A Novel Dynamic Structural Neural Network with Neuron-Regeneration and Neuron-Degeneration Mechanisms

Yingtung Hsiao[1], Chenglong Chuang[1], Joeair Jiang[2], Chiang Wang[1], and Chengchih Chien[1]

[1] Department of Electrical Engineering
Tamkang University, Taipei 251, Taiwan, China
{hsiao,clchuang,cwang,chien}@ee.tku.edu.tw
[2] Department of Bio-Industrial Mechanical Engineering
National Taiwan University, Taipei 106, Taiwan, China
jajiang@ccms.ntu.edu.tw

Abstract. This paper presents a novel neural network model with free structure style called dynamic structural neural network (DSNN). The neurons are randomly generated in a virtual three-dimensional space. The structure of the DSNN is reconfigurable for increasing the learning capacity of the network. Therefore, in this work, a structure reconfiguration algorithm is also proposed to achieve this goal. Finally, several simple pattern recognition problems are applied to the proposed DSNN to demonstrate the efficiency and performance of the network.

1 Introduction

Artificial neural networks have been successfully applied to different kind of research fields and applications due to their model-free approximation capability to complex decision making process [3]. The alignment of the neurons of the artificial neural networks is layered structure.

In present time, most of the neural networks are constructed in layer structure. The performance of this kind of structure might be degraded by the fixed structure. Compare to the real life nerve system, the structure of the conventional neural networks are too simple and form in regular alignment patterns. The performance of the neural networks may be limited by the regular alignment patterns. The only way to solve the bad performance problem is to redesign the entire neural systems. However, the costs to redesign the neural systems are too high. The automatic structure optimization algorithm of the neural network is needed to improve the performance of the neural networks.

In this article, we developed an automatic neural network generating algorithm and its learning algorithm to optimize the structure of the proposed neural network. All generated neurons shall be deployed in a virtual three-dimensional (3-D) space. Moreover, the neuron-regeneration mechanism and the neuron-regeneration mechanism, which are developed according to the neuron-regeneration technique of the real life nerve system in the territory of medical science, are proposed to automatically reconstruct the neural network for achieving higher performance.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3496, pp. 380–384, 2005.

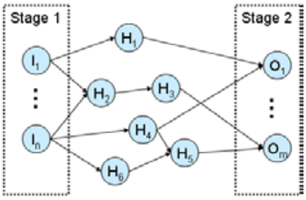

Fig. 1. Structure of the dynamic structure neural network. (DSNN)

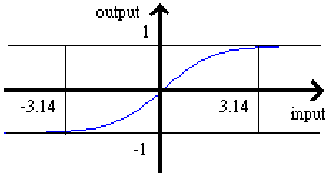

Fig. 2. Transfer function of the DSNN

This paper is organized as follows: In Section 2, the architecture of DSNN is described. And the structure optimization method of the proposed neural network is presented in Section 3. The testing results are showed in Section 4. Conclusions are given in Section 5.

2 Dynamic Structural Neural Network

The DSNN model constructs in a virtual 3-D space. Figure 1 represents the structure of the DSNN model. Each hidden neuron is labeled as H_j, where j=1,2,…,6. The stage 1 and the stage 2 represent the input and the output layers respectively. And n and m indicate numbers of neurons of input and output layers of the DSNN. The traditional artificial neural networks require trial-and-error processes for designing of an optimal structure for an artificial neural network under a given problem. The designer's experiences play an important role in designing of a neural network. Therefore, in this work, the structure and neural connections of DSNN are automatically generated to avoid the trial-and-error problem.

The activation function of a single neuron can be formulated as follow:

$$v_o(n) = \sum_{i \in C_o} w_{io}(n) y_i(n) + b_o(n) \tag{2}$$

$$y_o(n) = \varphi_o(v_o(n)) \tag{3}$$

where v_o is the internal activity level of the neuron o, w_{io} is weight value of neural connection from the hidden neuron i to the neuron o, b_o is the bias value of the neuron o, C_o is the neural connection set of the neuron o, φ_o represents the transfer function of the neuron o, and y_o is the actual output of the neuron o.

