Minimum Risk Neural Networks and Weight Decay Technique

I-Cheng Yeh[1], Pei-Yen Tseng[2], Kuan-Chieh Huang[3], and Yau-Hwang Kuo[4]

[1] Department of Civil Engineering, Tamkang University, Taiwan
[2] Department of Information Management, Chung Hua University, Taiwan
[3] Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan
[4] Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan
140910@mail.tku.edu.tw, lytetseng@gmail.com,
m9104041@chu.edu.tw, kuoyh@ismp.csie.ncku.edu.tw

Abstract. To enhance the generalization of neural network model, we proposed a novel neural network, Minimum Risk Neural Networks (MRNN), whose principle is the combination of minimizing the sum of squares of error and maximizing the classification margin, based on the principle of structural risk minimization. Therefore, the objective function of MRNN is the combination of the sum of squared error and the sum of squares of the slopes of the classification function. Besides, we derived a more sophisticated formula similar to the traditional weight decay technique from the MRNN, establishing a more rigorous theoretical basis for the technique. This study employed several real application examples to test the MRNN. The results led to the following conclusions. (1) As long as the penalty coefficient was in the appropriate range, MRNN performed better than pure MLP. (2) MRNN may perform better in difficult classification problems than MLP using weight decay technique.

Keywords: multi-layer perceptrons, weight decay, support vector machine, structural risk minimization.

1 Introduction

This study attempts to enhance the generalization of neural network model. Its basic principle is the combination of minimizing the sum of squares of errors and maximizing the classification margin based on principle of structural risk minimization. In this paper, the Minimum Risk Neural Networks (MRNN) is proposed. The objective function of MRNN is the combination of the sum of squares of errors and the sum of squares of the slopes (differential) of the classification function. This paper will prove that a more sophisticated formula similar to the traditional weight decay technique can be derived from the objective function of MRNN, resulting in a more rigorous theoretical basis for the technique. In Section 2, we will introduce the theoretical background of MRNN. Then we will derive the learning rules of MRNN in Section 3.

D.-S. Huang et al. (Eds.): ICIC 2012, CCIS 304, pp. 10–16, 2012.
© Springer-Verlag Berlin Heidelberg 2012

We will demonstrate the performance of MRNN with several real application examples in the UCI databases in Section 4. Finally, in Section 5 we will make a summary of the testing results in the entire study.

2 Theoretical Background

2.1 Multi-layer Perceptrons with Weight Decay (MLPWD)

Although minimizing error function enables the neural network to build precise non-linear model fitting to the training examples, that is, the model possesses repetition. However, this model may not have the capacity to predict the testing samples, that is, the model may not possess generalization. This phenomenon is called over-learning. In order to overcome the phenomenon, some researchers have suggested weight decay technique, that is, the sum of squares of weights is added to the error function [1-5],

$$E = E_1 + E_2 = \frac{1}{2}\sum_j (T_j - Y_j)^2 + \frac{\lambda}{2}\sum_{k=1}^{N_W} W_k^2 \tag{1}$$

where W_k = the k-th weight in network; N_w = number of weights in network; λ = penalty coefficient of the sum of squares of weights, controlling the degree of weight decay, and its value is greater than or equal to 0.

2.2 Support Vector Machine (SVM) and Structural Risk Minimization

Support Vector Machine (SVM) is a new learning method proposed by Vapnik based on the statistical theory of Vapnik Chervonenks Dimension and Structural Risk Minimization Inductive Principle, and can better solve the practical problems like small amount of samples, high dimension, non-linear and local optimums. It has become one of the hottest topics in the study of machine learning, and is successfully used in classification, function approximation and time series prediction, etc. [6-10]. In SVM, the following objective function is used [6,7],

$$E = E_1 + E_2 = C\sum_{i=1}^{k} \xi_i + \frac{1}{p(w,b)} \tag{2}$$

where C = penalty coefficient, and $C \geq 0$. The greater C is, the greater the penalty of the classification error. ξ_i= the slack variable, and $\xi_i \geq 0$, on behalf of the degree of the classification error of i-th sample. $p(w,b)$ = margin of classification.

