Associative Memory Using Random Vector Enhanced Phasor Neural Network

Dukki Chung

Dept. of Computer Engineering and Science
Case Western Reserve University
Cleveland, Ohio 44106
Phone: (216)368-8871
E-mail: dchung@alpha.ces.cwru.edu

Francis L. Merat

Associate Professor
Dept. of Electrical Engineering and Applied Physics
Case Western Reserve University
Cleveland, Ohio 44106
Phone: (216)368-4572
FAX: (216)368-2668
E-mail: flm@po.cwru.edu

Proceedings of the OAI Neural Network Symposium and Workshop
pp. 267-278, 1995

Associative Memory Using Random Vector Enhanced Phasor Neural Networks

Dukki Chung[1] and Francis L. Merat[2]
Department of Computer Engineering and Science[1]
Department of Electrical Engineering and Applied Physics[2]
Case Western Reserve University, Cleveland, OH 44106

Abstract

A new associative memory model composed of random vector enhancement followed by association is proposed. The Moore-Penrose pseudoinverse is employed to obtain the least-squares optimal solution; a gradient-descent type approach could also be used to obtain a near optimal solution. Computer simulations are performed to evaluate the model. Comparison of the proposed model with real domain backpropagation neural networks is presented. Simulations show that the proposed model can be trained orders of magnitude faster than real domain backpropagation neural networks.

I. Introduction

An associative memory associates one set of input vectors with another set of output vectors. The output vectors are usually user-specified. When the input and output vectors are different, the associative neural memory is called heteroassociative; when the input and output vectors are the same, the associative memory is called autoassociative. An associative memory can be implemented as electrical networks, optical filters or neural networks. There have been many associative neural memory models proposed during the last twenty years [1, 2, 3, 4]. These neural memories differ in terms of architectures, learning algorithms, recall mode, etc. The mapping performed by the autoassociative memory is apparent: it simply reproduces the input as an output. However, the autoassociative memory has a very distinct feature. It can correct for noisy and distorted inputs, and can accurately retrieve the input pattern even when the given input is partly missing.

II. Linear Associative Memory

Let $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ be the i-th associated pair of patterns where

$\boldsymbol{x}_i = \left[x_{i1}, x_{i2}, \ldots, x_{in}\right]^T$ and

$\boldsymbol{y}_i = \left[y_{i1}, y_{i2}, \ldots, y_{im}\right]^T$,

^{T} denotes the matrix transpose, and i=1,...,p, where p is the total number of pattern pairs. The patterns are linearly transformed by the relation

$$\mathbf{y}_i = \mathbf{M}\mathbf{x}_i \qquad \text{for } i=1, \ldots, p \tag{1}$$

where M is an m by n matrix. In terms of the key vector matrix **X** of size n by p (with the $\mathbf{x}_k$ as its columns) and the recollection vector matrix **Y** of size m by p (with the $\mathbf{y}_k$ as its columns), the associative memory must satisfy $\mathbf{Y} = \mathbf{MX}$. Generally **X** is not square, and the Moore-Penrose pseudoinverse is used [5]

$$\mathbf{M} = \mathbf{YX}^{+}. \tag{2}$$

If **X** is square and nonsingular, the solution becomes $\mathbf{M} = \mathbf{YX}^{-1}$.

III. Random Vector Enhancement

In a linear associative memory, the memory capacity and recall accuracy is limited by the pattern dimension and linear independence among patterns. Moreover, Minsky and Papert demonstrated that the linear neural network model can not solve even simple nonlinear problems [6], e.g., the XOR problem. Kohonen discusses various preprocessing methods to increase the orthogonality among patterns [7]. Various approaches have been proposed in the form of higher order neural network models [8]. One can also use products between pattern elements to enhance the input to the net. This was used in early versions of the functional link net [9]. Sutherland also used the same method to improve recall accuracy and increase the storage capacity of the holographic neural net [10]. Nonlinear mapping using polynomials is used in the complex domain nonrecurrent associative memory model [11]. These approaches are problem-dependent and not systematic solutions.

Random vector enhanced versions of the functional link net were proposed by Pao [12]. Random vector enhancement is a systematic way of increasing the pattern dimension, or including higher order information in the pattern sets. The random vector enhanced functional link net is easy to use, and was proved to be a universal function approximator [13]. This same type of approach is applied to the random vector enhanced phasor neural network (RV-PNN).