Because the output neurons can be trained by the target vector while we train the neural network, the weight vector of the output neuron can be tuned simply by BPN-like learning algorithm. The weight and bias tuning algorithm for output neuron of DSNN is described as follow:

$$e_o(n) = d_o(n) - y_o(n) \tag{4}$$

$$\Delta w_{io}(n) = l \cdot \eta \cdot e_o(n) \,|\, y_o(n) \,| \tag{5}$$

$$\Delta b_o(n) = \eta \cdot e_o(n) \tag{6}$$

$$l = \begin{cases} 1 & if\ \Delta y_o(n) > 0 \\ -1 & if\ \Delta y_o(n) < 0 \end{cases} \tag{7}$$

where e_o is the error value of the output neuron o, d_o is the target value of the output neuron o, Δw_{io} is the tuning value of the weight of the neural connection from the hidden neuron i to the output neuron o, Δb_o is the tuning value of the bias of the output neuron o, and η is the learning rate of the network.

Because the hidden neurons and their neural connections are randomly generated, the structure of the hidden neurons is very complex and unable to tune by any existing learning algorithm. Therefore, we provided a tuning indicator for tuning the weight and the bias of the hidden neurons. The tuning indicator is defined as below:

$$g_i(n) = \eta \cdot e_o(n) \,|\, y_o(n) \,| \tag{8}$$

where g_i is the tuning indicator of the hidden neuron i that connected to output neuron o. For the hidden neurons, the tuning of the weight and the bias are depending on the tuning indicator g_i. According to the tuning indicator g_i, we can get the average tuning indicator s_i by following equation:

$$s_i(n) = \frac{g_i(n)}{Nc} \tag{9}$$

where s_i is the average tuning indicator that use for determining the tuning range of the weight and the bias of the hidden neuron i. Nc is the total number of neurons that connected to the hidden neuron i. Since the si is obtained, the weight and bias of the output neuron of DSNN can be tuned by applying the following equations:

$$\Delta w_{ji}(n) = l \cdot \eta \cdot s_i(n) \,|\, y_i(n) \,| \tag{10}$$

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

$$\Delta b_i(n) = \eta \cdot s_i(n) \tag{12}$$

where Δw_{ji} is the tuning value of the weight of connection from hidden neuron j to hidden neuron i, and Δb_i is the tuning value of the bias of the output neuron i.

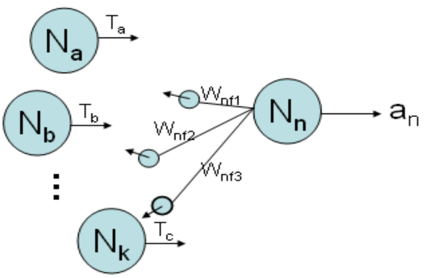

Fig. 3. The concept of the neuron-regeneration

Table 1. The Growing Rules of Free Connectors

W_L	a_n	T_R	Att/Exc
+	+	+	Att
+	+	-	Rep
+	-	+	Rep
+	-	-	Att
-	+	+	Rep
-	+	-	Att
-	-	+	Att
-	-	-	Rep

3 Automatically Structural Optimization Technique

Figure 3 shows the concept of the neuron-regeneration process; where for each neuron N_n has three free connectors. Each of them has assigned a randomly generated weighting, such as W_{nf1}, W_{nf2} and W_{nf3} in Figure 4. When the DSNN is under training process, the free connectors shall search for near neurons and try to create new connections. Table 1 shows the rules of determining the growing direction of free connectors, where a_n is the output of the neuron that belong to the corresponding free connector, WL is the weight of the free connector, TR is the tuning indicator of the near neuron, and the Att means the free connectors will be attracted to the position of the near neuron, the Rep means the free connectors will be distracted from the position of the near neuron. The attraction or repulsion process is described as below:

$$\Delta(x_{fn}, y_{fn}, z_{fn}) = D \cdot \frac{|g_j|}{L} \cdot (x_j, y_j, z_j) \tag{13}$$

$$D = \begin{cases} 1 & \text{if Attraction} \\ -1 & \text{if Repulsion} \end{cases} \tag{14}$$

where (x_{fn}, y_{fn}, z_{fn}) is the current coordinate of the free connector, g_i is the tuning indicator of the near neuron, L is the distance between the original neuron of the free connector and the near neuron, (x_j, y_j, z_j) is the coordinate of the near neuron. If the distance from free connector to any neuron of the system is less then 1, a new neural connection is established between closest neuron and the neuron that own the free connector, and the free connector is then reinitialized.

