Proceedings of 1993 International Joint Conference on Neural Networks

An Efficient Multilayer Quadratic Perceptron for Pattern Classification and Function Approximation

BAO-LIANG LU, YAN BAI, HAJIME KITA and YOSHIKAZU NISHIKAWA
Department of Electrical Engineering, Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-01, Japan
E-mail: lbl@rotary.kuee.kyoto-u.ac.jp

Abstract: *We propose an architecture of a multilayer quadratic perceptron (MLQP) that combines advantages of multilayer perceptrons(MLPs) and higher-order feedforward neural networks. The features of MLQP are in its simple structure, practical number of adjustable connection weights and powerful learning ability. In this paper, the architecture of MLQP is described, a backpropagation learning algorithm for MLQP is derived, and the learning speed of MLQP is compared experimentally with MLP and other two kinds of the second-order feedforward neural networks on pattern classification and function approximation problems.*

1. Introduction

The multilayer perceptron(MLP) is one of the most popular architectures of neural networks for pattern recognition etc., trained with a tutored learning. However, in some applications, MLP does not work well in learning and generalization. In order to improve the performance of MLP, several architectures having higher-order connectivity have been proposed [1-3]. While increasing the order of connectivity enhances the computational power of the network, it may not be favorable in learning speed because of increase in adjustable connection weights. Hence it is needed to find efficient ways of introducing higher-order connectivity considering the following points: (a) Computational power of the network, (b) Learning speed and (c) Generalization ability, in practical applications.

In this paper, we propose an architecture of a multilayer quadratic perceptron (MLQP) considering the above discussions. A backpropagation learning algorithm for MLQP is presented, and the learning speed of MLQP is compared experimentally with MLP and other two kinds of the second-order feedforward neural networks(SFNs) on pattern classification and function approximation problems.

2. The MLQP Architecture

The basic idea of MLQP is an introduction of quadratic terms into the net input to a unit. The computational element of MLQP is shown in Fig. 1. The characteristic of the unit in MLQP is defined by

$$net_{kj} = \sum_{i=1}^{N_{k-1}} (u_{kji} x_{k-1,i}^2 + v_{kji} x_{k-1,i}) + bias_{kj} \quad (1)$$

$$x_{kj} = f(net_{kj}) \quad (2)$$
$$\text{for } k = 2, 3, \cdots, M; \quad j = 1, 2, \cdots, N_k$$

where both u_{kji} and v_{kji} are the weights connecting the ith unit in the layer $k-1$ to the jth unit in the layer k, $bias_{kj}$ is the bias of the jth unit in the layer k, N_k is the number of units in the layer $k (1 \le k \le M)$, and $f(\cdot)$ is a sigmoidal activation function.

The features of MLQP are in its simple structure

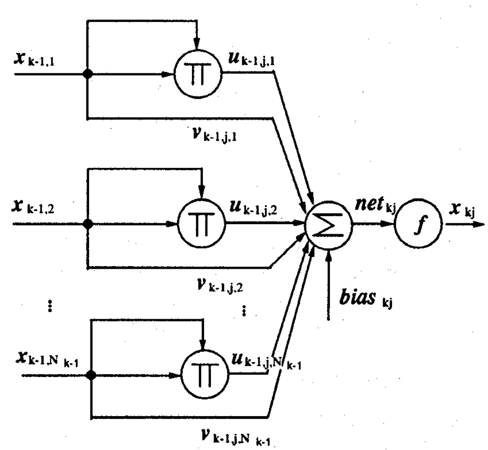


Figure 1: Computational element in MLQP.

and practical number of adjustable weights. If an MLQP and an MLP have the same number of units at each layer, the number of adjustable weights in MLQP is about two times as many as in MLP. Therefore, MLQP can scale well to large problems, similarly to the conventional MLP.

From Eqs. (1) and (2), we see that the decision region of a unit in the hidden and the output layers in MLQP is a hypersphere(including a hyperplane as a special case). This indicates that MLQP can realize both local representation as in RBF [4] and gloable one as in MLP. As a result, MLQP combines the flexibility of local neural network approaches and the generlization property of global neural network approaches.

