Novel Dynamic Structure Neural Network for Optical Character Recognition

Ying-Tung Hsiao, Cheng-Long Chuang, and Cheng-Chih Chien

Department of Electrical Engineering
Tamkang University, Taipei, Taiwan, 251, R.O.C.
Tel: +886-2-26215656 Ext. 2786, Fax: +886-2-26209814
E-mail: clchuang@ee.tku.edu.tw

Abstract—**This paper presents a novel dynamic structure neural network (DSNN) and a learning algorithm for training DSNN. The performance of a neural network system depends on several factors. In that, the architecture of a neural network plays an important role. The objective of the developing DSNN is to avoid trial-and-error process for designing a neural network system. The architecture of DSNN consists of a three-dimensional set of neurons with input/output nodes and connection weights. Designers can define the maximum connection number of each neuron. Moreover, designers can manually deploy neurons in a virtual 3-D space, or randomly generate the system structure by the proposed learning algorithm. This work also develops an automatic restructuring algorithm integrated in the proposed learning algorithm to improve the system performance. Due to the novel dynamic structure of DSNN and the restructuring algorithm, the design of DSNN is fast and convenient. Furthermore, DSNN is implemented in C++ with man-machine interactive procedures and tested on many cases with very promising results.**

I. Introduction

Artifical neural networks derive their computing power through their massively parallel distributed structure and their ability to learn and therefore generalize. These information-processing capabilities make it possible for neural networks to solve complex problems that currently intractable. However, the developed conventional neural networks have a potential common defect. The architecture of a conventional neural network is configured with a regular form, and the designer has to improve the system performance by arranging the architecture of his designing network with trial-and-error. Because designers are impossible to derive all cases for forming a neural network, especially markets requires minimum designing time, the trial-and-error method may not only causes a neural network exhibits a degradation in performance, but also consumes considerable time in structuring a neural network.

There are many types of the neural networks for pattern recognition. These networks have been successfully applied in many problems of various fields. Some of these networks can be trained quickly but they do not have a high performance, such as CPN. In the other hand, some networks may have high accuracies, but they need to take significant time and design cycles to design and train the system, e.g. BP. It is difficult to get a balance between time-consumption and performance-requirement for designing neural networks. Thus, it is need to develop a new neural network system that can be constructed and reconstructed its network structure by the system itself for easy implement and high performance. Therefore, this study proposes a novel neural network, called dynamic structure neural network (DSNN), which requires a minimum designing time and few training iterations but produces a high accuracy output.

The performances of many types of neural networks are limited by their structure. Therefore the network with fixed architecture seems to lack adaptability. The DSNN overcomes this problem by allowing new neurons and connections to be created in the training stage. By the dynamic architecture of DSNN, the data with great similarity and interaction with each other can be classified as belonging to its category correctly with high efficiency. For pattern recognition, the form of input data constitutes the very design of the neural network, and therefore holds the key to its performance. 2-D Wavelet retains some spatial information of the characters. This paper uses this information for recognition characters to test and verify the DSNN performance.

This paper is organized as follows. Section II describes the optical character recognition technique. Section III proposes the DSNN and its learning algorithm. Section IV discusses the simulation results. Finally, the conclusions are drawn in the last section.

II. Dynamic Structure Neural Network

This work proposes a novel neural network to provide high accuracy, fast learning speed and quick convergence for pattern recognition. The designing time of the proposed neural network is almost zero for various kinds of applications. Moreover, this study develops a training algorithm to increase recognition accuracy for incorrect input data. The details of the DSNN regarding the supervised training algorithm and dynamic neural structure are described in the following subsections.

