Logical Combinations of Neural Networks

Lapas Pradittasnee*, Arit Thammano**, and Suthichai Noppanakeepong*

*Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang,
Ladkrabang, Bangkok 10520 Thailand

**Advanced Computer Application and Design Research Group
Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang,
Ladkrabang, Bangkok 10520 Thailand
Tel: (662)737-2551-4 ext. 529, Fax: (662)326-9074
E-mail: arit@it.kmitl.ac.th

Abstract: In general, neural networks based modeling involves trying multiple networks with different architectures and/or training parameters in order to achieve the best accuracy. Only the single best-trained neural network is chosen, while the rest are discarded. However, using only the single best network may never give the best solution in every situation. Many researchers, therefore, propose methods to improve the accuracy of neural networks based modeling. In this paper, the idea of the logical combinations of neural networks is proposed and discussed in detail. The logical combination is constructed by combining the corresponding outputs of the neural networks with the logical "And" node. The experimental results based on simulated data show that the modeling accuracy is significantly improved when compared to using only the single best-trained neural network.

1. Introduction

Neural networks based modeling often involves trying multiple networks with different architectures, such as different numbers of hidden layers and/or different numbers of neurons per layer, and training parameters, such as the learning rate values or the initial conditions, in order to achieve the best accuracy. Typically, the trained neural network that yields the lowest mean square error (MSE) is chosen as best, while the rest are discarded. However, many literatures reported remarkable success using an ensemble of trained neural networks, instead of simply using only the best neural network, e.g., Hansen and Salamon [1], and Baxt [2] used an ensemble of trained neural network to solve the classification problems. In the function approximation area, Hashem et al. [3][4] combined the corresponding outputs of a number of trained neural networks and illustrated that using optimal linear combinations can significantly improve model accuracy compared to the use of the single best network or the simple averaging of the corresponding outputs of the component networks. The objective of this paper is to propose another technique to improve the modeling accuracy by using the logical combinations. An overview of logical combinations of neural networks is discussed in the following section. Then an experimental study is conducted to examine the effectiveness of logical combinations in improving model accuracy for well-trained sub-networks.

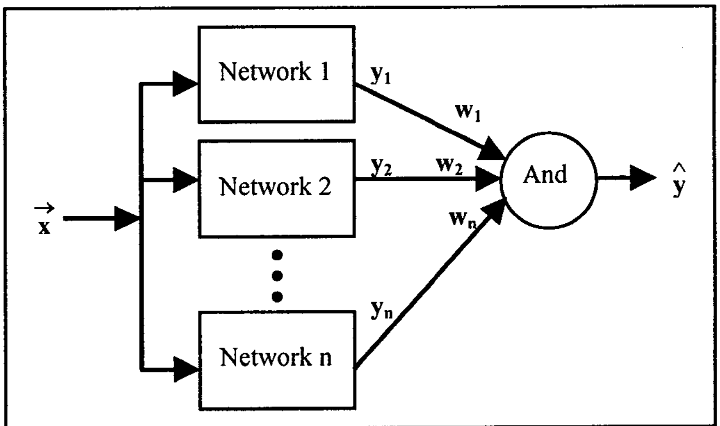

Figure 1. Logical combination of the outputs of n-trained neural networks

2. Logical Combinations of Neural Networks

The logical combination is constructed by combining the corresponding outputs of n-trained neural networks with the logical "And" node (Figure 1). In the training process, unlike in the backpropagation network where every weight in the network must be modified at each iteration, only the weight on a single active path will be modified. The active path (j) is the path that the weighted output of the sub-network is minimum. In this study, a multiple-input single-output mapping is considered. For a given training data$(\vec{x}, T(\vec{x}))$, the output of the combined model, $\hat{y}(\vec{x}, \vec{w}(t))$, is given by

