

A Coded Block Neural Network System Suitable for VLSI Implementation Using an Adaptive Learning-Rate Epoch-Based Back Propagation Technique

M. W. Mao, B. Y. Chen and J. B. Kuo
 P.O.Box 23-194, Dept. of Elect. Eng., National Taiwan University
 Taipei, Taiwan 106-17

ABSTRACT

This paper presents a coded block adaptive neural network system suitable for VLSI implementation using an adaptive learning-rate epoch-based back propagation technique to train large-volume input patterns. Using the adaptive learning-rate epoch-based back propagation technique, the coded block neural network system, 500 frequently-used Chinese characters have been successfully trained in 47.2 hours using a 28 MIPs computer. In addition, training of the epoch-based system is much less sensitive to initial weights and irrelevant to the order of the input patterns as compared to the system using the conventional back propagation algorithm.

SUMMARY

A pattern recognition system composed of a translation-invariant network and a standard adaptive two-layer network is used to map a retinal image into multi-bit outputs [1][2][3]. The standard adaptive two-layer network [4][5] can be trained to provide output responses corresponding to the original image as required. Software implementation of the adaptive neural network using back propagation [6][7][8] has been demonstrated. For large-volume Chinese character pattern recognition, training time is critically determined by the number of neurons and connections between them. A block adaptive network structure suitable for VLSI implementation has been reported to enhance the learning speed [9] for alphanumeric recognition. A coded block neural network system using the conventional back propagation algorithm suitable for VLSI implementation, which requires an order-of-magnitude fewer local blocks, has also been suggested to provide a faster learning process for the alphanumeric recognition [10]. However, for training large-volume patterns such as Chinese characters, the coded block neural network system using the conventional back propagation algorithm still needs a substantial learning time [11]. In this paper, a coded block adaptive neural network system using an epoch-based back propagation algorithm with an adaptive learning rate for training large-volume Chinese character patterns, which is insensitive to the initial weights and irrelevant to the order of input patterns and suitable for VLSI implementation, is presented.

Fig. 1 shows a coded block [9] neural network system using the back propagation algorithm to train large-volume input patterns made of $n \times n$ lattice

Coded Local Block Structure

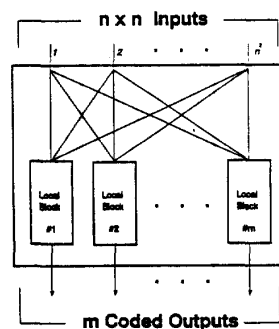


Fig. 1. The coded block adaptive neural network structure.

dots. In the coded block neural network system, there are m local blocks with outputs coded corresponding to the input patterns. Each local block as shown in Fig. 2 contains $S + 1$ neurons and $Sn^2 + S$ connections between them, which are grouped into two layers. Instead of $2n^4$ global connections in

Local Block Structure

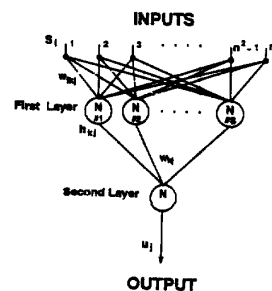


Fig. 2. A simplified hierarchical structure of a local block.

two layers as in the standard structure, only Sn^2 local connections are needed in the first layer and S connections in the second layer of the local block structure [9]. Compared to the standard structure, the second layer neurons in all local blocks are separated from one another. Therefore, in each local block, not only the number of neurons can be minimized, but also the number of synapses can be greatly reduced. In addition, only m local blocks are used in the coded block system. Consequently, the computation referred to all blocks in the coded block system during training can be shortened substantially.

However, for training large-volume input patterns, the coded block adaptive neural network using the conventional back propagation algorithm still cannot provide an acceptable learning time. Hence, the capability of learning in terms of the number of input patterns allowable is still limited. Furthermore, the learning time for the coded block system using the conventional back propagation algorithm is quite sensitive to the initial synaptic weights and the order of the input patterns to train. In a coded block system as shown in Fig. 1, there are m local blocks. Fig. 2 shows the j th local block in the coded block structure with s first layer neurons. During training, first, an epoch of patterns are presented to the inputs of the j th local block, as shown in Fig. 2, primed with random initial weights. Then, the weight value associated with the synapse between the second layer output and the k th first layer neuron ($w_{kj}^{(l)}(n)$) is changed as l th input pattern trained according to Eqs.(1)-(6) as shown in Fig. 3. Using the conventional back propagation algorithm, numer-