A. Network Architecture

The architecture of DSNN consists of three layers illustrated in figure 2. Layer 1 is the input layer with the same size as the input vector. Layer 3 is the output layer, and the middle (hidden) layer consists of preseted number of neurons that distributed randomly in a virtual 3-D space. Each neuron of the hidden layer is labeled as H_j, where j is the sequence number of the hidden neurons. The incoming arrow indicates the input of the neuron, and the

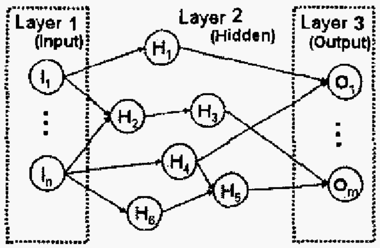

Figure 2. the schema of DSNN.

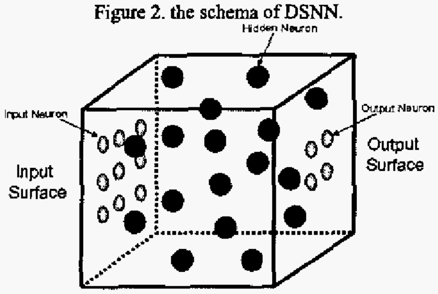

Figure 3. Illustration of a virtual 3-D cube space.

outgoing arrow indicates the output of the neuron. In the OCR application, the output stage of DSNN does not perform recognition function directly. The output layer of the DSNN consists of eight neurons for outputting the recognition result in ASCII form. The architecture of DSNN is reconfigurable. Initially, DSNN is configured with a preset number of neurons for the input, hidden and output layer. The position and connection of every hidden neuron are randomly generated. Compares to the traditional neural network, the DSNN does not require to decide the structure of the designing network system. This particular feature of the DSNN avoids designers to determine the number of layers and neurons by trial-and-error method that may cause the system falls into local minimum.

The virtual 3-D space of the hidden layer of the DSNN is a cube space in computer memory, shows in figure 3. The edge length of the virtual cube space depends on the neuron number of the entire network. As long as the edge length is defined, the edge length is fixed and will not be changed during the network operating. The edge length L_d can be defined as follows

$$L_d = \rho \times (10 \times N) \tag{7}$$

where N is the total initial neuron number of the network, and $(10 \times N)^3$ is the space that preserved for each initially generated neurons. ρ is the space preservation factor used to preserve extra space for further neurons generating. Typically, ρ is preseted between 1.5 to 3. Designers can set the value of ρ by experiment for various kinds of applications.

B. Neuron Operation

The input vector will be transmitted to the hidden neurons from the input layer to calculate the activation level. The activation level of each hidden neuron is formulated as follows

$$y_o(n) = \varphi_o\left(\sum_{i \in C} w_{io}(n) \cdot y_i(n) + b_o(n)\right) \tag{8}$$

where y_o is the output of neuron o, n is the iteration number of the process, w_{io} is the weighting of connection from original neuron i to destination neuron o, y_i is the input of neuron i, b_o is the bias of neuron o, φ_o is the activation function, and the activation function φ_o is always settled between 1 and -1 to maintain the convergence property.

C. Training of Output Layer

In contrast with BP, DSNN has no distinct layered structure for target vector propagating backward to train the hidden layer neurons. During training, for an input vector, the network feeds the input vector forward and generate a corresponding output value with the initial randomly generated weightings, biases, and structure. For the target vector corresponding to its input vector, we may define the output error as

$$e_o(n) = d_o(n) - y_o(n) \tag{9}$$

where e_o is the error of output neuron o, d_o is the target value of output neuron o, and y_o is the actual output value of output neuron o. After getting each error value of all output neurons, we start to refine the parameters of the output neurons. The correction $\Delta w_{io}(n)$ applied to $w_{io}(n)$ is defined by

$$\Delta w_{io}(n) = l \cdot \eta \cdot e_o(n) \cdot |y_o(n)| \tag{10}$$

$$l = \begin{cases} 1 & if \quad \Delta y_o(n) > 0 \\ -1 & if \quad \Delta y_o(n) < 0 \end{cases} \tag{11}$$

where Δw_{io} is the weighting variation value of the connection from original terminal neuron i to destination terminal neuron o. η is the learning rate, l is the refine direction indicator used to decide the direction for weighting turning.

