in a 8×8 eBAM, and repeat the algorithm fro 10 times to get Table 1.

No.	b	No.	b
1	2.790619	6	2.855878
2	2.790124	7	2.832530
3	2.757570	8	2.790527
4	2.733255	9	2.807437
5	2.738821	10	2.821274

Table 1: The simulation of searching an optimal radix for the 8×8 eBAM.

Note that the reason why the b's are different is the 200 pairs are randomly generated. However, the radix derived from the searching algorithm is close to the natural base, $e \approx 2.71828$. This result, thus, encourages the utilization of the natural e in the implementation of the eBAM by VLSI circuits. The evolution of b and $J(b)$ during the searching procedure is shown in Fig. 1 and Fig 2., respectively.

Example 2. An 8×8 eBAM is stored with 70 different pattern pairs. We use the radix searching algorithm in Section 2.3.2 to find the radix which makes the eBAM stable and pairs recallable, and then use the training algorithm in Section 2.3.4 to derive the radix which ensures every pair is recalled within its basin, $r = 1$. The results are shown in Table 2.

No.	b (searching)	b (training)	No.	b (searching)	b (training)
1	2.251508	2.270246	6	2.349620	2.532752
2	2.277386	2.277387	7	2.134500	2.779011
3	2.213780	2.664818	8	2.317027	2.491975
4	2.201022	2.830397	9	2.263478	2.509213
5	2.438400	2.575281	10	2.304018	2.326273

Table 2: The simulation of training the 8×8 eBAM.

Note that the trained b is always larger than b, which is expected. The reason is we want every trained pair to be recalled with $r = 1$. This condition is more restrict than just be stable and recallable.

After the training procedure, all of the 70 pairs are correctly recalled with one bit error in the above 10 simulations. The evolution of b and $J(b)$ during the training procedure is shown in Fig. 3, and Fig. 4, respectively.

4 Conclusion

A searching algorithm of the optimal radix for an eBAM is present. It shows a method to derive the smallest radix and hence the largest dimension of the eBAM. The radix updating algorithm successfully increase the radix when new pattern pairs are stored without any loss of the optimality of the radix. Finally, the training algorithm defines an approach to store pattern pairs with fault tolerance.

References

[1] T. D. Chiueh, and R. M. Goodman, "High-capacity exponential associative memory," *Proc. IJCNN,* vol. I, pp. 153-160, 1988.

[2] T. D. Chiueh, and R. M. Goodman, "Recurrent correlation associative memories," *IEEE Trans. on Neural Networks,* vol. 2, no. 2, pp. 275-284, 1991.

[3] L. A. Glasser and D. W. Dopperpuhl, "The Design and Analysis of VLSI Circuits." Reading, MA: Addison-Wesley, 1985.

[4] Y.-J. Jeng, C.-C. Yeh, and T.-D. Chiueh, "Exponential bidirectional associative memories," *IEE Electronics Letters,* vol. 26, no. 11, pp. 717-718, May 1990.

[5] B. Kosko, "Bidirectional associative memory," *IEEE Trans. Systems Man Cybernet,* vol. 18, no. 1, pp. 49-60, Jan./Feb. 1988.

[6] C. A. Mead, "Analog VLSI and Neural Systems." Reading, MA: Addison-Wesley, 1989.

[7] C.-C. Wang, and H.-S. Don, "An analysis of high-capacity discrete exponential BAM," *IEEE Trans. on Neural Networks* (to appear)

[8] T. Wang, X. Zhuang, and X. Xing, "Weighted learning of bidirectional associative memories by global minimization," *IEEE Trans. on Neural Networks,* vol. 3, no. 6, pp. 1010-1018, Nov. 1992.

Fig. 1. The radix in the searching process

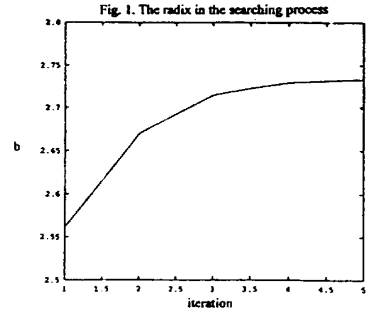

Fig. 2. The cost function in the searching process.

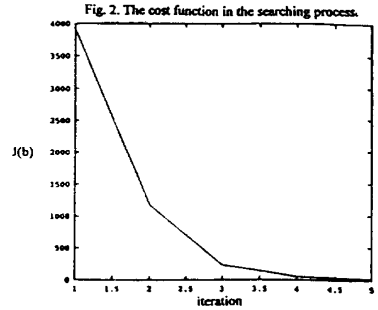

Fig. 3. The radix in the training process.

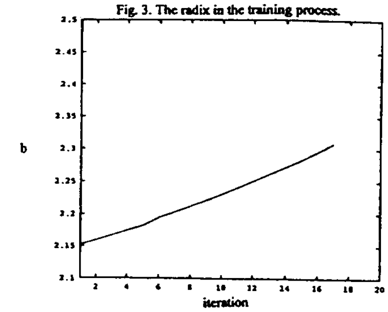

Fig. 4. The cost function in the training process.