$$\begin{aligned} \Delta w_{kj}^{(l)}(n) &= \eta \delta_j^{(l)}(n) w_{kj}^{(l)}(n) + \alpha \Delta w_{kj}^{(l)}(n-1), \quad k = 1, 2, \dots, s \quad (1) \\ \delta_j^{(l)}(n) &= w_j^{(l)}(n) (1 - w_j^{(l)}(n)) (\delta_j^{(l)}(n) - s^{(l)}(n)) \quad (2) \\ w_j^{(l)}(n) &= \frac{1}{1 + e^{-\sum_{i=1}^s w_{ij}^{(l)}(n) s_i}} \quad (3) \\ \delta_j^{(l)}(n) &= \lambda_j^{(l)}(n) (1 - \lambda_j^{(l)}(n)) \delta_j^{(l)}(n) w_j^{(l)}(n-1), \quad k = 1, 2, \dots, s \quad (4) \\ \lambda_j^{(l)}(n) &= \frac{1}{1 + e^{-\sum_{i=1}^s \lambda_{ij}^{(l)}(n) s_i}} \quad (5) \\ \Delta w_{ij}^{(l)}(n) &= \eta \delta_j^{(l)}(n) w_{ij}^{(l)}(n) + \alpha \Delta w_{ij}^{(l)}(n-1), \quad k = 1, 2, \dots, s; \quad i = 1, 2, \dots, n^2 \quad (6) \\ \Delta w_{ij}^{(l)}(n) &= \sum_{k=1}^s w_{kj}^{(l)}(n) \delta_j^{(l)}(n) w_{ij}^{(l)}(n) + \alpha \Delta w_{ij}^{(l)}(n-1), \quad k = 1, 2, \dots, s \quad (7) \\ \Delta w_{ij}^{(l)}(n) &= \sum_{k=1}^s w_{kj}^{(l)}(n) \delta_j^{(l)}(n) w_{ij}^{(l)}(n) + \alpha \Delta w_{ij}^{(l)}(n-1), \quad k = 1, 2, \dots, s; \quad i = 1, 2, \dots, n^2 \quad (8) \\ \delta_j^{(l)}(n) &= \lambda_j^{(l)}(n) (1 - \lambda_j^{(l)}(n)) \delta_j^{(l)}(n) w_j^{(l)}(n-1), \quad k = 1, 2, \dots, s \quad (9) \\ \lambda_j^{(l)}(n) &= \frac{1}{1 + e^{-\sum_{i=1}^s \lambda_{ij}^{(l)}(n) s_i}} \quad (10) \\ E^{(l)}(n) &= (e^{(l)} - w_j^{(l)}(n))^2 \quad (11) \\ E(n) &= \sum_{l=1}^m (e^{(l)} - w_j^{(l)}(n))^2 \quad (12) \end{aligned}$$

Fig. 3. Important Equations.

ous epoches of pattern activation and error feedback are performed with all the training patterns until the output of each local block for all input patterns is correct. The conventional back propagation algorithm is classified as "pattern-based" since each individual weight is updated immediately after each input pattern is applied. In addition, during the training process, the learning rate is set to a constant. Until now, the conventional back propagation algorithm has been successfully used to train small-volume Chinese characters [10]. However, for large-volume pattern recognition applications, the pattern-based back propagation algorithm cannot provide an efficient training process. In order to facilitate training of large-volume patterns, an improved back propagation algorithm - the "epoch-based" back propagation has been developed. Instead of updating each individual weight value immediately after each pattern is presented at the input, the new epoch-based back propagation algorithm updates each individual weight value in the first and second layers only after all patterns have been sequentially applied in an epoch. Consider the j th local block in the epoch-based system, the weight value associated with the synapse between the local block output and the k th neuron in the first

layer is changed only once per epoch as shown in Eq. (7) in Fig. 3. Initially, a learning rate equal to the inverse of number of the input patterns is selected. After every training epoch, the learning rate associated with a specific pattern stays unchanged if the output of the system for the specific pattern matches its desired output. Otherwise, the learning rate increases by a value equal to 20% of its initial value. After the specific pattern matches its desired output, the learning rate associated with that specific pattern is decreased by a value equal to 20% of its initial value. If further training is necessary such that the learning rate associated with the specific pattern is supposed to be lower than its initial value, the learning rate has been set to be at its initial value. After training is successfully concluded, the learning rate associated with each input pattern may be different. As for the momentum (α), a value of 0.85 has been used. For the epoch-based system, the weight value associated with the synapse between the i th input (s_i) and the k th first layer neuron is updated only once per epoch according to the formula as shown in Eqs. (8)-(10) in Fig. 3. The better convergence capability of the epoch-based back propagation algorithm can be understood considering the error function associated with a local block, which is composed only one neuron and two synapses (w_1, w_2), as shown in Fig 4(a). The error function associated with the local block