Suppose that there are n original attributes (features, or elements) and j enhanced attributes in the pattern $\mathbf{x}$. For the enhanced attribute x_i, the attribute is defined as

$$x_i = e_i \quad \text{for } i = n+1, \ldots, n+j \tag{3}$$

where $e_i = a_{i1}x_1 + \ldots + a_{in}x_n = \boldsymbol{a}_i^T \boldsymbol{x}$

$$\boldsymbol{a}_i = [\, a_{i1} \quad a_{i2} \quad \ldots \quad a_{in} \,]^T$$

$$\boldsymbol{x} = [\, x_1 \quad x_2 \quad \ldots \quad x_n]^T$$

and $\boldsymbol{a}_i$ is the random vector. The elements of the vector $\boldsymbol{a}_i$ are randomly generated on the real interval $[-\Omega, \Omega]$.

IV. Transformation into Complex Domain

The complex domain is used to model physical processes in electrical circuits, optics, quantum mechanics, etc. Since a single complex domain neuron can capture more information than a similar real domain neuron, the complex domain representation may yield more realistic neural models, e.g. modeling phase differences in the arrival of signals at the neuron. There are also many signal processing applications involving complex valued data in which phase information is an important feature. This is evident when using the output of the Fourier transform. The proposed model can represent information in the complex domain and is suitable for these application areas, as well as problems entirely in the real domain.

Usually, neural net inputs and outputs are represented by real numbers. Therefore, a mapping from the real number domain to the complex number domain is needed. The real number domain is typically unbounded (i.e., from $-\infty$ to $+\infty$). The real numbers can be converted into complex numbers by simply assigning phase angles and unit magnitudes. The resultant vectors are bounded by the unit circle on the complex plane (phasors). This mapping can be represented by

$x \rightarrow e^{i\theta}$. One way to assign a phase angle θ is to use the Z-score

$$\theta = \frac{2\pi}{1 + e^{-\frac{x-\mu}{\sigma}}} \tag{4}$$

where μ and σ are, respectively, the mean and standard deviation of the input key vectors. Equation (4) converts unbounded real values into phase angles from 0 to 2π.

After the enhancements, $\mathbf{X}_E$ (enhanced key vector matrix) and $\mathbf{Y}$ (recollection vector matrix) are transformed to the complex number matrices $\mathbf{X}_{Ec}$ and $\mathbf{Y}_c$, respectively. The pseudoinverse of $\mathbf{X}_{Ec}$ can then be calculated to form the memory matrix **M**, by equation (2).

For the complex number input patterns, complex number random vectors are generated over a magnitude range which avoids saturation of the transfer function. After the enhancement by equation (3), these patterns are fed into a sigmoidal transfer function like $tanh(real(x)) + i\ tanh(imag(x))$. As described before, these enhanced complex patterns and the recollection vectors (complex domain) then form the memory matrix **M** by equation (2).

V. Iterative Learning

Greville's theorem [14] provides the recursive formula for computing the pseudoinverse of a matrix. The idea is to partition the original matrix into columns and combine them one at a time, thereby computing the pseudoinverse of the new submatrix from the already computed pseudoinverse of a smaller submatrix and the new column. However, the same effect can be approximately obtained by using

$$\mathbf{M}^{new} = \mathbf{M}^{old} + \alpha(\mathbf{Y} - \mathbf{M}^{old}\mathbf{X})\mathbf{X}^{*} \qquad (6)$$

where * is the conjugate transpose, and α is the learning rate (a small positive number). By equation (6), it is possible to construct the pseudoinverse based **M** matrix iteratively. This learning rule is similar to the complex LMS algorithm [15].

Since the learning is achieved by a linear approach, finding the global minimum by using the gradient is guaranteed. With this scheme, it is necessary to present the pairs ($\mathbf{x}_1$, $\mathbf{y}_1$), ($\mathbf{x}_2$, $\mathbf{y}_2$), ... ($\mathbf{x}_P$, $\mathbf{y}_P$) several times in order to achieve a desired error level. Thus, equation (6) represents a gradient-descent type iterative learning algorithm. The pseudoinverse approach may not be suitable for a real-time neural based system. In this situation, equation (6) can be used as a generalized, iterative solution.

