

A Structured Adaptive Neural Network for Pattern Recognition VLSI

J. B. Kuo, E. J. Wong, C. C. Chen, C. C. Hsiao

Rm. 526, Dept. of Electrical Eng., National Taiwan University
1, Roosevelt Rd. Sec. 4, Taipei, Taiwan 107
FAX:886-2-363-8247, Telephone:886-2-363-5251

Abstract

Current adaptive neural network structures using back-propagation or LMS algorithms for pattern recognition have bottlenecks in speed, stability and fault-tolerance during training. In addition, a huge number of interconnects between adaptive linear neurons required severely limit VLSI implementation capabilities, which are critical for high speed pattern recognition. In this paper, a block adaptive neural network structure suitable for VLSI implementation is presented. According to simulation results, the adaptive network with local blocks provides a shorter learning time, a better stability, a better fault-tolerance property, and a much better expansion capabilities.

Summary

Fig. 1(a) shows a pattern recognition system, which is composed of a translation invariant networks and an adaptive two-layer network. The translation invariant networks are used to map a retinal image into multi-bit outputs [1]-[4]. The adaptive two-layer network [5],[6] made of $2n^2$ adaptive neurons and $2n^4$ connections between them, can be trained to provide output responses to the original image as required. Software implementation of the adaptive neural network using back-propagation [7]-[9] or least-mean-square(LMS) algorithms [1],[2],[10],[11] has been demonstrated. The computation speed is determined by the number of neurons and connections between them. As the size of the input patterns increases, the number of neurons and connections increases. Consequently, the computation time increases dramatically. As a matter of fact, neural network implemented by hardware has been used to enhance

the speed performance [12]-[14]. However, for high speed large scale pattern recognition, the standard adaptive neural network structure may not be sufficient for hardware implementation. Fig. 1(b) shows the block adaptive neural network structure for an $n \times n$ input array. In the block structure, the system is partitioned into n^2 blocks. Each block contains $n + 1$ neurons and $n^3 + n$ connections between them, which are grouped into two layers. Instead of $2n^4$ global connections at two layers in the standard structure, only n^3 local connections at the first layer and n connections at the second layer are needed in the local block structure. Compared to the standard structure, the second layer neurons in all local blocks are separated from one another. Therefore, not only the number of neurons can be minimized, but also the global interconnect crossovers can be greatly reduced, which facilitates VLSI implementation considering the two-dimensional metal routing capabilities. In addition, the local block structure can be easily concatenated to build a large adaptive neural network. Furthermore, each local block needs to consider convergence associated with only a portion of inputs during training. Consequently, the block structure converges faster during training since it is much easier to identify different patterns in a multi-dimensional space. Both back-propagation and LMS algorithms have been applied in the block adaptive neural network structure. The adaptive neuron with a modified LMS adaptive algorithm has been used in the system. The k th error, ϵ_k is: $\epsilon_k = (d_k - x_k^T w_k) \frac{|d_k - q_k|}{2}$ where ϵ , d_k , x_k , w_k , and q_k are the k th error, desired output, input, weight, and binary output respectively. With the new error function in the adaptive neuron, the adaptation goal for each adaptive neuron changes from $d_k = x_k^T w_k$ to $d_k = q_k$. As a result, the learning time can be shortened owing to the fact that a binary instead of an analog value is used at the neuron output. Widrow's adapt procedure [1],[2] has been used in the block adaptive neural network structure. With binary inputs and outputs, the training objective of the LMS algorithm is to reduce the number of output errors to zero according to the minimal disturbance principle. The back-propagation algorithm [15]-[17] has also been used in the block adaptive neural network structure. The objective of back propagation is to establish the desired responses for

the neurons by propagating back the output error signal through the network to modify the weight values.