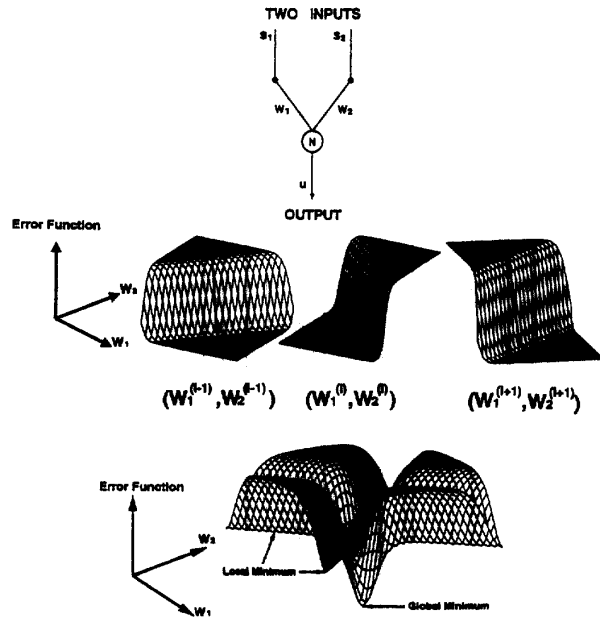


Fig. 4. The error function surface associated with a local block composed of a neuron and two synapses using the pattern-based and epoch-based back propagation algorithms.

- (a) The local block.
- (b) Three samples of the error function surfaces associated with the local block using the pattern-based back propagation algorithm.
- (c) A sample of the error function surface associated with the local block using the epoch-based back propagation algorithm.

as shown in Fig. 4(a) for a pattern-based system at the time the l th pattern trained is defined as the square of the difference between the desired target ($t^{(l)}$) and the neuron output ($u^{(l)}(n)$) as shown in Eq. (11) in Fig. 3. Based on the above equation, Fig. 4(b) shows three samples of the error function surfaces associated with the local block using the pattern-based back propagation algorithm at times $(l-1)$ th, l th, and $(l+1)$ th patterns trained in an epoch. As shown in Fig. 4(b), at times when $(l-1)$ th, l th, and $(l+1)$ th patterns trained, two weights change themselves from $(w_1^{(l-1)} \& w_2^{(l-1)})$ to $(w_1^{(l)} \& w_2^{(l)})$ and $(w_1^{(l+1)} \& w_2^{(l+1)})$ according to the back propagation algorithm. But the error function surface also changes as a result of the pattern-dependent property associated with the pattern-based system. Consequently, the chance for the pattern-based system adapts to the minimum error condition depends not only on the synaptic weights but also on their associated error function surfaces. On the other hand, for the local block as shown in Fig. 4(a), using the epoch-based back propagation algorithm, the error function is defined as sum of the square of the difference between the desired target and the neuron output for all input patterns, as shown in Eq. (12) in Fig. 3. Consequently, the error function surface of the epoch-based system stays fixed regardless of what input pattern is applied since it is independent of the specific input pattern. As a result, the error function surface maintains unchanged during training as shown in Fig. 4(c), where a global minimum along with two local minima exist [12]. Consequently, the neural network system using the epoch-based back propagation algorithm can have a much quicker convergence during training since it's much easier for the synaptic weights to find the location where the global minimum in the error function surface exists as a result of the fixed error function surface during training. In addition to the better convergence property in training, the epoch-based system has advantages in stability for the overall system. Using the pattern-based back propagation algorithm, the neural network system often suffers from the stability problems - the training time is very sensitive to the initial weights and the order of the input patterns. Thanks to the fixed error function surface, the coded block system using the epoch-based back propagation algorithm is much less sensitive to the initial weights and irrelevant to the order of the input patterns while training large-volume patterns. This is helpful for building a reliable neural network system. In order to confirm the advantages of the epoch-based back propagation algorithm with an adaptive learning rate in the coded block neural network system for training large-volume patterns, performance of the system in terms of learning time for training 500 frequently-used Chinese characters composed of 24×24 lattice dots. In the coded block structure under study, there are 16 local blocks with 12 neurons in the first layer in each local block. Fig. 5 shows the learning time vs. the number of Chinese characters used as input patterns. The learning time is measured in terms of CPU time using a 28 MIPs computer. For a small volume of input patterns, the difference in training time between the systems using pattern-based and epoch-based back