VI. Computer Simulations

1. XOR Test

This is the most famous neural network test example. For this test, the conventional backpropagation neural network has 2 hidden nodes, and the RV-PNN has 4 enhancements. The backpropagation neural network was trained until the mean squared error reached 10^{-4}, using the Quickprop algorithm [16]. The RV-PNN was trained using the iterative algorithm described in section V, until the mean squared error reached 10^{-4}. Both were trained 10 times with the results shown in Table 1.

	Average Epochs	Std. Dev.
Backpropagation Neural Network (2 hidden nodes)	102.6	58.1
RV-PNN (4 enhancements)	13.9	7.3

Table 1. Comparison of XOR test

2. SONAR Test (Classification)

Gorman and Sejnowski [17] published an early realistic example of sonar signals bounced off a metal cylinder (111 signals) and a roughly cylindrical rock (97 signals). The signals consist of energy measurements in each of 60 wavebands. Twelve-thirteenths of the data was used as a training set. Classification error rates of 12-15% were obtained using a real domain backpropagation neural network with 6-24 hidden nodes. The comparison between the RV-PNN and the classical neural network model is summarized in Table 2. All backpropagation neural networks were trained using the Quickprop algorithm until the mean squared error reached 10^{-4}. The RV-PNN was trained using the iterative algorithm described in section V, until the mean squared error reached 10^{-4}. Both models were trained and tested 10 times. Training times were measured on an HP 9000 712/60 workstation.

		% Right (Training)	% Right (Test)	Std. Dev (% Right)	Training Time (sec)	Std. Dev. (Training Time)
Real Domain Backpropagation Neural Network	6 Hidden Nodes	100%	84%	3.74	98.3	129.7
	12 Hidden Nodes	100%	85%	2.43	74.2	37.3
	24 Hidden Nodes	100%	87%	3.02	163.7	156.5
RV-PNN (300 enhancements)		100%	90%	1.4	3.5	0.6

Table 2. Performance comparison between backpropagation neural networks and the RV-PNN

3. Complex Domain XOR Test

This is the counterpart of the real domain XOR problem. The complex domain backpropagation neural network has 4 hidden nodes, and the RV-PNN has 10 enhancements. The complex domain neural network was trained until the mean squared error reached 10^{-2}, using backpropagation. The RV-PNN was trained using the pseudoinverse described in section II. Both were trained 10 times. Training times were measured on a 486DX/33MHz PC.

Input 1	Input 2	Output
i	i	0
i	$1+i$	1
$1+i$	i	1
$1+i$	$1+i$	0
0	i	i
0	$1+i$	$1+i$
1	i	$1+i$
1	$1+i$	i

Input 1	Input 2	Output
1	1	0
1	$1+i$	i
$1+i$	1	i
$1+i$	$1+i$	0
0	1	1
0	$1+i$	$1+i$
i	1	$1+i$
i	$1+i$	1

Table 3. Complex domain XOR inputs and outputs for the neural networks

	Training Time (sec)	Std. Dev.
Complex Domain Backpropagation Neural Network (4 Hidden Nodes)	73.7	29.2
RV-PNN (10 enhancements)	0.18	0.02

Table 4. Comparison of complex domain XOR test

4. Autoassociative Memory using RV-PNN

Geometric shapes can be represented by Fourier boundary descriptors in the complex domain [18]. Typical sample shapes used in this test are shown in Figure 1 and Figure 2. The shape boundaries were described by 140 to 240 samples, and transformed by FFT. Sixty-four of the transform coefficients were used as a Fourier boundary descriptor. The noise-free boundary descriptors from Figure 1 were recorded using the RV-PNN autoassociative memory with 64 enhancements. Next, the noisy descriptors from Figure 2 were used to test the autoassociative memory. The outputs from the autoassociative memory corresponding to the noisy inputs of Figure 1 are shown in Figure 3. This autoassociative memory shows excellent noise filtering characteristics suitable for many pattern recognition applications.