3. Learning Algorithm

A backpropagation learning algorithm for MLQP is derived in a similar way to that obtaining the backpropagation algorithm for the conventional MLP. This learning algorithm is comprised of the following sequence of operations:

- Calculate the error terms by

$$\delta_{Mk} = (d_{Mk} - x_{Mk}) f'_{Mk}(net_{Mk}) \qquad (3)$$
$$k = 1, 2, \cdots, N_M$$

for the output units, and by

$$\delta_{kj} = f'_{kj}(net_{kj}) \sum_{i=1}^{N_{k+1}} \delta_{k+1,i}(2u_{k+1,ij}\, x_{kj} + v_{k+1,ij}) \qquad (4)$$
$$k = 2, 3, \cdots, M-1, \quad j = 1, 2, \cdots, N_k$$

for the hidden units, where d_{Mk} is the desired output.

- Update the weights by

$$\Delta u_{kji}(t+1) = \alpha_1 \Delta u_{kji}(t) + \eta_1 \delta_{kj} x_{k-1,i}^2 \qquad (5)$$
$$\Delta v_{kji}(t+1) = \alpha_2 \Delta v_{kji}(t) + \eta_2 \delta_{kj} x_{k-1,i} \qquad (6)$$
$$k = 2, 3, \cdots, M; \quad j = 1, 2, \cdots, N_k$$
$$\text{and} \quad i = 1, 2, \cdots, N_{k-1}$$

where α_1 and α_2 are momentums, and η_1 and η_2 are learning rates. Results of various simulations show that setting different values to η_1 and η_2 improves speed of convergence.

4. Experimental Comparison

In order to investigate the computational power of MLQP and to compare it with the conventional MLP and other two kinds of SFNs on pattern classification and function approximation problems, simulations are carried out for three benchmark problems. In the simulations, modified backpropagation learning algorithms [5] are used for training the following four kinds of the networks.

(a) Conventional MLP.

(b) MLQP.

(c) Network with quadratic sigmoidal activation function(NQA) [2]. NQA is a network having the connectivity same as the conventional MLP, and the activation function is defined by:

$$f(net_{kj}) = 1/(1 + \exp(net_{kj}^2 - \theta_{kj})) \qquad (7)$$

where θ_{kj} is an adjustable coefficient. In NQA, higher-order terms are introduced with few extra adjustable coefficients relatively to the conventional MLP, and it is more restrictive in comparison with MLQP. The characteristic of NQA is its partially localized representation, i.e., its representation is local in the direction of the connection weight and global in its orthogonal space.

(d) Network with the second-order unit(NSU) [1]. The characteristic of the unit in NSU is represented by

$$x_{kj} = f\Big(\sum_{i=1}^{N_{k-1}} (u_{kji} x_{k-1,i}^2 + v_{kji} x_{k-1,i}) +$$
$$\sum_{i=1}^{N_{k-1}-1} \sum_{h=i+1}^{N_{k-1}} w_{kjih} x_{k-1,i} \cdot x_{k-1,h} + bias_{kj}\Big) \qquad (8)$$
$$k = 2, 3, \cdots, M; \quad j = 1, 2, \cdots, N_k$$

where u_{kji}, v_{kji}, w_{kjih} and $bias_{kj}$ denote the connection weights and the bias, respectively. For NSU, a unit in the layer k needs

$$(N_{k-1} + 1)(N_{k-1} + 2)/2 \qquad (9)$$

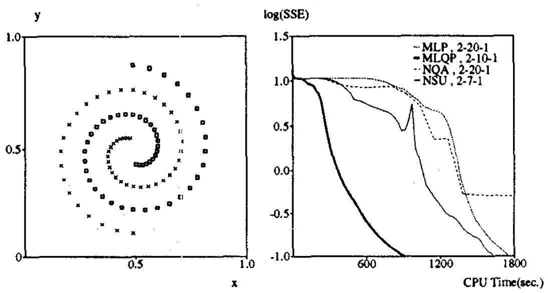
adjustable weights(including bias). Thus, the number of weights required to accommodate all the second correlations increase extremely with the input dimension and the numbers of hidden and output units, and hence it is not suitable to large problems.

