Machine Intelligence & Robotic Control, Vol. 1, No. 1, 43–45 (1999)

Letter

Generalized Asymmetrical Bidirectional Associative Memory*

Tae-Dok Eom†, Changkyu Choi†, and Ju-Jang Lee‡

Abstract: A classical bidirectional associative memory (BAM) suffers from low storage capacity and abundance of spurious memories though it has the properties of good generalization and noise immunity. In this paper, Hamming distance in recall procedure of usual asymmetrical BAM is replaced with modified Hamming distance by introducing weighting matrix into connection matrix. This generalization is validated to increase storage capacity, to lessen spurious memories, and to enhance noise immunity using simulation work.

Keywords: Associative Memory, Bidirectional Associative Memory (BAM), Storage Capacity

1. Introduction

THE original bidirectional associative memory (BAM) was proposed by Kosko [1] extending the Hopfield auto-associative memory to a bidirectional one. It can associate an input pattern with a different stored output pattern of a stored pattern pair. Owing to its good generalization and noise immunity, BAM is well-suited for pattern recognition. Among many efforts to improve the performance of Kosko BAM (KBAM) by introducing new learning algorithms and adding dummy neurons, more layers, or interconnections inside each layer, the symmetrical BAM using the Hamming stability learning algorithm (SBAM) achieves the highest performance [2]. However, the logical symmetry of SBAM limits its use for knowledge representation and inference. To overcome the drawback, an asymmetrical BAM (ABAM) was proposed [3], which requires linear independence of stored patterns limiting its storage capacity. Inferred from that the capacity of feedforward multilayer network and radial basis function network is greater than the number of network neurons, the general BAM (GBAM) [4] used linear separability condition and increased the capacity slightly greater than the number of neurons in its layer.

In our generalized asymmetrical BAM (GABAM), simply multiplying input weighting matrix to the transition matrix of KBAM and deriving unique learning algorithm outperform the previous models.

2. Structure

Let $(x^i, y^i), i = 1, \cdots, p$, be the desired bipolar pattern pairs. Dimensions of x^i and y^i are n and m respectively. KBAM learns the pairs using transition matrix below:

$$W = \sum_i \frac{y^i {x^i}^T}{{x^i}^T x^i}. \qquad (1)$$

* Received July 28, 1999; accepted August 27, 1999.

† Department of Electrical Engineering, Graduate School of Electrical Engineering, Korea Advanced Institute of Science and Technology, 373-1, Gusung-Dong, Yusung-Gu, Taejon, Korea. E-mail: {tdum, flyers}@odyssey.kaist.ac.kr

‡ Department of Electrical Engineering, Faculty of Electrical Engineering, Korea Advanced Institute of Science and Technology, 373-1, Gusung-Dong, Yusung-Gu, Taejon, Korea. E-mail: jjlee@eekaist.kaist.ac.kr

If a pattern x is applied to input layer, each y^i is summed in output layer with a weighting factor proportional to $({x^i}^T x)/({x^i}^T x^i) = 1 - 2d_h(x^i, x)$ where $d_h(x^i, x)$ is $(\sum_{x_k \neq x_k^i} 1)/n$, the Hamming distance divided by n. W^T is used in backward association.

When human recognizes patterns, there are certain feature points more helpful to make a decision. Considering the different usefulness of each pixel information, diagonal matrix $\Lambda^i = \text{diag}(\lambda_1^i, \ldots, \lambda_n^i)$ can be multiplied before the correlation matrix. We suggest new forward and backward transition matrices W_f and W_b as follows:

$$W_f = \sum_i \frac{y^i {x^i}^T \Lambda_f^i}{{x^i}^T \Lambda_f^i x^i}, \quad W_b = \sum_i \frac{y^i {x^i}^T \Lambda_b^i}{{x^i}^T \Lambda_b^i x^i}. \qquad (2)$$

Applying a pattern x, each y^i is summed with a weighting factor $({x^i}^T \Lambda^i x)/({x^i}^T \Lambda^i x^i) = 1 - 2d_{\Lambda^i}(x^i, x)$, if new distance measure, $d_{\Lambda^i}(x^i, x)$, is defined as $(\sum_{x_k \neq x_k^i} \lambda_k^i)/(\sum_k \lambda_k^i)$, where λ_k^i reflects the usefulness of k-th input element. W_b is used in backward association.