$$y_i(\vec{x}, \vec{w}(t)) = w_i(t) y_i(\vec{x}) \quad (1)$$

$$e_i(\vec{x}, \vec{w}(t)) = T(\vec{x}) - y_i(\vec{x}, \vec{w}(t)) \quad (2)$$

$$e_j(\vec{x}, \vec{w}(t)) = \bigwedge_{i=1}^{n} e_i(\vec{x}, \vec{w}(t)) \quad (3)$$

$$\hat{y}(\vec{x}, \vec{w}(t)) = y_j(\vec{x}, \vec{w}(t)) \quad (4)$$

$$w_j(t+1) = w_j(t) + \alpha e_j(\vec{x}, \vec{w}(t)) \quad (5)$$

Where α is the learning rate; $e_j(\vec{x}, \vec{w}(t))$ and $y_j(\vec{x}, \vec{w}(t))$ are the error and the weighted output of the active path.

3. Process Description

In this section, the effectiveness of the proposed logical combination in improving model accuracy is examined. The function that is selected to test the performance of the combined network is [5]:

$$f(x, y) = 0.1 + \frac{1.0 + \sin(2x + 3y)}{3.5 + \sin(x - y)} \quad (6)$$

The graphical representation of this function is shown in Figure 2. An experiment is conducted using four (n=4) sub-networks (NN1, NN2, NN3, and NN4). NN1 is 2-9-1 NN (one hidden layer with nine hidden neurons). NN2 and NN4 are 2-6-1 NNs. NN3 is 2-5-1 NN. The activation function for the hidden neurons and the output neuron of NN1 is the sigmoid function. The activation function for the hidden neurons and the output neuron of NN2 is the gaussian function. The activation functions for the hidden neurons and the output neuron of NN3 are the sigmoid and the gaussian functions respectively. The activation functions for the hidden neurons and the output neuron of NN4 are the sigmoid and the linear functions respectively. Table 1 summarizes the configurations of the four NNs. All four NNs are trained using a common training set that is constructed from the nonlinear Equation (6) by sampling in two dimensions at equally spaced grid points, an interval of 0.1 being used for both x and y. The range of f(x, y) is [0, 1]. Training is carried out using the backpropagation algorithm. When all four NNs are well-trained, the same data set is again used in computing the combination-weights according to Equation (5).

4. Experimental Results

The training and testing results are shown in Table 2 and 3. From the training result, the best NN among the four trained NNs yields a root mean square error (RMSE) of 0.0077. The logical combined network yields a RMSE of 0.0033, which is 57% less than that of the best NN. A separate data set sampled from the same multivariate

