J. Infrastructure Plan. and Man., JSCE, No.618/IV-43, 95-104, 1999 April

GENERALIZATION TECHNIQUES FOR LAYERED NEURAL NETWORKS IN THE CLASSIFICATION OF REMOTELY SENSED IMAGES

Eihan SHIMIZU [1], Morito TSUTSUMI [2] and Le Van TRUNG [3]

[1]Member of JSCE, Dr. of Eng., Professor, Dept. of Civil Eng., University of Tokyo
(Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan)
[2]Member of JSCE, M. of Eng., Research Associate, Dept. of Civil Eng., University of Tokyo
(Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan)
[3]Non-member of JSCE, Ph. D., Dept. of Civil Eng., The National University of Hochiminh City
(268 Ly Thuong Kiet Street, District 10th HoChiMinh City, Vietnam)

In recent years, researchers have paid a lot of attention to Layered Neural Networks (LNNs) as a non-parametric approach for the classification of remotely sensed images. This paper focuses on the generalization capability of LNNs, that is, how well an LNN performs with unknown data. First, we clarify its description from the point of view of information statistics. With this discussion, we provide a feasible technique to design the LNN in consideration of its generalization capability. Finally, we apply the proposed technique to a practical land cover classification using remotely sensed images, and demonstrate its potential.

Key Words : *layered neural network, generalization, Akaike's Information Criterion, Tikhonov's regularization*

1. INTRODUCTION

Among supervised classification methods for remotely sensed data, Maximum Likelihood Classification (MLC) is presently the most widely known and utilized (e.g. Curran and Hay [1], Yool *et al.*[2]). MLC is often used as a standard classification routine against which other classification algorithms are compared. This popularity is due to the fact that MLC is the optimal classifier in the sense of minimizing Bayesian error. However, MLC is a parametric classification method where the underlying probability density function must be assumed a priori. We may obtain a poor MLC performance if the true probability density function is different from what is assumed in the model. In recent years, researchers have attempted to provide non-parametric classification methods to overcome this disadvantage of MLC, and Layered Neural Networks (LNNs) have been proposed as suitable for the efficient classification of remotely sensed images.

When using an LNN classifier, however, users have often been faced with a generalization problem; generalization is concerned with how well an LNN model performs with input on which the model has not been previously trained. That is, an LNN classifier usually performs well on a set of training data, but it may not guarantee good generalization over all unknown data during the actual classification process.

This paper discusses LNN classifier generalization, a controversial and often vague term in the neural network literature [3], and clarifies its description.

We introduce some techniques for generalization based on Akaike's Information Criterion and provide a feasible technique to design an LNN in consideration of its generalization capability. Finally, we apply the proposed technique to a practical land cover classification using remotely sensed images, and demonstrate its potential.

2. BASIC FORMULATION OF THE LAYERED NEURAL NETWORK CLAS-SIFIER

It has been proved that a three-layered neural network, when the appropriate number of nodes are set in the hidden layer and the sigmoidal activation function is used in the hidden and output nodes, can approximate any continuous mapping (Gallant and White [4], Funahashi [5], Cybenko [6], and Hornik *et al.*[7]). Therefore, in this study, we only focus on three-layered neural networks, as shown in **Figure 1**.

Layered feed-forward neural networks have been broadly applied to prediction, classification, pattern recognition and other modeling problems. Hill *et al* [8] gave an almost perfect review of studies comparing LNNs with conventional statistical models.

Let $x = \{x_i\}\,(i = 1,2,\cdots,I)$ represent the feature vector which is to be classified. Let the possible classes be denoted by $\omega_j\,(j = 1,2,\cdots,J)$. Consider the discriminant function $d_j(x)$, and then the decision rule is

$$x \in \omega_j, \text{ if } d_j(x) \geq d_{j^*}(x) \quad \text{for all } j^* \neq j. \tag{1}$$

An LNN is expected to be the input-output (I/O) system corresponding to the discriminant function.

The multi-layered neural network being applied to a variety of classification problems has an input layer, an output layer and several hidden layers. The neural network to be trained can be viewed as a parameterized mapping from a known input to the output which should be as close as possible to the training data.

A feature vector is an input to the input layer; that is, the number of neurons in the input layer corresponds to the dimension of the feature vectors. The number of neurons in the hidden layer can be adjusted by the user. The output layer has the same number of neurons as the classes.