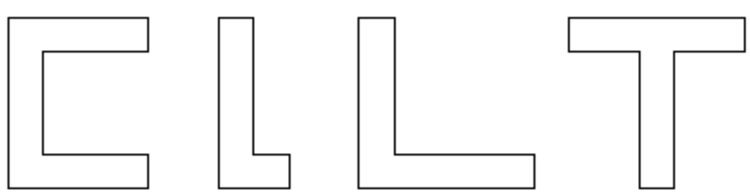

Figure 1. Simple geometric shapes

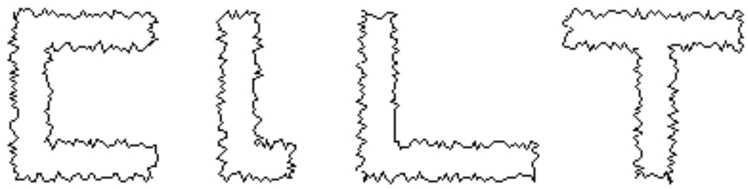

Figure 2. Noisy geometric shapes

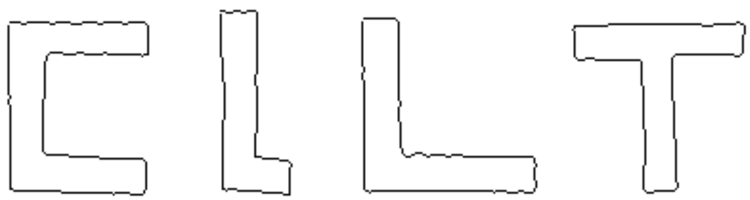

Figure 3. Geometric shapes filtered by RV-PNN autoassociative memory

VII. Conclusions

An associative memory model using a random vector enhanced phasor neural network (RV-PNN) has been described and compared with classical real domain backpropagation neural networks using computer simulations with various data. The RV-PNN performs as well as classical neural networks in the examples tested, and the training speed is orders of magnitude faster. Since the model can represent information in the complex domain, it is especially suitable for complex domain problems such as invariant pattern feature detection and signal processing. The RV-PNN can also be used to solve problems involving only the real domain.

References

1. Amari, S. I., "Learning Patterns and Pattern Sequences by Self-Organizing Nets of Threshold Elements," in *IEEE Transactions on Computers*, C-21, pp. 1197-1206, 1972

2. Kohonen, T., "Correlation Matrix Memories," in *IEEE Transactions on Computers*, vol. C-21, pp. 353-359, 1972.

3. Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," in *Proc. Nat. Acad. Sci. U.S.A.*, vol. 79, 2445-2558, 1982.

4. Kosko, B., "Bidirectional Associative Memories," in *IEEE Transactions on System, Man, and Cybernetics,* SMC-18, pp. 49-60, 1988.

5. Penrose, R., "A Generalized Inverse for Matrices," in *Proc. Cambridge Philos. Soc.* (England), LI, pp. 406-413, 1955.

6. Minsky, M. and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1988.

7. Kohonen, T., *Self Organization and Associative Memory*, Spring-Verlag, 1989.

8. Giles, C. and T. Maxwell, "Learning, Invariance, and Generalization in High-Order Neural Networks," in *Applied Optics*, vol. 26, no. 23, pp. 4972-4978, 1987.

9. Pao, Y. H., *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, 1989.

10. Sutherland, J. G., "A Holographic Model of Memory, Learning and Expression," in *International Journal of Neural Systems*, vol. 1, no. 3, pp. 259-267, 1990.

11. Hunt, B., S. Nadar, P. Keller, E. VonColln, and A. Goyal, "Synthesis of a Nonrecurrent Associative Memory Model Based on a Nonlinear Transformation in the Spectral Domain," in *IEEE Transactions on Neural Networks*, vol. 4, no. 5, pp. 873-878, Sep 1993.

12. Pao, Y. H., G. H. Park, and D. J. Sobajic, "Learning and Generalization Characteristics of the Random Vector Functional-Link Net,", in *Neuro Computing*, vol. 6, pp. 163-180, 1994.

13. Igelnik, B. and Y. H. Pao, "Stochastic Choice of Basis Functions in Adaptive Function Approximation and the Functional-Link Net," in *IEEE Transactions on Neural Networks*, 1994 (accepted for publication).

14. Greville, T.N.E., "Some Applications of the Pseudoinverse of a Matrix," in *SIAM Rev*., II, pp 15-22, 1960.

15. Widrow, B., J. McCool, and M. Ball, "The complex LMS algorithms," in *Proceedings of IEEE*, vol. 63, no. 4, pp. 719-720, April, 1975.

16. Fahlman, S. E., "Faster Learning Variations on Back Propagation: An Empirical Study," in *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann Publishers, 1988.

17. Gorman, R. P. and T. J. Sejnowski, "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets," in *Neural Networks*, vol. 1, pp. 75-89, 1988.

18. Gonzalez, R. C. and P. Wintz, *Digital Image Processing*, Addison Wesley, 1987.