The adjustment applied to $b_o(n)$ is defined as

$$\Delta b_o(n) = \eta \cdot e_o(n) \tag{12}$$

where Δb_o is the bias variation value of the output neuron o. Now, we define the update formula for the io-th synaptic weight and the o-th bias as

$$w_{io}(n+1) = w_{io}(n) + \Delta w_{io} \tag{13}$$

$$b_o(n+1) = b_o(n) + \Delta b_o \tag{14}$$

where $w_{io}(n+1)$ and $b_o(n+1)$ are the refined weighting and bias of output neuron o, respectively.

D. Training of Hidden Neurons

After the supervised training process of the output neurons, we can start to propagate the error value backward to refine the parameters of the hidden neurons. First, the tuning momentum g is defined for hidden neurons to determine the turning value amount of them. For the hidden neurons connected to the output neuron, such as H_3 in figure 2, that can be observed the supervised training result of the output neuron O_m, the process of updating the hidden neuron can be formulated as follow

$$g_i(n) = \eta \cdot e_o(n) \cdot |y_o(n)| \tag{15}$$

where $g_i(n)$ is the turning momentum of the hidden neurons to the output neurons. For hidden neurons connected to the other hidden neurons, such as H_2 connected to H_3 in figure 2, the turning momentums of them are calculated from the turning momentum of rear side hidden neurons. For example, the turning momentum of neuron H_2 must be calculated by referencing the turning momentum of neuron H_3. The momentum of these hidden neuron *i* is defined as

$$s_i(n) = \frac{g_i(n)}{C_i} \tag{16}$$

and

$$g_j(n) = \eta \cdot s_i(n) \cdot |y_i(n)| \tag{17}$$

where $s_i(n)$ is the momentum of the hidden neuron *i*, $g_j(n)$ is the turning momentum of the hidden neuron *j* connected to the hidden neuron *i*, C_i is the total connection number of neuron *i*. The update weighting is in the form

$$\Delta w_{ji}(n) = l \cdot \eta \cdot s_i(n) \cdot |y_i(n)| \tag{18}$$

$$l = \begin{cases} 1 & if \;\; \Delta y_i(n) > 0 \\ -1 & if \;\; \Delta y_i(n) < 0 \end{cases} \tag{19}$$

where $\Delta w_{ji}(n)$ is the weighting variation value of the connection from original neuron *j* to destination terminal neuron *i*. The correction to $b_i(n)$ is

$$\Delta b_i(n) = \eta \cdot s_i(n) \tag{20}$$

where Δb_i is the bias variation value of the hidden neuron *i*.

E. Dynamic Connection

This work proposes the restructuring optimization algorithm simulated from the neuron regeneration of the real nerve system. According to the latest discovered knowledge of medical science, the neuron connections of the real nerve system can be reconfiguration by reaching the growth cone of the neuron for the place that contains high density of attraction particles [12-16]. The study simulates the neuroregeneration process to reconstruct and optimize the connections in the hidden layer for optimizing DSNN.

The proposed restructuring algorithm can produce or prune neurons and the connections between neurons in an unsupervised manner. This mechanism can help the system to escape the local optima by increasing positive connections for promoting a better system performance than of it before restructuring, and prune the no contribution connections to reduce the calculation complexity. The concept of searching the new connections is shown in figure 4. In figure 4, neuron N_n has three free connectors, with weighting W_{nf1}, W_{nf2} and W_{nf3}. These weighting is randomly generated from the initial stage. During the training stage, for each the free connector, this mechanism scans the output values of every hidden neuron N_k (k = a, b, …, and k ≠ n). According to

Table 1

The Growing rules of growth cone

W_{nfi}	y_n	y_k	*Att or Rep*
+	+	+	*Att*
+	+	-	*Rep*
+	-	+	*Rep*
+	-	-	*Att*
-	+	+	*Rep*
-	+	-	*Att*
-	-	+	*Att*
-	-	-	*Rep*