Let the state of the h th neuron be represented by

$$\begin{aligned} u_h &= g(x,w) \\ &= \sum_{i=1}^{I} x_i w_{ih}, \end{aligned} \tag{2}$$

where w_{ih} is a parameter functioning as a synaptic weight between neurons included in the designed LNN. These parameters are mainly constituted by the connection weights (synaptic weights) between neurons. The output signal from the j th neuron in the output layer is regarded as the discriminant value. The output of LNN, under presentation of x, is

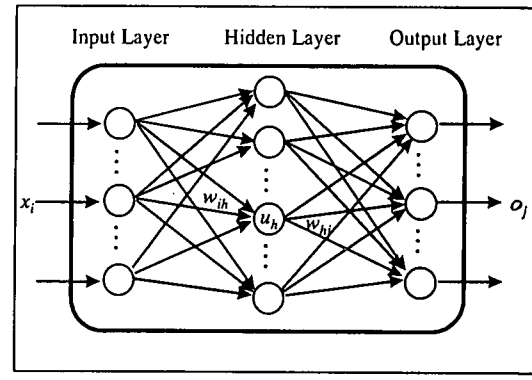

Fig.1 Architecture of three-layered neural network.

$$o_j(x,w) = f(u_j), \tag{3}$$

where $f(\cdot)$ is an activation function. The following sigmoid function, which is bounded, monotonic and non-decreasing, is frequently used,

$$f(u_j) = \frac{1}{1+\exp(-u_j)}. \tag{4}$$

The feature vectors x_k $(k = 1,2,\cdots,K)$ for training the LNN are prepared. The classes to which these feature vectors belong are all known. Training data (target data) are given as follows:

$$d_j(x_k) = \begin{cases} 1 & \text{if } x_k \in \omega_j \\ 0 & \text{otherwise}. \end{cases} \tag{5}$$

Training of the LNN is performed through the adjustment of connection weights. The most commonly used method is so-called *back propagation*, which is a gradient descent method in essence.

An error function can be defined as the sum of the squares of the errors for the overall training set:

$$E = \sum_{k=1}^{K} E_k, \tag{6}$$

where

$$E_k = \frac{1}{2}\sum_{j=1}^{J}\{o_j(x_k,w) - d_j(x_k)\}^2. \tag{7}$$

The LNN is trained by minimizing the value of the error function; that is,

$$\min_{w} \quad E. \tag{8}$$

Among several methods for training the LNN, gradient descent methods are most commonly used. There are two approaches for their application to feed-forward neural networks [9]. One is based on modifying the weights according to the rule,

$$w_{ih}^{new} = w_{ih}^{old} + \eta \cdot \Delta w_{ih} , \quad (9)$$

$$w_{hj}^{new} = w_{hj}^{old} + \eta \cdot \Delta w_{hj} , \quad (10)$$

$$\Delta w_{ih} = -\frac{\partial E}{\partial w_{ih}} , \quad (11)$$

$$\Delta w_{hj} = -\frac{\partial E}{\partial w_{hj}} , \quad (12)$$

where $\eta > 0$ is the step-size parameter. The other is based on modifying the weights according to the rule (9), (10) and

$$\Delta w_{ih} = -\frac{\partial E_k}{\partial w_{ih}} , \quad (13)$$

$$\Delta w_{hj} = -\frac{\partial E_k}{\partial w_{hj}} . \quad (14)$$

The data are repeatedly presented in either approach, until the processes converge, although there is no guarantee of convergence to the solution.

Following precedents, we call the former approach periodic updating and the latter continuos updating [9]. And an entire pass through all the data set is called an *epoch*.