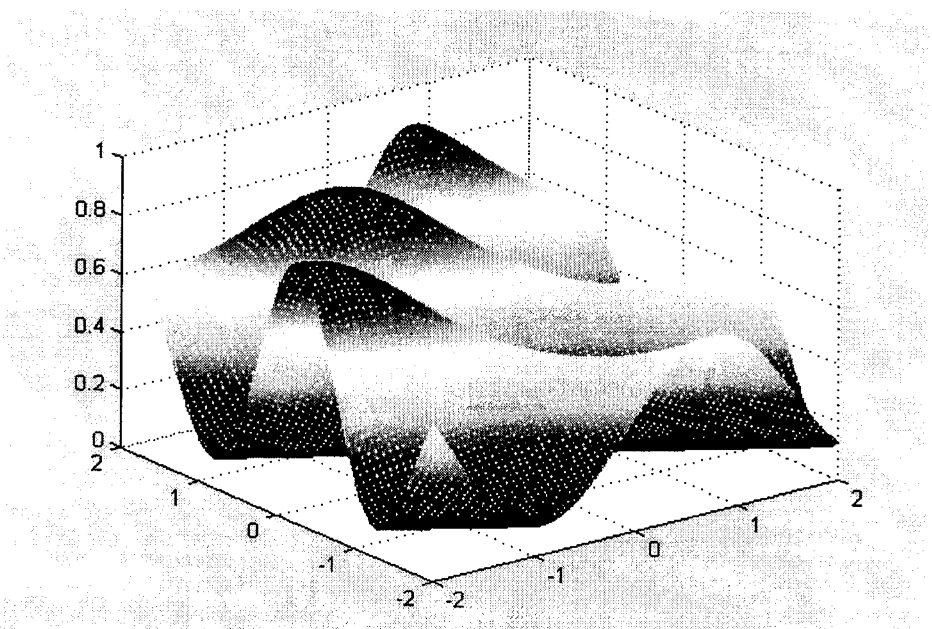

Figure 2. The graphical representation of the test function

Table 1. The configurations of the sub-networks

Network	Hidden Neuron	Hidden Layer Activation Function	Output Neuron	Output Layer Activation Function
1	9	$1/(1+e^{-net})$	1	$1/(1+e^{-net})$
2	6	$e^{-(net)^2}$	1	$e^{-(net)^2}$
3	5	$1/(1+e^{-net})$	1	$e^{-(net)^2}$
4	6	$1/(1+e^{-net})$	1	net

Table 2. The training result

Data No.	Target	Network 1	Network 2	Network 3	Network 4	Combination of Network
1	0.385714	0.367851	0.461931	0.369439	0.35966	0.370785
2	0.481017	0.485866	0.461131	0.482448	0.493907	0.484052
3	0.573943	0.593445	0.531617	0.587864	0.605786	0.577244
4	0.656511	0.673652	0.6435	0.670107	0.688232	0.655345
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
120	0.100021	0.085488	0.082422	0.083375	0.090868	0.103221
121	0.111736	0.063204	0.054303	0.062129	0.087503	0.099506
SSE =		0.00899986	0.037999	0.0072	0.0185	0.001354
RMSE =		0.008624327	0.017721211	0.007713892	0.012364973	0.003345158

Table 3. The testing result

Data No.	Target	Network 1	Network 2	Network 3	Network 4	Combination of Network
1	0.456401	0.443439	0.499584	0.445668	0.437278	0.446533
2	0.547827	0.556897	0.525692	0.55303	0.563647	0.549086
3	0.630477	0.646077	0.611117	0.640678	0.657227	0.636592
4	0.696903	0.704161	0.710382	0.70157	0.71723	0.694041
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
99	0.109508	0.10648	0.111252	0.103551	0.104425	0.110344
100	0.100202	0.077017	0.076118	0.073982	0.081699	0.100086
SSE =		0.00299996	0.020500167	0.002099327	0.008999952	0.000505496
RMSE =		0.00547719	0.014317879	0.004581841	0.009486808	0.002248324

distribution is used in testing the robustness of the combined network. From the testing result, the best NN yields a RMSE of 0.0046. The logical combined network yields a percentage reduction of 52% in the RMSE compared to using the single best NN.

5. Conclusions

In this paper, the logical combination of standard feedforward neural networks is proposed. The simulation results show that the combined network has superior accuracy compared to using only the single best neural network. Combining the trained networks can help integrate the knowledge acquired by the component networks, therefore produces superior model accuracy compared to using only the single best-trained neural network.

References

[1] L. K. Hansen and P. Salamon, "Neural network ensembles," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 10, pp. 993-1001, 1990.

[2] W. G. Baxt, "Improving the accuracy of an artificial neural network using multiple differently trained networks," Neural Computation, Vol. 4, pp. 772-780, 1992.

[3] S. Hashem and B. Schmeiser, "Approximating a function and its derivatives using MSE-optimal linear combinations of trained feedforward neural networks," Proceedings of the 1993 World Congress on Neural Networks, New Jersey, pp. I:617-620, 1993.

[4] S. Hashem, Y. Yih, and B. Schmeiser, "An efficient model for product allocation using optimal combinations of neural networks," Intelligent Engineering Systems through Artificial Neural Networks, Vol. 3, pp. 669-674, ASME Press, 1993.

[5] T. Masters, Practical Neural Network Recipes in C++, Academic Press, San Diego, CA., 1993.