In the simulations, all the networks have three layers. In order to compare the performances of the networks systematically, two strategies for choosing the number of hidden units are used. One is to select different number of hidden units to make each of the networks have similar number of adjustable weights, and another is to select same number of hidden units for each of the networks. The numbers of the input, the hidden and the output units are shown in the captions of Fig. 2(b), 4(b) and 5(b). The learning rates are shown in Table 1. They are optimized in convergence speed in the specified problems and the network architectures in preliminary experiments. The momentums are set all to 0.9. In the simulations, the learning is considered completed when the sum of squared error(SSE) between the desired and the actual outputs gets less than a specified value.

4.1 Pattern Classification

For simplicity of illustration, a simplified version of spirals problem(SP) [6] is considered. The training data for this problem are shown in Fig. 2(a).

The points at which the network should output 0's and 1's are represented by small crosses and squares, respectively. The learning curves are illustrated in Fig. 2(b). From Fig. 2(b), we see that MLQP is far superior to MLP, NQA and NSU in learning speed. In order to illustrate the internal representation of MLQP, the decision regions of units in the hidden layer formed by MLQP are depicted in Fig. (3).



(a) (b)

Figure 2: The training input data of the spirals problem(a) and the learning curves(b).

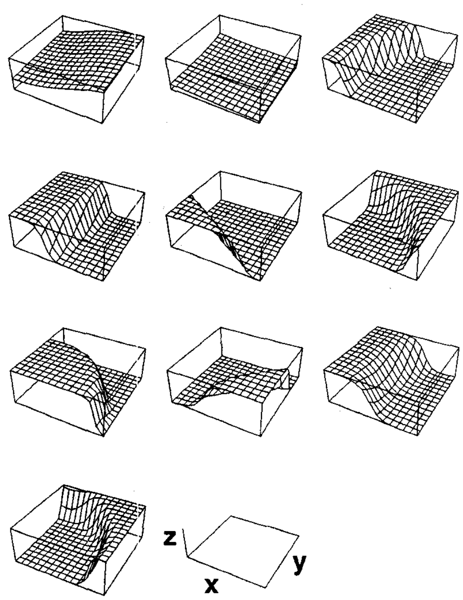


Figure 3: The decision regions of units in the hidden layer formed by MLQP for the problem SP, where x, y and z denote the inputs and the output of the hidden unit, respectively.

4.2 Function Approximation

The following two functions [2] are approximated.

- Stair case function(SF):

$$f_1(x) = \sum_{\omega \in S} \frac{1}{1 + e^{-50(x+\omega)}} \tag{10}$$

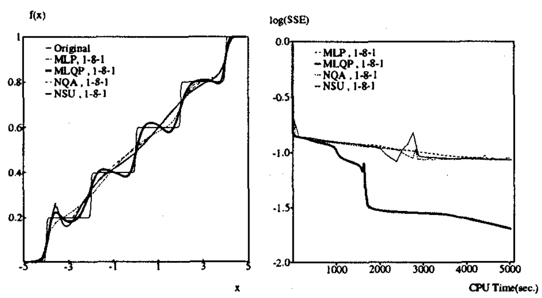
where $S = \{-4, -2, 0, 2, 4\}$ and $x \in D = [-5, 5]$.

- Ripple function(RF):

$$f_2(x, y) = \frac{1 + \sin(6(x^2 + y^2))}{2} \tag{11}$$

where $(x, y) \in D = [0, 1]^2$.

For learning of the stair case and the ripple functions, 100 and 200 training patterns are sampled uniformly from their domains, respectively. The results of approximating the stair case function and the learning curves are shown in Figs. 4(a) and 4(b), respectively. For the ripple function, the original function(RF) and the learning curves are shown in Figs. 5(a) and 5(b), respectively. The approximated functions by MLP, MLQP, NQA, and NSU are shown in Figs. 6(a) through 6(d). The errors between the original ripple function and the approximated functions are shown in Figs. 7(a) through 7(d). From these figures, it is clear that MLQP converges much faster and approximates much better than MLP, NQA and NSU.



(a) (b)

Figure 4: The results of approximating the stair case function(a) and the learning curves(b).

5. Conclusion

This paper has presented a novel multilayer quadratic feedforward neural network, and shown that the proposed network is efficient for pattern classification and function approximation in comparison with the conventional multilayer perceptron and the existing other two kinds of the second-order feedforward networks.

In future studies, we will analyze the computational power of MLQP theoretically, investigate the internal representation deeper and examine its performance on realistic problems such as recognition of hand written characters.