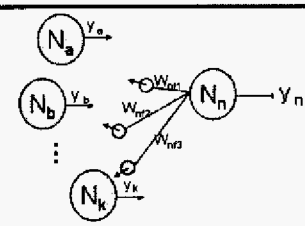

Figure 4. The concept of the neuroregeneration.

the output value and weighting of the scanned neuron *k*, the correction of the coordinate of the free connectors can be formulated as follow

$$\Delta(x_{fn}, y_{fn}, z_{fn}) = \sum_j D \cdot \frac{g_j}{L_j}(x_j, y_j, z_j) \tag{21}$$

$$D = \begin{cases} 1 & \text{if attraction} \\ -1 & \text{if repulsion} \end{cases} \tag{22}$$

where $\Delta(x_{fn}, y_{fn}, z_{fn})$ is the correction of coordinate of the free connector, L_j is the distance between the free connector and the scanned neuron, and (x_j, y_j, z_j) is the coordinate of the scanned neuron. The growing direction of a free connector (attraction or repulsion) described in (22) and is according to Table 1. To reduce the computational complexity of the network, it is need to prune the redundant connections. Considering a single neuron, if the absolute value of weighting of a connection is relatively smaller than of the others connections after several training iterations, the connection will be considered without contribution and deleted (pruned) from the system to reduce trivial calculation.

F. Dynamic Middle Layer

The mechanism of the dynamic growth of the middle layer allows the network to increase its learn ability by producing new neurons. When the network is unable to increase its accuracy in the training stage, the network reaches its maximum learning capability. If the network still keeps learning the training patterns under this status, the network may produce incorrect recognition. To increase the learning ability of the network, new neurons are created in the middle layer described as the former section. In order to prevent the network unlimited growing, new neurons are generated according to a probabilistic rule. The probability *P* of a new neuron being created is given by

$$P = \sum_i e_i \cdot \left(\frac{N^h_{max} - N^h}{N^h_{max}} \right) \tag{23}$$

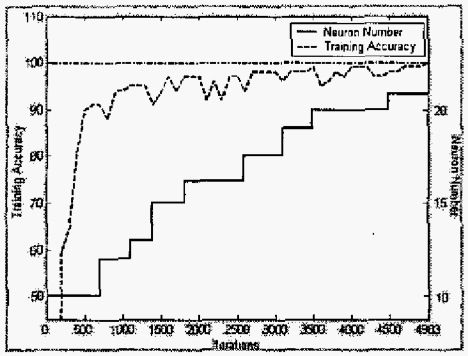

Figure 5. Training of DSNN based OCR.

Table 2
Comparison of Test Results.

Network Type	Network Size	Avg. Accuracy	Avg. Training Time (Sec)
DSNN	11x26x8	94%	5.95
CPN	11x50x8	91.33%	2.97
MLP	11x10x8	93.6%	824

where e_i is the error of the output neuron i, N^h is the current number of the hidden neurons in the middle layer. N^h_{max} is the maximum number of neurons that can be created in the virtual cube space.

For a new generating neuron, its connection, weighting, and bias are randomly generated. The designing DSNN with its new structure is to be training again by the training algorithm mentioned above before the new network serve.

G. Convergence Criterion

The training algorithm stops iterating either when the percentage of recognition error has been reaches the preset value or when a maximum number of iterations have been conducted. After the training process is completed, the network is tested on a new set of the test data to evaluate the performance of the network.

III. Simulation Results

A set of the numeral characters with 300 samples is prepared for training and testing the DSNN. Each sample is a handwritten numeral character, with ranges between '0' to '9'. These data are in random order and separated into two groups, one for the training set containing 200 samples, the other for the set with 100 testing samples for evaluation. Wavelet descriptor of resolution level 3 is calculated for each sample, and then encoded and saved in a text file for simulator to load and decode for training or testing.