3. Learning and Recall Process

Each weighting, ${x^i}^T \Lambda^i x$, is equal to $1-(x-x^i)^T \Lambda^i (x-x^i)/2$ if each λ_k^i is normalized at every learning step to sum into one. The generalized radial basis function (GRBF) network can be expressed as

$$y = \sum_i w^i f((x - c^i)^T \Lambda^i (x - c^i)), \qquad (3)$$

where activation function $f(u) = \exp(-u)$. Although GRBF does not include convergence procedure, GABAM can be translated as GRBF with $w^i = y^i$, $c^i = x^i$, and $f(u) = 1 - u/2$ from the static viewpoint. Original BAM has no hidden layer. However, augmenting weighting matrix Λ^i derives the virtual hidden layer from the transition matrix. The word *virtual* means that it only exists during learning process and merges into single transition matrix in recall process. If the hyperbolic tangent function replaces the usual sign output activation function, whole network is differentiable and any kind of gradient descent algorithm can be utilized to find Λ^i's which minimize the overall classification errors.

 Paper No. 1345–269X/99/010043-03

For the learning process, backpropagation algorithm which uses only the first derivatives is adopted. To learn the forward association, energy function is denoted by

$$E = \frac{1}{2}\|y - y^i\|^2 = \frac{1}{2}\sum_k (y_k - y_k^i)^2.$$

If a stepsize of learning, η, is sufficiently small, this pattern learning has the same performance with batch-type learning for all training pattern pairs. First, feedforward propagation is performed for the current training pair, (x^i, y^i). Calculating the output error, the error is propagated backwards. δ_k^o and δ_i^h denote the backpropagated error in output and hidden layer respectively. Then,

$$\delta_k^o = (y_k - y_k^i), \tag{4}$$

$$\delta_i^h = \sum_k y_k^i (1 - y_k^2)\delta_k^o, \tag{5}$$

$$\frac{\partial E}{\partial \lambda_j^i} = -2\frac{m(x_j^i \neq x_j)\sum_k \lambda_k^i - \sum_{x_k^i \neq x_k} \lambda_k^i}{(\sum_k \lambda_k^i)^2}\delta_i^h, \tag{6}$$

where

$$m(condition) = \begin{cases} 1, & \text{if } condition \text{ is true} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Finally, λ_j^i is updated by

$$\Delta\lambda_j^i = -\eta\frac{\partial E}{\partial \lambda_j^i}. \tag{8}$$

The backward association is learned in similar way.

The recall process is same with other BAM's. Input pattern continues propagating forwards and backwards repeatedly until it converges to a fixed point.

Like GRBF, the capacity of GABAM can be enhanced through the node addition techniques. Although the activation function of hidden layer is linear so that it suffers from low function approximation capability, nodes addition at centers of malfunctions brings about capacity improvement.

4. Experimental Results

The performance of an associative memory is usually evaluated in terms of its storage capacity, noise immunity, and spurious memories. These properties of GABAM were compared with those of the most promising SBAM, ABAM, and GBAM for the pattern recognition problem.

Storage capacities of five different models were compared in case of $m = n = 10$ by randomly generating 1000 test sets for K desired states (x-axis in **Fig. 1**). GABAM was superior to KBAM, competitive with ABAM and SBAM, and overwhelmed by GBAM. GBAM, which updates all elements in transition matrix W and does not contain correlation matrix, is rather similar to feedforward one-layer network. However, GABAM divides W into correlation matrix and input weighting matrix and adjusts only the latter. Therefore, it locates between BAM and feedforward network.

To investigate noise immunity, 26 test pattern pairs, each consisting of matched small-case and large-case alphabet letters, were considered [4]. Dimensions of input and output patterns were 49 (7 by 7 pixels). **Figure 2** shows the recall result of four BAM's in case of random-noise injection from zero to twenty percent. Three variants of GABAM, which used different number of virtual-layer-nodes and training patterns, were suggested. The first variant results in the intermediate performance between GBAM and ABAM. The graphs of the second and GBAM are almost overlapped over all noise injection range because learning algorithm of GBAM, stemming from linear separability condition, trains 49 by 49 weights to properly place hyperplanes, which is up to 49 by 52 weights updated by the second variant of GABAM. Although the linear separability condition limits the capacity of both GBAM and GABAM, GABAM converges faster due to the correlation matrix. The third variant is trained in the broader training set and overwhelms every BAM model.