Through chain differentiation

$$\frac{\partial E_k}{\partial w_{ih}} = \frac{\partial E_k}{\partial o_j} \cdot \frac{\partial o_j}{\partial u_j} \cdot \frac{\partial u_j}{\partial f} \cdot \frac{\partial f}{\partial u_h} \cdot \frac{\partial u_h}{\partial w_{ih}}$$
$$= (o_j - d_j) \cdot f'(u_j) \cdot w_{hj} \cdot f(u_h) \cdot x_h , \quad (15)$$

$$\frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial o_j} \cdot \frac{\partial o_j}{\partial u_j} \cdot \frac{\partial u_j}{\partial w_{hj}}$$
$$= (o_j - d_j) \cdot f'(u_j) \cdot x_h . \quad (16)$$

Back propagation has the advantage that the derivatives on the right sides of (15) and (16) are calculated by the inputs and the outputs of the neurons, for the sigmoid function has a simple derivative as follows

$$f'(u) = f(u) \cdot \{1 - f(u)\} . \quad (17)$$

3. GENERALIZATION OF THE LAYERED NEURAL NETWORK

When using an LNN classifier, however, users have often been faced with a generalization problem. In this chapter, we discuss LNN classifier generalization, which has been a controversial and often vague term in the neural network literature[3], and provide a clear description from the viewpoint of architecture design and learning paradigm. Then, we provide a clear description of the generalization of an LNN classifier based on information statistics by introducing an information criterion.

(1) Generalization and network architecture design

The architecture of an LNN is basically defined as the number of layers, the number of nodes in each layer and the form of the activation function, so that the selected model will identify the relationship between input and output, and will predict correctly with new input data.

The problem of choosing the optimal number of nodes and layers is analogous to choosing an optimal subset of regressor variables in a regression model [10]. We know that if a model $y = \varphi(x)$ with too many free parameters is used to fit a given set of training data (x, y) then it may "over-fit" the training data, while with too few parameters it may not be powerful enough to describe the relationship between input x and output y. In other words, if the number of parameters is too large, the calibrated function may pass through all the specified training data (x_i, y_i) without error. However, the function could be highly oscillatory, leading to large errors at unknown data that are not included in the training data set. This phenomenon is to be explained in detail in section (5).

(2) Generalization and learning paradigm

Once the architecture is fixed, the behavior of the trained model depends on the values of connection weights obtained from the training paradigm and the limited number of training data.

Given the architecture of an LNN, it is possible that repeated training iterations successively improve the performance of the LNN on training data by "memorizing" training patterns. However, due to the limited number of training data and the presence of noise, *over-training* usually presents problems. Over-training is the phenomenon whereby after a certain number of training epochs, more training epochs will further reduce the learning error (only slightly in many cases) on the training data set but

will produce greater errors on a new data set which are not included in the training data set.

(3) Akaike's Information Criterion

In order to discuss the generalization of an LNN classifier based on information statistics, we introduce an information criterion.

Let us assume that $p(x)$ is a probability density function and $p(x,w)$ is a model distribution. The discrepancy between the model and the real distribution can be measured in terms of Kullback-Leibler's information distance:

$$D\{p(x),p(x,w)\} \equiv \int p(x)\ln\frac{p(x)}{p(x,w)}dx \quad .(18)$$

If we assume that the real distribution $p(x)$ belongs to a model set and the number of observations is sufficiently high, the information distance in (18) can be expressed as a function of the distance between the real and the Maximum Likelihood Estimate (MLE) of the parameters:

$$D(p(x,w),p(x,w_O)) = D(w,w_O) \approx \frac{1}{2}(w-w_O)^t M(w-w_O) \quad , \quad (19)$$

where M is Fisher's information matrix and t denotes a transpose of a matrix, w is the true parameter vector and w_O is the parameter vector for MLE.

Under the assumption that the competing models belong to a sequence of hierarchy where the lower dimensional models are included into the higher dimensional ones as sub-models, Akaike has extended Maximum Likelihood Estimation (MLE) in such a way that *An Information Criterion (Akaike's Information Criterion*: AIC) can be used to simultaneously address both the parameter estimation and optimal model selection [11]. Akaike defined AIC as an estimator of the expected information distance by using approximation (19),

$$E[2K\cdot D\{p(x,w),p(x,w_O)\}] \approx \mathrm{AIC} \equiv 2\left\{-\sum_{k=1}^{K}\ln p(x,w_O)+l\right\} \quad , \quad (20)$$

where $E[\cdot]$ denotes the expectation operator, K is the number of data and l is the number of independent parameters.