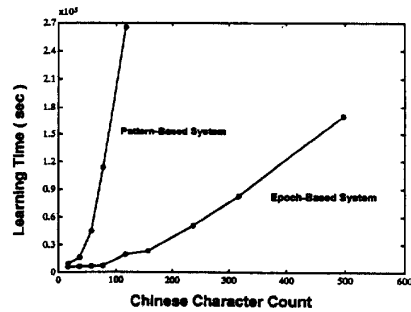


Fig. 5. The learning time vs. the input pattern numbers for training large-volume Chinese characters using the coded block neural network system with the epoch-based and pattern-based back propagation algorithms.

propagation algorithms is small. However, as the number of input patterns increases, the difference in learning time between the two algorithms grows wider. For training 500 patterns, the learning time for the system using the coded block structure with pattern-based back propagation algorithm is just not practical on a 28 MIPs computer. On the other hand, with the epoch-based algorithm, the coded block system can learn 500 input patterns in 47.2 hours. Fig. 6 shows the learning time vs. the learning rate for a coded block system using the pattern-based and the epoch-based back propagation algorithms for training 40 Chinese characters. The initial learning rate of the

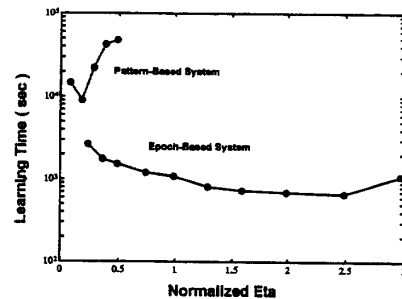


Fig. 6. The learning time vs. the initial learning rate for training 40 Chinese characters using the coded block neural network system with the epoch-based and pattern-based back propagation algorithms. The learning rate is normalized by a value, which is equal to the inverse of the number of the input patterns.

epoch-based system is normalized by a value equal to the inverse of the number of the input patterns. Using the pattern-based system, the system can have an acceptable learning time only for a limited range of learning rate. On the other hand, using the epoch-based back propagation algorithm, the learning time stays relatively insensitive to the initial learning rate. This is important for the system with an adaptive learning rate technique since the learning rate can have more space to be adjusted automatically. For training up to 500 input patterns, using the epoch-based back propagation algorithm with the adaptive learning rate technique, a much quicker convergence is always

guaranteed.

REFERENCES

- [1] B. Widrow, et. al., "Layer Neural Nets for Pattern Recognition," *IEEE TASSP*, 88
- [2] D. E. Rumelhart, et. al., "Parallel Distribution Processing," *MIT Press*
- [3] Darpa Neural Network Study, 88
- [4] G. Mirchandani, et. al., "On Hidden Nodes for Neural Nets," *IEEE TCAS*, 89
- [5] D. S. Touretzky, et. al., "What's Hidden in the Hidden Layers," *Neural Networks*, 89
- [6] W. P. Jones, et. al., "Back-Propagaion: A Generalized Delta Learning Rule," *Byte*, 87
- [7] K. Yamada, et. al., "Handwritten Numeral Recognition by Multi-layered Neural Network with Improved Learning Algorithm," *IJCNN*, 89
- [8] T. P. Vogl, et. al., "Accelerating the Convergence of the Back Propagation Method," *Biol Cybern*, 88
- [9] J. B. Kuo, et. al., "A Coded Block Adaptive Block Neural Network Structure for Pattern Recognition VLSI," 1991 Inter. Symp. on VLSI TSA
- [10] J. B. Kuo, et. al., "An Adaptive Neural Network Structure with Coded Local Blocks for Pattern Recognition VLSI," *IJCNN*, 1991
- [11] M. W. Mao, and J. B. Kuo, "A Coded Block Adaptive Neural Network System with a Radical-Partitioned Structure for Large-Volume Chinese Characters Recognition," *Neural Networks*, 92
- [12] B. Widrow, et. al., "30 Years of Adaptive Neural Networks Perception, Medaline and Back Propagation," *IEEE Proc.*, 90