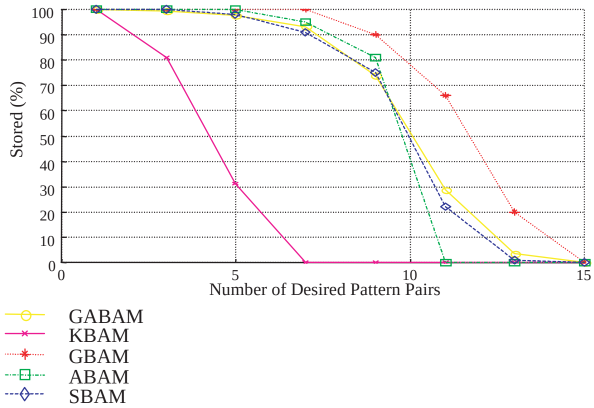

Fig. 1 Comparison of models (storage capacity)

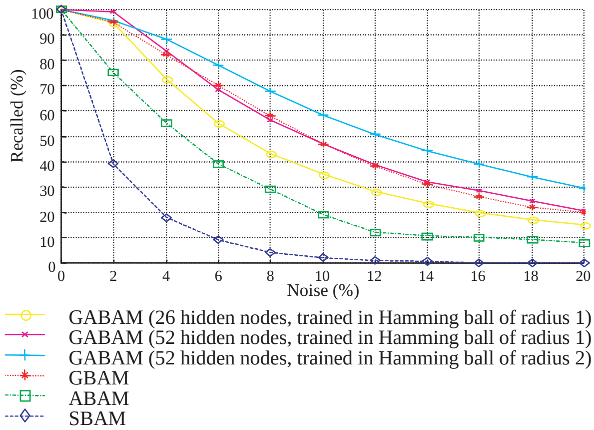

Fig. 2 Comparison of models (noise immunity)

Percentage of spurious memories was examined by generating 10,000 random initial states and checking whether they converged into spurious states. The results are 71%, 86%, 92%, and 98% for GABAM, GBAM, ABAM, and SBAM respectively.

5. Conclusion

The concept of BAM was generalized by introducing weighting matrix. The similarity to GRBF enables GABAM to utilize various learning techniques for weighting matrix. The proposed GABAM was proved to have large storage capacity, the best noise immunity, and the least spurious memories among existing models.

References

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

[2] X. Zhuang, Y. Huang, and S. Chen, "Better learning for bidirectional associative memory," *Neural Networks*, vol. 6, no. 8, pp. 1131–1146, 1993.

[3] Z. Xu, Y. Leung, and X. He, "Asymmetric bidirectional associative memories," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 24, pp. 1558–1564, 1994.

[4] H. Shi, Y. Zhao, and X. Zhuang, "A general model for bidirectional associative memories," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 28, no. 4, pp. 511–519, 1998.

Biographies

Tae-Dok Eom received B.E. and M.E. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology in 1993 and 1995, respectively, and is currently working toward D.E. degree in the same institute.

From 1995 to 1997, he had developed adaptive cruise control system using scanning laser as an industry-sponsored researcher. From January to February in 1999, he was a visiting researcher of Japan Automobile Research Institute. His research interests are in neural network for control (especially for skill learning), intelligent transportation system, and robotic control.

Changkyu Choi received the B.E., M.E., and D.E. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology in 1992, 1994, and 1999, respectively.

His research interests are in neural network for multiple association, control of chaotic system, and robotic control for medical application.

Ju-Jang Lee was born in Seoul, Korea, on November 14, 1948. He received the B.S. and M.S. degrees, both in Electrical Engineering from Seoul National University in 1973 and 1977, respectively, and the Ph.D. degree in Electrical Engineering from the University of Wisconsin in 1984. From 1977 to 1978, he was a research engineer at the Korean Electric Research & Testing Institute in Korea. From 1978 to 1979, he was a design and processing engineer at G.T.E. Automatic Electric Co., U.S.A. In 1983, he joined briefly the project engineer in the Research and Development Department of the Wisconsin Electric Power Co., U.S.A. He joined the Department of Electrical Engineering at KAIST in 1984, where he is currently an Associate Professor. In 1987, he was a visiting professor at the Robotics Laboratory of Imperial College Science and Technology. From 1991 to 1992, he was a visiting scientist at the Robotics of Carnegie-Mellon University. His research interests are in the areas of space robotics, flexible manipulators, variable structure control, intelligent control of mobile robot, chaotic control system, electronic control unit for automobile, and power system stabilizer. Dr. Lee is a member of IEEE, IEEE Industrial Electronics Society, KIEE, KITE, and KISS.