(4) Application of AIC to LNN

We should notice that neural networks and traditional statistical classifiers are not to be related in normal cases. However, when the output of the LNN has been trained with a sufficient number of training data, it is considered as an approximated estimate of a Baysian posterior probability (Wan[3] and Ruck *et al.*[12]). It makes a big difference that we have sufficient number of training data in the case of the classification of remotely sensed images, which provide a theoretical interpretation of the output from the LNN classifiers as an estimate of a posterior probability. Thus, AIC is applicable to LNNs as a criterion for generalization and is determined by

$$\mathrm{AIC} = 2\left(-\sum_{k=1}^{K}\sum_{j=1}^{J}\ln\{p_j(x_k,w)\}+L\right) \quad , \quad (21)$$

where p_j is output of the node j in the output layer (corresponding to class j) and L is the number of independently adjusted parameters.

The model that minimizes AIC is the best model. If only one model set is used, that is, the number of parameters is fixed, then AIC will result in the MLE solution. If two different model sets have the same value of the maximum likelihood, the model with the smaller number of parameters will be selected (*principle of parsimony* or *Occam's razor*).

(5) Generalization based on AIC

In case of LNNs, the number of parameters in (21) is determined by the number of parameters in the activation function and the number of connection weights. The number of parameters is basically determined by the numbers of input nodes, output nodes, hidden layers and hidden nodes. Here, the trade-off between the number of parameters and the overall goodness-of-fit of the model is also inevitable.

Although the expectation of AIC is asymptotically unbiased up to the terms of order $O(1)$, it has a large variance, so that the procedure discussed in the previous sections are also problematic[13].

To give an example of over-training, let y be a linear function of x,

$$y_k = x_k w + \varepsilon_k \qquad (k=1,\cdots,K) \quad , \quad (22)$$

where ε is the error term, and w is the coefficient vector.

Let the regression model be expressed by matrices

$$y = Xw + \varepsilon \quad . \quad (23)$$

Suppose $\varepsilon \sim N(0,\sigma^2\mathbf{I})$, where σ is an unknown

standard error. ML estimator

$$w_O = \left(X^t X\right)^{-1} X^t y \qquad (24)$$

is the solution to

$$\max_{w,\sigma^2} \log L(w,\sigma^2) = -\frac{K}{2}\log 2\pi - \frac{K}{2}\log\sigma^2 - \frac{1}{2\sigma^2}(y - Xw)^t (y - Xw) \quad . \quad (25)$$

The situation where the determinant of X^tX is nearly zero is called multicollinearity where the problem (25) is ill-posed, that is, unstable. It is regarded as an over-fitting.

The generalization of LNN classifiers is considered to be a problem of the search for an appropriate architecture and appropriate training algorithm so that the LNN performs well and minimizes the errors over all unknown data. In the following chapters, we propose an LNN design with some techniques for improving the generalization of LNN classifiers; these are characterized by the architecture and training algorithm.

4. LAYERED NEURAL NETWORK ARCHITECTURE DESIGN BASED ON AIC

In this chapter, we propose an LNN architecture design for choosing an appropriate model in terms of not only the size of network but also a suitable activation function based on AIC.

(1) Choosing the appropriate size based on AIC

In the three-layered neural network, the number of parameters L is determined by the number of hidden nodes H, the number of input nodes I and the number of output nodes J as follows

$$L = I \times H + H \times J \quad . \qquad (26)$$

The numbers of nodes in the input and output layers are, in general, fixed according to the practical application problem. The users, therefore, are only able to adjust the number of nodes in the hidden layer.

a) Choosing the number of hidden nodes

If we begin to select a hidden layer with too few nodes, the LNN may not be powerful enough for a given learning task, while too many hidden nodes would lead to over-fitting the training data. Therefore, an appropriate number of hidden nodes should be chosen so that the LNN can guarantee the generalization ability.

The relationship between the number of hidden nodes H and the number of output nodes J has been fairly well discussed by Mehrotra *et al.*[14], Weigend and Rumelhart[15], and Amirikian and Nishimura[16]. They conclude that an LNN with one hidden layer and H hidden nodes the number of which equals the number of output nodes J, is considered an appropriate size to execute a given classification task.