Table 1 The learning rates for pattern classification and function approximation problems

	MLP	NQA	NSU			MLQP	
	η	η	η_1	η_2	η_3	η_1	η_2
SP	0.1	0.018	0.01	0.05	0.05	0.1	0.2
SF	0.22	0.015	0.23	0.3	0.2	0.27	0.47
RF	0.02	0.008	0.026	0.04	0.02	0.05	0.08

Note: η_1, η_2 and η_3 are learning rates for u_{kj}, v_{kj} and w_{kjih} of the NSU, respectively.

References

[1] C. L. Giles and T. Maxwell, "Learning, invariance, and generalization in high-order neural networks", *Applied Optics*, Vol. 26, No. 3, pp.4972-4978, 1987.

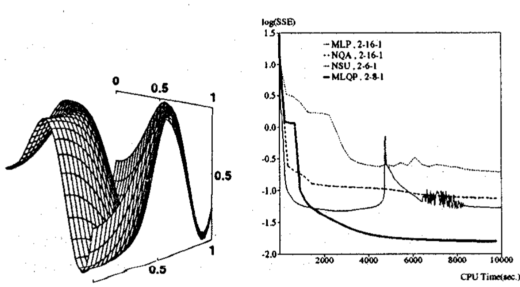
[2] C. C. Chiang and H. C. Fu, "A variant of second-order multilayer perceptron and its application to function approximations", *Proc. of International Joint Conference on Neural Networks*, Baltimore, pp. II 887- II 892, 1992.

[3] F. Piazza, A. Uncini and M. Zenobi, "Artificial neural networks with adaptive polynomial activation function", *Proc. of International Joint Conference on Neural Networks.* Beijing, pp. II 343- II 348, 1992.

[4] J. Moody and C. J. Darken, "Fast learning in networks of locally tuned processing units", *Neural Computation*, vol. 1, pp. 281-294, 1989.

[5] A. V. Ooyen and B. Nienhuis, "Improving the convergence of the backpropagation algorithm", *Neural Netorks*, vol. 5, pp. 465-471, 1992.

[6] K. Lang and M. Witbrock, "Learning to tell two spirals apart", *Proc. of 1988 Connectionist Models Summer School*, Morgan Kanfann, pp. 52-59, 1988.



(a) (b)

Figure 5: The original ripple function(a) and the learning curves(b).

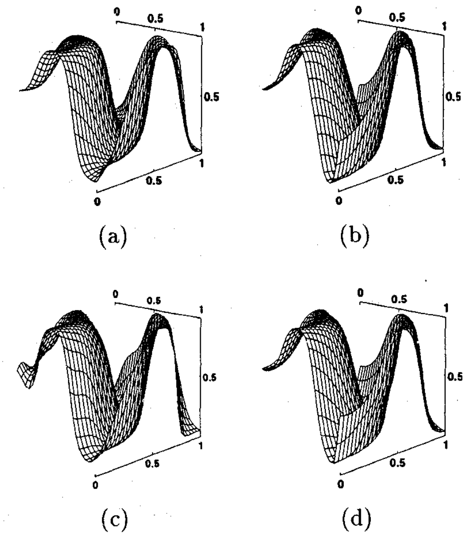


Figure 6: The approximated functions by MLP(a), MLQP(b), NQA(c), and NSU(d).

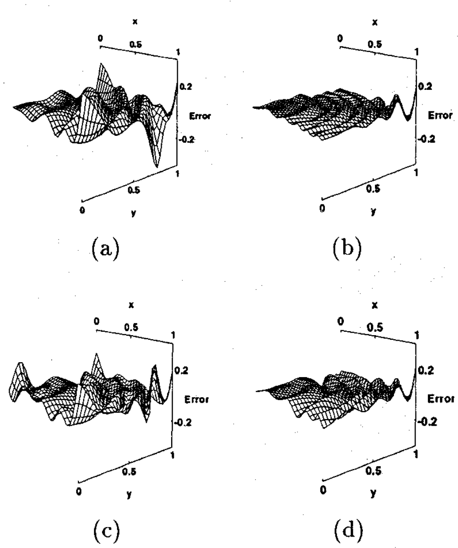


Figure 7: The errors between the original ripple function as shown in Fig. 5(a) and the approximated functions as shown in Fig. 6(a) through 6(d).