Bull. Coll. Med. Sci. Tohoku Univ. **11**(1) : 1〜6, 2002

A General Second-order Neural Unit

Noriyasu HOMMA[1,2] and Madan M. GUPTA[2]

[1]*Department of Radiological Technology, College of Medical Sciences, Tohoku University*
[2]*Intelligent Systems Research Laboratory, College of Engineering, University of Saskatchewan*

一般化2次ニューラルユニット

本 間 経 康[1,2], Madan M. Gupta[2]

[1]東北大学医療技術短期大学部　診療放射線技術学科
[2]サスカッチワン大学工学部　知的システム研究所

Key words: Neural Networks, Second-order Systems, Learning Algorithms, Classification, Function Approximation

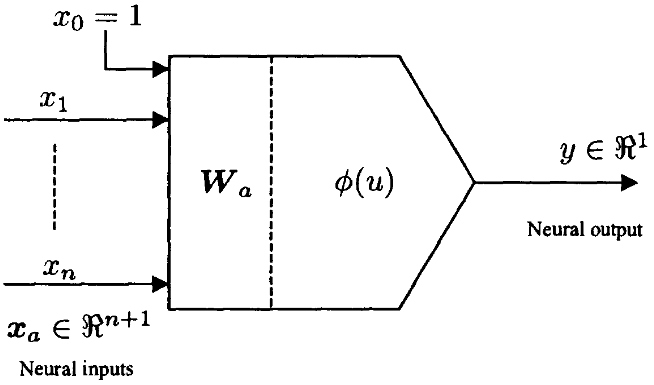
A general *second-order neural unit* (SONU) is developed using a new matrix form for providing a general *second-order combination* of the input signals and the synaptic weights. It is shown that the widely used linear combination neural units are only a subset of the proposed SONUs. Simulation studies demonstrate that the learning and generalization abilities of the proposed SONUs are much superior to that of the linear combination neural units.

Introduction

The neural networks, consisting of the *first-order* neurons which provide us with the neural output as a nonlinear function of the *linear combination* of the neural inputs and the synaptic weights, have been successfully used in various applications such as in pattern recognition or classification, system identification, adaptive control, optimization and signal processing[1)2)3)].

The *higher-order combination* of the inputs and the weights will yield the higher neural performance. However, one of disadvantages encountered in the previous development of the higher-order neural units is the larger number of parameters (weights) corresponding to the higher-order nonlinearity of the input features space[4)]. To optimize the features space, a learning capability assessment method has been proposed by Villalobos and Merat[5)].

In this paper, in order to reduce the number of parameters without loss of the higher performance, a second-order neural unit (SONU) is proposed. Using a novel general matrix form of the second-order operation, the SONU provides us with the output as a nonlinear function of the *second-order combination* of the input signals and the synaptic weights. Simulation studies illustrate the usefulness of the SONU by



$$u = \boldsymbol{x}_a^T \boldsymbol{W}_a \boldsymbol{x}_a$$
$$y = \phi(u)$$

where,

$$\boldsymbol{x}_a = [x_0, x_1, \cdots, x_n]^T \in \Re^{n+1}, \quad x_0 = 1$$

$$\boldsymbol{W}_a = \begin{bmatrix} w_{00} & w_{01} & \cdots & w_{0n} \\ 0 & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{nn} \end{bmatrix}$$

Figure 1. A *second-order neural unit* (SONU) defined by Eqns. (1) and (2).

using both the pattern classification and function approximation problems.

2 Formulation of the Second-Order Neural Unit (SONU)

A novel *second-order* neural unit with the n-dimensional neural inputs, $\boldsymbol{x} \in \Re^n$, and a single neural output, $y \in \Re^1$, is developed in this paper (Fig. 1). Let $\boldsymbol{x}_a = [x_0\ x_1 \cdots x_n]^T \in \Re^{n+1}$, $x_0 = 1$ be an augmented neural input vector.