In the initial stage, the input layer has 11 neurons, the output layer has 8 neurons, and hidden layer has 10 neurons. The hyperbolic tangent transfer function is adopted as the activation function of every neuron. During the training, the program will record the data of the training time, recognition accuracy, and neuron number of the hidden layer. After the training process completed, the network is tested by the testing data set. The convergence of the training process is illustrated in fig. 5. From fig. 5, it shows that the DSNN produces new neurons in the middle of the training stage to increase the recognition accuracy of the network. In table 2, it shows that DSNN not only be able to provide a smaller size of the middle layer than of other networks, but also provide a better recognition accuracy and acceptable training time than of other networks.

IV. Conclusion

In this paper, a new dynamic structure neural network has been presented. The network can solve the numeric character recognition problem and its performance is better than of other popular neuron network with small network size and fast training. Especially, it requires almost zero designing time to implement the network for applications. During training, the proposed training algorithm is able to reach a well network structure by creating new neurons and deleting the redundant neurons to enhance the system capability in unsupervised manner. Finally, the result of the testing applies wavelet descriptors of resolution level 3 for DSNN to classify the symbol image. From the simulation results, DSNN provides the 96.91% recognition accuracy on the numerical OCR problem.

References

[1] J. A. Freeman and D. M. Skapura, *Neural network algorithms applications and programming techniques*, Addison-Weslwy, 1992.

[2] H. Kim and K. Nam, "Object recognition of one-DOF tools by a back-propagation neural net," *IEEE Trans. Neural Networks*, vol. 6, pp. 484-487, Mar, 1995.

[3] H. I. Avi-Itzhak, T. A. Diep, and H. Garland, "High accuracy optical character recognition using neural networks and centroid dithering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 218-224, Feb, 1995.

[4] P. Wunsch and A. F. Laine, "Wavelet descriptors for multiresolution recognition of handprinted characters," *Pattern Recogn.*, vol. 28, pp. 1237-1249, 1995.

[5] I. P. Morns and S. S. Dlay, "Character recognition using Fourier descriptors and a new form of dynamic semi-supervised neural network," *Microelectron. J.*, vol. 28, pp. 73-84, 1997.

[6] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," IEEE *Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 674-693, July 1989.

[7] P. Antognetti, V. Milutinovic, *Neural Networks Concepts, Applications, and Implementation*, Volume I, Prentice Hall, 1991.

[8] P. Antognetti, V. Milutinovic, *Neural Networks Concepts, Applications, and Implementation*, Volume II, Prentice Hall, 1991.

[9] S. Haykin, *Neural Networks, A Comprehensive Foundation*, Prentice Hall, 1994.

[10] S. W. Moon, S. G. Kong, "Block-based neural networks," *IEEE Trans. Neural Networks*, vol 12, pp 307-317, 2001.

[11] G. Ming, S. T. Wong, J. Henley, X. Yuan, H. Song, N. C. Spitzer, and M. Poo, "Adaptation in the chemotactic guidance of nerve growth cones," *Nature 417*, pp 411-418, 2002.

[12] R. J. Pasterkamp, J. J. Peschon, and M. K. Spriggs, A. L. Kolodkin, "Semaphorin 7A promotes axon outgrowth through integrins and MAPKs," *Nature 424*, pp 398-405, 2003.

[13] V. H. Hker, D. Shewan, M. Tessier-Lavigne, M. Poo, and C. Holt, "Growth-cone attraction to netrin-1 is converted to repulsion by laminin-1," *Nature 401*, pp 69-73, 1999.

[14] J. P. Himanen, K. R. Rajashankar, M. Lackmann, C. A. Cowan, M. Henkemeyer, and D. B. Nikolov, "Crystal structure of an Eph receptor-ephrin complex," *Nature 414*, pp 933-938, 2001.

[15] R. J. Giger and A. L. Kolodkin, "Neurobiology: Frazzled precision guides axons," *Nature 406*, pp 841-843, 2000.