Based on the block adaptive neural network structure, the performance in terms of learning time, fault tolerance has been studied for the adaptive neural network for pattern recognition. Forty training patterns in an 8×8 array as shown in Fig. 3(a), including 36 alphanumerics and 4 Chinese characters, have been used. As shown in Fig. 3(b), initially, over 40% of the output patterns are incorrect. After two iterations, the system adapts itself such that the output patterns are getting closer to the desired patterns as shown in Fig. 3(c). Fig. 4 shows the learning curves (the output percentage error vs. the CPU time) for the adaptive network with and without local blocks using the LMS algorithm. The learning curves for the structure with local blocks indicate a much quicker learning process. The local block structure can adapt to the desired patterns within four iterations. Since much fewer computations required for each iteration, the learning time, which is defined as the CPU time required for the neural network adapts to the input patterns with zero output errors, is much less for the block system. Using a 12.5 MIPS workstation, the learning time of the block system is about 9 seconds, as compared to over 150 seconds for the case without local blocks. In fact, the training time needed for hardware version of the adaptive neural network using local blocks can be even much less owing to parallel processing properties. Therefore, adopting the local block structure in the adaptive neural network can be very helpful for high speed pattern recognition applications. The number of digits after the decimal point in computation is critical in determining the learning times for both back-propagation and LMS systems with the block structure. The requirement of the computation precision in the adaptive network has been studied. Fig. 4 also shows the learning curves of the adaptive system using local blocks with 3, 4 and 7 digits precisions after the decimal point in the computation during the adaptation process. The cases with local blocks show no difference in learning time for computation with various precisions. In fact, a 3-digit precision in computation is sufficient for the adaptive network with local blocks using the LMS algorithm. The less rigorous precision requirement leads to an easier circuit implementation. As for the back-propagation algorithm, a much more rigorous requirement on the computation precision for the standard adaptive structure limits the feasibility of hardware implementation. More than 5-digit precision is necessary for the system without local blocks. Consequently, VLSI implementation is much more difficult due to the complicated analog circuits involved. With local blocks, the difficulties can be resolved for the back-propagation system. Fig. 5 shows the learning curves for the cases with local blocks using the back propagation algorithm for various precisions. Using a 2 MIPS computer, the learning time is below 120 seconds for the more than 2-digit precision cases, which is two orders of magnitude shorter as compared to that for the standard back propagation system. As a result, the back

propagation algorithm can be competitive as long as the block structure is used in the system. More insight into designing back-propagation adaptive network using the block structure can be obtained by studying the influence of the learning rate, η . Fig. 6 shows the learning times of the back propagation system with local blocks for various η 's. Cases with η 's of .25, 0.5, 0.7, 1.0 have been studied. A smaller η provides a better stability [3],[4] during training, however, its learning time is longer. On the other hand, a large η may lead to instability. As a matter of fact, the dependence of learning time on the learning rate is strongly correlated to the input patterns. Fig. 7 shows the learning time vs. the learning rate, η , changing from 0.1 to 1.0. For an η around 0.75, the energy associated with the overall system drops to a local minimum [10],[15],[16] and no convergence exists during training. Consequently, a very long learning time is expected. As long as η 's not around 0.75, convergence is guaranteed during training. Different from the network without the block structure, the system can work with an η exceeding 1.

The block structure in adaptive neural network system can be expedient for both software and hardware implementations. For the software version, the block structure can be taken advantages to shorten the learning time with a much better stability. In addition, the precision requirement on computation is less rigorous. Furthermore, the expansion capability of the local block structure facilitates a large scale pattern recognition system, which can be either high resolution patterns and/or large volume input patterns. For example, in a 32×32 input system, as shown in Fig. 8, the 8×8 local blocks can be used as basic modules to construct the system. Local frames of 8×8 arrays out of the 32×32 inputs can be used as inputs to 1024 local blocks. Consequently, the complexity of the overall interconnects still remains the same as for the 8×8 systems, which is very critical for VLSI implementation. In addition to the advantages in pattern learning and feasibility of VLSI implementation, the adaptive structure with the local block concept has a better fault-tolerance capability. For the adaptive neural network without local blocks, a portion of the network broken may lead to a widespread interference. Using the local blocks, a partial malfunction in the network results in a localized damage in a single output. Therefore, a much better fault tolerance exists in the adaptive neural network with the local block structure.