In the objective function of Eq. (2), the first item is to minimize the classification error to enable the model with the repetition; the second item is to maximize the classification margin of the hyper-plane to improve the generalization of the model. Compared with Eq. (1), the first item is equivalent to the sum of squares of errors and the second item is equivalent to the sum of squares of weights.

2.3 Minimum Risk Neural Network (MRNN)

Inspired by multi-layer perceptrons with weight decay (MLPWD) and SVM, in this paper, we proposed the Minimum Risk Neural Networks (MRNN). The objective function of MRNN is the combination of the sum of squares of errors and the sum of squares of the slopes (differential) of the classification function.

$$E = E_1 + \gamma \cdot E_2 = \frac{1}{2}\sum_j (T_j - Y_j)^2 + \frac{\gamma}{2}\sum_i \sum_j \left(\frac{\partial Y_j}{\partial X_i}\right)^2 \tag{3}$$

where T_j is the target value of the j-th output variable of the training examples; Y_j is the inference value of the j-th output unit in the output layers for the training examples; γ is the penalty coefficient controlling the proportion of the sum of square of the slopes in the objective function, and its value is greater than or equal to 0.

Comparisons of the objective functions of Multi-layer perceptrons with weight decay technique (MLPWD), support vector machine (SVM), and the minimum risk neural network (MRNN) are shown in Table 1.

Table 1. Comparisons of three kinds of objective function

Model	Error Term E_1	Generalization Term E_2	Principle of Regularization
MLP with weight decay (MLPWD)	$\frac{1}{2}\sum_j (T_j - Y_j)^2$	$\frac{\lambda}{2}\sum_{k=1}^{N_W} W_k^2$	Minimize the sum of square of weights
Support Vector Machine (SVM)	$C\sum_{i=1}^{k} \xi_i$	$\frac{1}{p(w,b)}$	Minimize the reciprocal of the classification margin
Minimum Risk Neural Network (MRNN)	$\frac{1}{2}\sum_j (T_j - Y_j)^2$	$\frac{\gamma}{2}\sum_i \sum_j \left(\frac{\partial Y_j}{\partial X_i}\right)^2$	Minimize the sum of squares of slopes of classification function

3 Theoretical Derivation

3.1 Minimum Risk Neural Network (MRNN)

The output of the hidden unit in MLP is as follows

$$H_k = f(net_k) = \frac{1}{1+\exp(-net_k)} = \frac{1}{1+\exp(-(\sum_i W_{ik} X_i - \theta_k))} \tag{4}$$

where H_k is the output of the k-th unit in the hidden layer; X_i is the i-th input variable; W_{ik} is the connection weight between the i-th unit in the input layer and the k-th unit in the hidden layer; θ_k is the threshold of the k-th unit in the hidden layer.

The output of the output unit in MLP is as follows

$$Y_j = f(net_j) = \frac{1}{1+\exp(-net_j)} = \frac{1}{1+\exp(-(\sum_k W_{kj} H_k - \theta_j))} \tag{5}$$

where W_{kj} is the connection weight between the k-th unit in the hidden layer and the j-th unit in the output layer; θ_j is the threshold of the j-th unit in the output layer.

In order to achieve the minimum of the objective function of MRNN in Eq. (3), we can use the steepest descent method to adjust the network parameters. The learning rules are derived in two steps as following.

(1) Connection weights between the hidden layer and the output layer

According to the chain rule in the partial differential, and let

$$\delta_j \equiv (T_j - Y_j) \cdot f'(net_j) \tag{6}$$

Then, we get

$$\Delta W_{kj} = \eta \cdot \left(\delta_j H_k - \gamma \cdot \sum_i \left(\sum_l f'(net_j) \cdot W_{lj} \cdot f'(net_l) \cdot W_{il} \right) \left(f'(net_j) \cdot f'(net_k) \cdot W_{ik} \right) \right) \tag{7}$$

(2) Connection weights between the input layer and the hidden layer

According to the chain rule in the partial differential, and let

$$\delta_k \equiv \left(\sum_j \delta_j W_{kj} \right) \cdot f'(net_k) \tag{8}$$

Then, we get

$$\Delta W_{ik} = \eta \cdot \left(\delta_k X_i - \gamma \cdot \sum_j \left(\sum_l f'(net_j) \cdot W_{lj} \cdot f'(net_l) \cdot W_{il} \right) \left(f'(net_j) \cdot W_{kj} \cdot f'(net_k) \right) \right) \tag{9}$$