Here, a new second-order aggregating formulation is proposed by using an augmented weight matrix $\boldsymbol{W}_a \in \Re^{(n+1)\times(n+1)}$ as

$$u = \boldsymbol{x}_a^T \boldsymbol{W}_a \boldsymbol{x}_a \tag{1}$$

then the neural output, y, is given by a nonlinear function of the variable u as

$$y = \phi(u) \in \Re^1 \tag{2}$$

Since both the elemental weights w_{ij} and w_{ji}, i, $j \in \{0, 1, \cdots, n\}$ yield the same second-order term $x_i x_j$ (or $x_j x_i$), an upper triangular (or lower triangle) matrix is used in this paper, and the upper triangular matrix can give the general second-order combination as

$$u = \boldsymbol{x}_a^T \boldsymbol{W}_a \boldsymbol{x}_a = \sum_{i=0}^{n} \sum_{j=i}^{n} w_{ij} x_i x_j \tag{3}$$

Note that the conventional first-order (linear combination) aggregation form is only a special case of this second-order matrix form. For example, the special weight matrix (row vector), $\boldsymbol{W}_a \doteq Row[w_{00}\ w_{01} \cdots w_{0n}] \in \Re^{(n+1)\times(n+1)}$, can produce the exactly equivalent linear combination, $u = \sum_{j=0}^{n} w_{0j} x_j$. Therefore, the proposed neural model with the second-order matrix operation is a more general and, for this reason, it is called general *second-order neural units* (SONUs).

The learning algorithm for the SONUs can be defined by

$$\boldsymbol{W}_a(k+1) = \boldsymbol{W}_a(k) + \Delta \boldsymbol{W}_a(k) \tag{4}$$

Here, k is the learning iteration and $\Delta \boldsymbol{W}_a$ denotes the changes in the weight matrix given by[6)]

$$\Delta \boldsymbol{W}_a(k) \doteq \gamma e(k) \boldsymbol{x}_a(k) \boldsymbol{x}_a^T(k) \tag{5}$$

where, $e(k) = y(k) - y_d(k)$ is the error between the neural output, $y(k)$, and the desired output, $y_d(k)$, and $\gamma = \eta \phi'(u)$, $\eta > 0$. Note that, taking the average of the changes for some input vectors, the changes in the weights, $\Delta w_{ij}(k)$, implies the correlation between the error $e(k)$ and the corresponding inputs term $x_i(k) x_j(k)$.

3 Simulation Studies

3.1 XOR problem

Since the two-input XOR function is not

linearly separable, it is one of the simplest logic functions that cannot be realized by a single *linear combination neural unit*. Therefore, it requires multilayered neural networks structure consisting of the linear combination neural units.

On the other hand, a single SONU can solve this XOR problem by using its general second-order functions such as in (3). To implement the XOR function using a single SONU, the four learning patterns corresponding to the four combinations of the two binary inputs $(x_1, x_2) \in \{(-1, -1), (-1, 1), (1, -1), (1, 1)\}$ and the desired output $y_d = x_1 \oplus x_2 \in \{-1, 1\}$ were given to the SONU.

For the XOR problem, the neural output, y, is defined by the signum function as $y = \phi(u) = Sgn(u)$. The correlation learning algorithm with a constant gain, $\gamma = 1$, in (5) was used in this case. The learning was terminated as soon as the error converged 0. Since the SONU with the signum function classifies the neural input data by using the second-order nonlinear function of the neural inputs $\boldsymbol{x}_a^T \boldsymbol{W}_a \boldsymbol{x}_a$ as in (1), many nonlinear classification boundaries are possible such as a hyperbolic boundary and an elliptical boundary (Table 1).

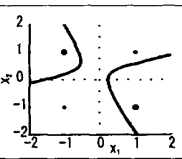
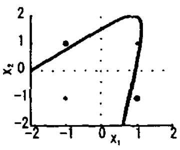
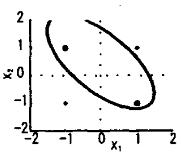
Note that the results of the classification boundary are depended on the initial weights (Table 1), and any classification boundary by the second-order functions can be realized by a single SONU. This realization ability is obviously superior to the linear combination neural unit which cannot achieve such nonlinear classification ; at least three linear combination neural units in a layered structure are needed to solve this problem.

Secondly, the number of parameters (weights) required for solving this problem can be reduced by using the SONU. In this simulation study, because of by using the upper triangular weight matrix, the number of parameters including the threshold required for the SONU was only 6, compared with at least 9 parameters required for the layered structure with three linear combination neural units.