References

- [1] B. Widrow, R. G. Winter and R. A. Baxter, "Layered Neural Nets for Pattern Recognition," *IEEE Trans. on ASSP*, pp.1109-1117, July 1988
- [2] B. Widrow and R. Winter, "Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition," *IEEE Computer*, pp.25-39, March 1988
- [3] D. E. Rumelhart, J. McClelland, "Parallel Distribution Processing," *Massachusetts Institute of Technology*, 1986

[4] *Darpa Neural Network Study*, October 1987-February 1988

[5] G. Mirchandani, and W. Cao, "On Hidden Nodes for Neural Nets," *IEEE Trans. on Circuits and Systems*, May 1989

[6] D. S. Touretzky and D. A. Pomerleau, "What's Hidden in the Hidden Layers," *Neural Networks*, August 1989

[7] D. J. Burr, "Experiments on Neural Net Recognition of Spoken and Written Text," *IEEE Trans. ASSP*, Vol. ASSP-36, No.7, pp.1162-1168, Jul. 1988

[8] T. P. Vogl, J.K. Mangis and A.K. Rigler, "Accelerating the Convergence of the Back Propagation Method," *Biol. Cybern.*, Vol. 59, pp.257-263, 1988

[9] R. P. Gotman and T.J. Sejnowski, "Learned Classification of Sonar Targets Using a Massively Parallel Network," *IEEE Trans. ASSP*, Vol. ASSP-36, No.7, pp.1135-1140, Jul. 1988.

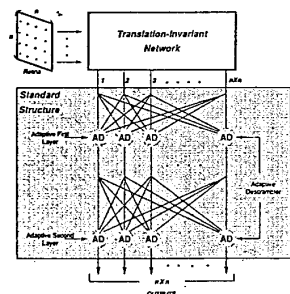
[10] Masfumi Hosokawa, Sigeru cmatu "A New Approach for Pattern Recognition by Neural Networks with Scrambles" *I Joint Conference of Neural networks (IJCNN)*, I-183-I-188, 1989.

[11] C.R. Winter and B.Widrow, "MADALINE RULE II: A Training Algorithm for Neural Networks", *Second Annual International Conference on Neural Networks*, 1-401 1988.

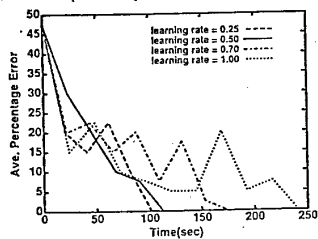
[12] H. P. Graf and L. D. Jackel, "Analog Electronic Neural Network Circuits," *IEEE Circuits and Devices Magazine* 1989

[13] M. A. C. Maher, Stephen P. Deweerth, M. A. Mahowald, and C. A. Mead, "Implementing Neural Architectures Using Analog VLSI Circuits," *IEEE Trans. on Circuits and Systems*, May 1989

[14] H. P. Graf, L. D. Jackel, and W. E. Hubbard, "VLSI Implementation of a Neural Network Mode," *IEEE*, March 1988



1 A pattern recognition system composed of a translation-invariant neural network and an adaptive two-layer network.



5 The learning curves of the adaptive neural network with local blocks using the back propagation algorithm for various learning rates $\eta = 0.25, 0.5, 0.7, 1$ smoothing factor, $\alpha = 0.9$.

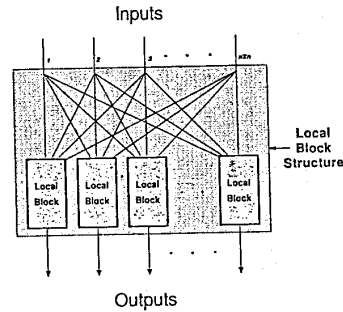


Fig. 2 A new adaptive structure with local blocks.

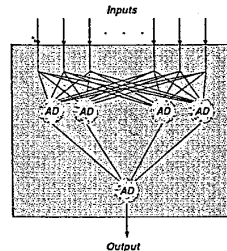


Fig. 3 The local block structure.