3.2 The Relation between MRNN and Weight Decay Technique

In this section, we will simplify the above formula to derive formulas of weight decay technique. In Eq. (7), the first order partial derivatives of the transfer functions must be positive. Hence, they can be omitted so as to simplify the formula. Therefore,

$$\Delta W_{kj} = \eta \cdot \left(\delta_j H_k - \gamma \cdot \sum_i \left(\sum_l W_{il} \cdot W_{lj} \right) \cdot W_{ik} \right) \tag{10}$$

Similarly, Eq. (9) can be simplified as

$$\Delta W_{ik} = \eta \cdot \left(\delta_k X_i - \gamma \cdot \sum_j \left(\sum_l W_{il} \cdot W_{lj} \right) \cdot W_{kj} \right) \tag{11}$$

Comparing Eq. (10) and (11) with Eq. (1) of weigh decay technique, we can find that both of them imply the rule that "the modification of weigh is in reverse proportion to weight". Hence, conventional weight decay technique can be considered as the simplified version of MRNN. These formulas establish reasonable theoretical foundation for weight decay technique.

4 Application Examples

In this section, we tested three real data sets in the UCI Machine Learning Repository [11] compare the performance of MRNN, MLPDW and MLP, including (1) detection of spam mail (2) recognition of remote sensing image of Landsat satellite (3) classification of forest cover type. To evaluate the effectiveness of learning, we used the 10-fold cross-validation. We tried γ= 0.0001, 0.001, 0.01, 0.03, 0.1, 0.3, 1, 3, 5, and 10 for MRNN, and $\lambda=10^{-7}\sim10^{-1}$ for MLPWD. The results are shown in Figure 1. It can be found that as long as the parameter γ or λ is in the appropriate range, both of them perform better than pure MLP. We also experimented on other 12 practical data sets in UCI databases [11] listed in Table 2 to compare the performances of MRNN, MLPWD, and SVM. To evaluate the effectiveness of learning, we used the 10-fold cross-validation. In addition, to avoid the influence of the initial connection weights, the error rates are the average of the results of 30 sets of various initial connection weights. To evaluate whether the performance differences between the three kinds of

Table 2. Testing results of error rate of the 15 UCI data sets

UCI data sets	Benchmark			MRNN		MRNN vs. Benchmark t-test (Significance=5%)	
	MLPWD		SVM				
	Avg.	Std.		Avg.	Std.	MLPWD	SVM
SPAMBASE	0.0642	0.0027	0.0653	0.0631	0.0018	0.037 *	<0.001 *
Landsat	0.0981	0.0016	0.098	0.0974	0.0013	0.036 *	0.008 *
Forest cover	0.232	0.002	0.215	0.208	0.003	<0.001 *	<0.001 *
Iris	0.0270	0	0.027	0.0270	0	>0.5	>0.5
Insurance	0.3366	0.0131	0.3365	0.3363	0.0160	0.468	0.473
Glass	0.2675	0.0036	0.2665	0.2667	0.0047	0.248	>0.5
Shuttle	0.0049	0.0001	0.004	0.0040	0.0001	<0.001 *	>0.5
Vowel	0.4123	0.0091	0.4052	0.3983	0.0096	<0.001 *	<0.001 *
Wine	0.0116	0.0002	0.0115	0.0113	0.0002	<0.001 *	<0.001 *
Letter	0.3474	0.0071	0.3418	0.3315	0.0073	<0.001 *	<0.001 *
Image	0.0422	0.0008	0.0422	0.0421	0.0009	0.330	0.281
Vehicle	0.1240	0.0020	0.1232	0.1230	0.0015	0.025 *	0.294
German	0.2393	0.0071	0.2365	0.2362	0.0052	0.034 *	0.397
Heart	0.1430	0.0019	0.143	0.1430	0.0019	>0.5	>0.5
Thyroid	0.0241	0.0002	0.0231	0.0198	0.0002	<0.001 *	<0.001 *