3.2 Function approximation problem

For evaluating function approximation ability of the SONUs, an example was taken from Klassen's function approximation prob-

Table 1. Initial weights ($k=0$), final weights and the classification boundaries for the XOR problem.

k	w_{00}	w_{01}	w_{02}	w_{11}	w_{12}	w_{22}	Boundaries
	(A hyperbolic boundary)						
0	0.323	−0.870	−0.153	0.977	0.031	−0.332	
4	−0.177	0.630	0.347	0.477	−1.469	−0.832	
	(A hyperbolic boundary)						
0	−0.773	0.818	0.748	0.793	−0.525	0.369	
4	−1.023	0.568	0.498	0.543	−0.775	0.119	
	(An elliptical boundary)						
0	0.847	0.397	0.779	−0.996	−0.961	−0.803	
3	0.947	0.497	0.679	−0.896	−1.061	−0.703	

lem[7]. The task consists of learning a representation for an unknown, one-variable nonlinear function, $F(x)$, with the only available information being the 18 sample patterns[5].

For this function approximation problem, a two layered neural network structure was composed of two SONUs in the first layer and a single SONU in the output layer (Fig. 2). The nonlinear activation function of the SONUs in the first layer was defined by a bipolar sigmoidal function as $\phi(u)=(1-\exp(-u))/(1+\exp(-u))$, but for the single output SONU, instead of the sigmoidal function, the linear function was used, that is, $y=\phi(u)=u$. The gradient learning algorithm with $\eta=0.1$ was used for this problem.

The mapping function obtained by the SONU network after 10^7 learning iterations appears in Fig. 3. In this case, the average square error taking over 18 patterns was 4.566E-6. The fact that the approximation accuracy shown in Fig. 3 is extremely high is an evidence for the high approximation ability of the proposed SONU network.

Klassen et al.[7] used 5 particular trigonometric functions, $\sin(\pi x)$, $\cos(\pi x)$, $\sin(2\pi x)$, $\cos(2\pi x)$ and $\sin(4\pi x)$ as special features of the extra neural inputs. Also, it has been reported by Villalobos and Merat[5] that the term $\cos(\pi x)$ is not necessary to achieve a lower accuracy within the error tolerance 1.125E-4, but still 4 extra features were required.

On the other hand, in this study the high approximation accuracy of the proposed SONU network was achieved by only two SONUs in the first layer and a single SONU in the output layer, and no special features were required to obtain the high accuracy. These are remarkable advantages of the proposed SONU network.

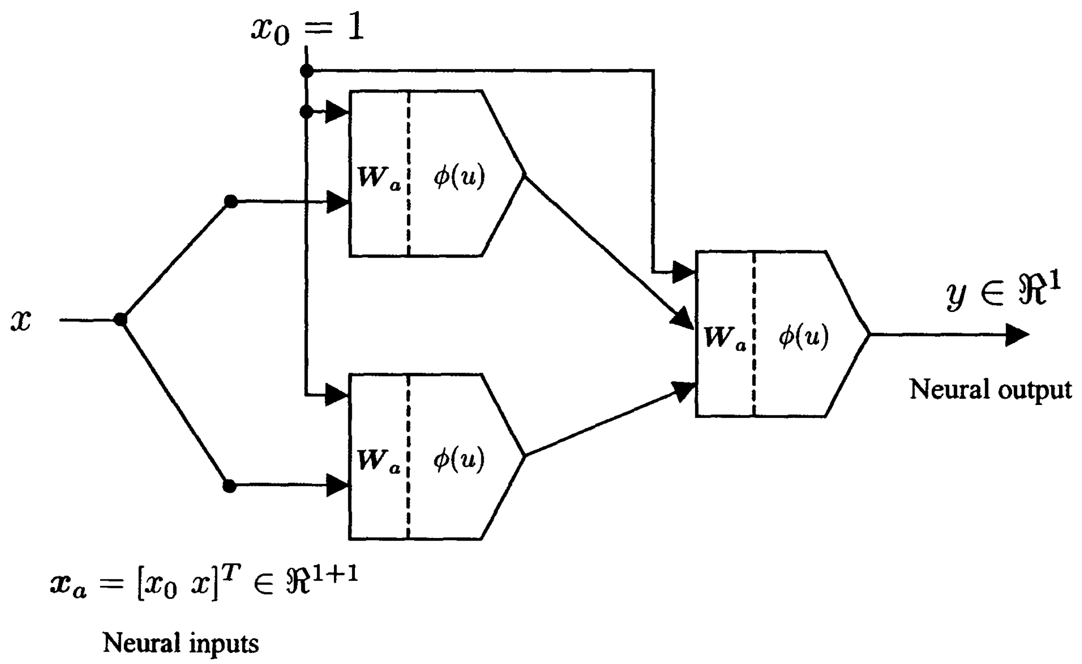


Figure 2. A two layered neural network structure with two SONUs in the first layer and a single SONU in the output layer for the function approximation problem.