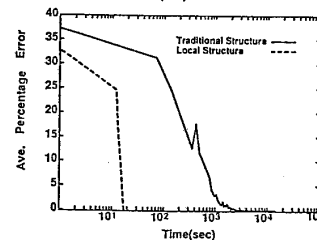
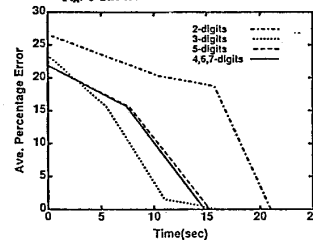


Fig. 4(a) The learning curves of the adaptive neural network with and without local blocks using the back propagation algorithm with a 7-digit precision in the simulation.

(b) The learning curves of the adaptive neural network with local blocks using the back propagation algorithm for various calculation precisions.

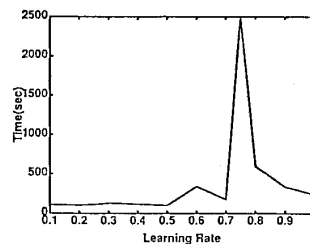
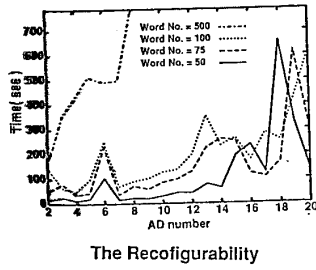


Fig. 6 The learning time vs. the learning rate for the adaptive neural network with local blocks using the back propagation algorithm.

Fig. 7 The learning time vs. AD number curves for the local block system with 50, 75, 100, and 500 8×8 input patterns.



The Reconfigurability

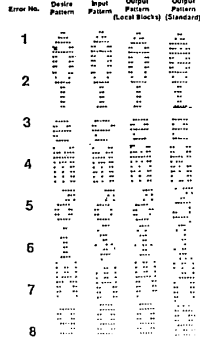
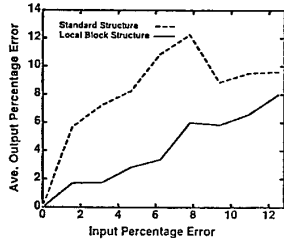


Fig. 8 (a) The reconfigurability of the local block system for 8×8 inputs. The average output percentage error vs. input percentage error curves. (b) Examples showing the reconfigurability of the local block system. 8×8 inputs with 1 to 8 errors, desired input patterns, output patterns with and without local blocks are shown.

Applications:

A 32×32 Pattern Recognition realized by 8×8 Local Blocks

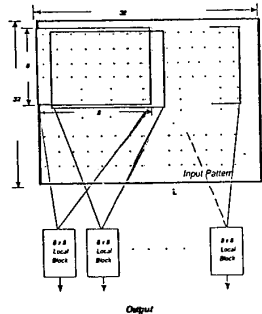


Fig. 11 A proposed 32×32 adaptive neural network structure using modules of 8×8 local blocks.

Fault Tolerance Property

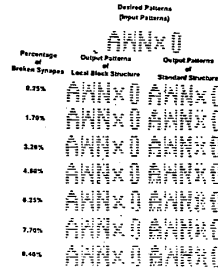
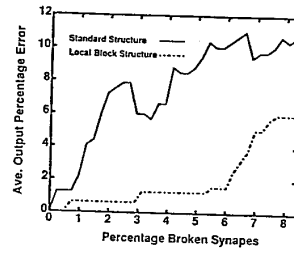


Fig. 9 (a) The fault-tolerance property of the local block system. The average output percentage error vs. the percentage broken synapse curves for the local block system. (b) The 5 input patterns to the system and the output patterns of the local block and standard structures with 0.25%, 1.7%, 3.2%, 4.6%, 6.25%, 7.7%, 8.4% broken synapses.

Learning Curve
(for Large Volume Training Patterns)

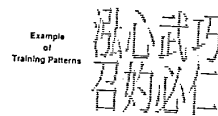
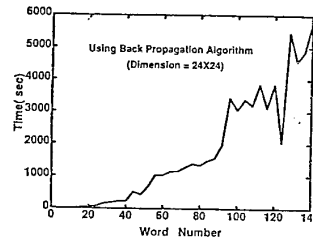


Fig. 10 The learning time vs. input word number curves for the local block system using back propagation algorithm with 24×24 Chinese character input patterns.