When the DSNN is under training process, some weight of the connections might be very small. If a neural connection that the weight value is very small for several iterations, then this neural connection will be defined as unnecessary connection that shall be pruned from the system to save the computational resources. The mechanism of the dynamic growth of the middle layer allows the network to increase it's learn ability by inventing new neurons. The probability P of a new neuron being created is given by:

$$P = \sum_i e_i \cdot \left(\frac{N_{\max\_h} - N_h}{N_{\max\_h}} \right) \tag{15}$$

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. $N_{\max\_h}$ is the maximum number of neurons that can be created in the virtual 3-D space. In (22), if output error of the network is high, the probability of generating a new neuron is high. After a new neuron generating, N is increased by 1. As long as N reaches the maximum number of neurons $N_{\max\_h}$, the probability P will reduce to 0 to prevent the unlimited growing of the hidden neurons.

4 Experimental Results

In this work, we have applied a problem to the proposed DSNN to verify the algorithm. The problem is a specific function as below:

$$Z = Y \cdot \exp^{-(X^2+Y^2)} \tag{16}$$

where X and Y are input values, and Z is the output value. The ideal output graph is shown in Figure 4. The number of neurons of the initially generated DSNN is 150. The generation of the training loop is 200. Since the simulating program has completed the training process, the DSNN created 2 new neurons and 12684 connections where 12595 connections were pruned by the restructure optimal program.

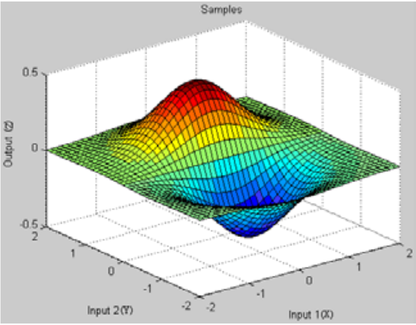

Fig. 4. The target graph of (16)

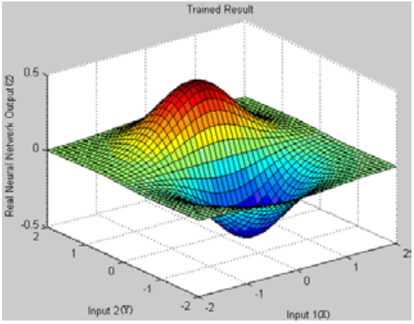

Fig. 5. The DSNN output graph of problem

5 Conclusions

A novel three-dimensional dynamic structure neural network architecture is proposed in this paper. The neurons and connections of DSNN are randomly generated to avoid the experience-dependence problem while the designers trying to design a properly structure for an artificial neural network under a given problem. The structure of DSNN is optimized automatically by the neuron-regeneration and the neuron-degeneration mechanisms. The DSNN also can invent new neurons into network for enhancing the performance of the DSNN. The proposed DSNN model could be applied to solve other different kind of problems in the future.

References

1. Matlab User's Guide, (1992)
2. Haykin, S: Neural Networks: A Comprehensive Foundation. Prentice Hall (1994)
3. Moon, S. W., Kong, S. G.: Block-based Neural Networks. IEEE Trans. Neural Networks, **12** (2001) 307-317
4. Ming, G., Wong, S.T., Henley, J., Yuan, X., Song, H., Spitzer, N.C., M. Poo.: Adaptation in the Chemotactic Guidance of Nerve Growth Cones. Nature, **417** (2002) 411-418
5. Pasterkamp, R.J., Peschon, J.J., Spriggs, M.K., Kolodkin, A.L.: Semaphorin 7A Promotes Axon Outgrowth Through Integrins and MAPKs. Nature **424** (2003) 398-405