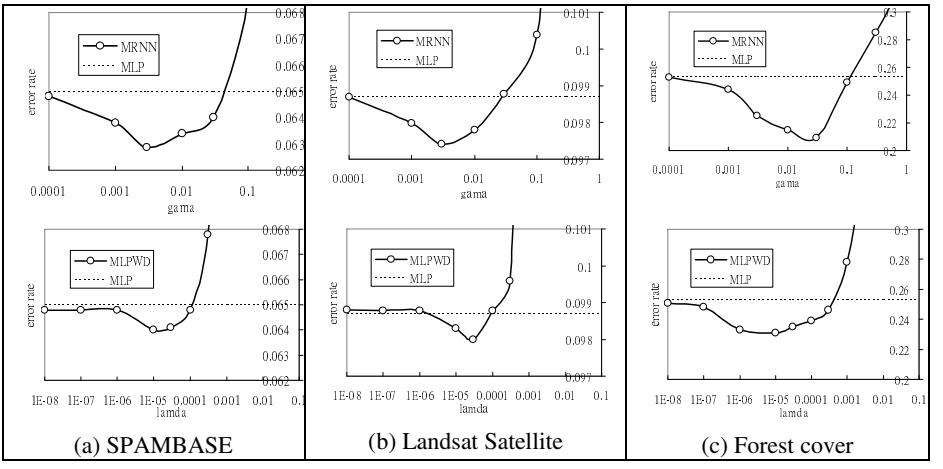

Fig. 1. The error rate of MRNN and MLPWD

neural networks are significant or not, the t-test was employed. From the experimental results listed in Table 2, we can see that there are 10 out of fifteen data sets whose error of MRNN is significantly smaller than that of MLPWD; however, there are only seven out of fifteen data sets whose error of MRNN is significantly smaller than that of SVM.

5 Conclusions

The generalization capability of a multilayer perceptron can be adjusted by adding a penalty (weight decay) term to the cost function used in the training process. To enhance the generalization of neural network model, inspired by SVM, we proposed the Minimum Risk Neural Networks, whose objective function is the combination of the sum of squares of errors and the sum of squares of the slopes of the classification function. Besides, this paper proved that a more sophisticated formula similar to the traditional weight decay technique can be derived from the MRNN, establishing a more rigorous theoretical foundation for the technique. This study employed fifteen real examples to test the MRNN. The results led to the following conclusions. (1) As long as the penalty coefficient was in the appropriate range, MRNN performed better than pure MLP. (2) MRNN may perform better in difficult classification problems than the MLP using weight decay technique.

Aknowledgements. This work was supported by the National Science Council, ROC, under Grant NSC-100-2221-E-032-070.

References

1. Wu, L.Z., Moody, J.: A Smoothing Regularizer for Feedforward and Recurrent Neural Networks. Neural Computation 8(3), 461–489 (1996)
2. Krogh, A., Hertz, J.A.: A Simple Weight Decay Can Improve Generalization. In: Moody, J.E., Hanson, S.J., Lippmann, R.P. (eds.) Advances in Neural Information Processing Systems, San Mateo, CA, pp. 450–957 (1992)
3. Krogh, A., Hertz, J.A.: A Simple Weight Decay Can Improve Generalization. In: Advances in Neural Information Processing Systems, vol. 4, pp. 950–957 (1992)
4. Hinton, G.E., Camp, D.: Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights. In: Proceedings of the Sixth Annual Conference on Computational Learning Theory, pp. 5–13 (1993)
5. Treadgold, N.K., Gedeon, T.D.: Simulated Annealing and Weight Decay in Adaptive Learning: the SARPROP algorithm. IEEE Transactions on Neural Networks 9(4), 662–668 (1998)
6. Cortes, F., Vapnik, V.: Support Vector Networks. Machine Learning 20(3), 273–297 (1995)
7. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
8. Drucker, H., Wu, D., Vapink, V.: Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks 10(5), 1048–1054 (1999)
9. Burges, C.: A Tutorial on Support Vector Machines for Pattern Recognitionl. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
10. Fan, R.E., Chen, P.H., Lin, C.J.: Working Set Selection using Second Order Information for Training Support Vector Machines. The Journal of Machine Learning Research 6, 1889–1918 (2005)
11. UCI Machine Learning Repository Content Summary (2008), `http://archive.ics.uci.edu/ml/`