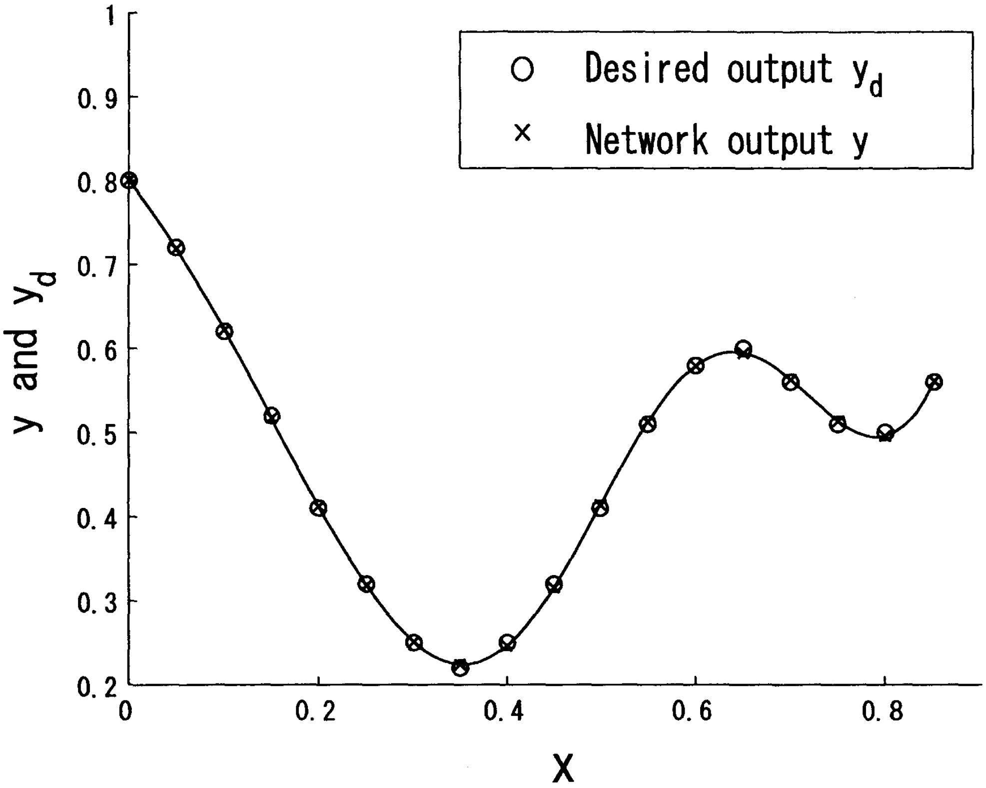


Figure 3. Training pairs and outputs estimated by the network with SONUs for the Klassen's function approximation problem.

4 Conclusions

In this paper, a general second-order neural unit (SONU) has been developed. Simulation studies for both the pattern classification and function approximation problems demonstrated that the learning and generalization abilities of the proposed SONU and networks consisting of SONUs are much superior to that of the widely used linear combination neural units and their networks. Indeed, it has been investigated that the linear combination neural units used widely in multilayered neural networks are only a subset of the proposed SONUs. Some extensions of these concepts to the radial basis function (RBF) networks, fuzzy neural networks and dynamic neural units are in progress.

References

1) Sinha, N., Gupta, M., Zadeh, L.: Soft Computing and Intelligent Control Systems: Theory and Applications. Academic Press, New York (1999)

2) Narendra, K., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. IEEE Trans. Neural Networks **1** (1990) 4-27

3) Cichochi, A., Unbehauen, R.: Neural Networks for Optimization and Signal Processing. Wiley, Chichester (1993)

4) Schmidt, W., Davis, J.: Pattern recognition properties of various feature spaces for higher order neural networks. IEEE Trans. Pattern Analysis and Machine Intelligence **15** (1993) 795-801

5) Villalobos, L., Merat, F.: Learning capability assessment and feature space optimization for higher-order neural networks. IEEE Trans.

Neural Networks **6** (1995) 267-272
6) Homma, N., Gupta, M.: Development of a general second-order neural unit with applications. In: Proc. of the 15th IFAC World Congress, Barcelona (2002) (submitted)
7) Klassen, M., Pao, Y., Chen, V.: Characteristics of the functional link net: a higher order delta rule net. In: Proc. of IEEE 2nd Annual Int'l. Conf. Neural Networks, San Diego (1988)