We suggest choosing an appropriate size of LNN by using an AIC that can be used to simultaneously address both the parameter estimation and forecasting the generalization ability of the model on unknown data during the training process. We can choose the appropriate number of hidden nodes by changing the number of hidden nodes to get different sizes of LNN. Their AIC values are determined after the completion of training. The LNN yielding the minimum value of AIC will be chosen as being of the appropriate size to execute a given classification task.

b) Pruning the connection weights

Network pruning algorithms can be applied to get the minimum number of parameters (reducing the redundant parameters) so that the LNN is more efficient in both forward computation time and generalization capability. These algorithms have already been discussed by several researchers such as Sietsma and Dow[17], Karnin[18] and Hassibi and Stork [19].

While the simplest procedure for network pruning is to heuristically find and prune those connection weights whose removal does not change output values, we suggest carrying out these procedures based on AIC. That is, we prune weights whose values are relatively small and re-train the network. Then, we calculate the AIC of each network and choose the appropriate combination of connection weights based on the minimization of AIC.

(2) Choosing the appropriate activation function

The architecture of an LNN classifier with one hidden layer is composed of the number of nodes in the hidden layer and the type of activation function. Therefore, the optimal architecture of the LNN should be chosen from the viewpoint of not only the size of network but also the activation function form suitable for the training data set at hand.

From the theoretical interpretation of the LNN by Shimizu [20], we can get various types of probability distribution by changing the value of parameter $a(\geq -1)$ in the generalized activation function

$$f(u_j) = \frac{1}{-a + \exp(-\beta u_j)}, \qquad (27)$$

to get the best fit to the training data. We estimate β conjoined with w_{ih} and w_{hj}. Hence, without essential loss of generality, let β be 1. For the use of the generalized activation function, the weight update rule of the ordinary back-propagation algorithm should be modified with its derivative

$$f'(u) = f(u) \cdot \{1 + a \cdot f(u)\}, \qquad (28)$$

in place of (14).

We apply the modified back-propagation algorithm and change parameter a of the generalized activation function in the training process to get the best fit to the training data based on AIC. After the appropriate size of the LNN has been fixed, the value of AIC is determined by the maximum log-likelihood alone.

$$-\mathrm{MLL} = -\sum_{k=1}^{K}\sum_{j=1}^{J} \ln\{p_j(x_k, w)\}. \qquad (29)$$

The advantage of this approach is to provide information of the best fit of the activation function to the training data at hand comparing competing activation function forms.

(3) Practical procedure

The procedure described in this chapter is only an example and there may be many alternatives. In addition, the procedure described here should be carried out with feedback or simultaneity searching for an optimal architecture. However, it is almost impossible or computationally too costly for us to make a search of all possible subsets of the models. Thus, from the viewpoint of practical application, it is to a certain degree reasonable to start by choosing the number of nodes in the hidden layer and to prune the connection weights before tuning the activation function. Then, we choose the model of LNNs which minimizes AIC.

5. LEARNING PARADIGM OF LAYERED NEURAL NETWORK

It is known that AIC has a large variance, so that the proposed procedures are problematic. Given the architecture of an LNN, it is possible that repeated training iterations will successively improve the performance of the LNN on training data, but due to the limited number of training data and the presence of noise, over-training often occurs.

In order to improve the over-training problem, we introduce Tikhonov's regularization.

(1) Tikhonov regularization

Tikhonov's regularization is a method that optimizes a function $E + \gamma \cdot z(w)$, where E is the original error function, $z(\cdot)$ is a smoothing function, and γ is a regularization parameter[21]. Although there are obviously many possibilities for choosing the forms of the smoothing functions, the Euclidean norm of the parameters is most commonly used. Thus, in this paper, we adopt it as a smoothing function.

In case of regression model (22), we can obtain the modified maximum log-likelihood as follows:

$$\max_{w,\sigma^2} \log\Lambda(w,\sigma^2) = -\frac{K}{2}\log 2\pi - \frac{K}{2}\log\sigma^2 - \frac{1}{2\sigma^2}(y - Xw)^t(y - Xw) - \gamma \cdot w^t w, \qquad (30)$$

yielding

$$w_\gamma = \left(X^t X + \gamma I\right)^{-1} X^t y. \qquad (31)$$

Solution (31) is called a Ridge estimator and was proposed to Least Squares Method by Hoerl and Kennard [22), 23)]. Mean Squared Error (MSE) of the parameters is defined as:

$$\mathrm{MSE} = E\left[(w_O - w)^t (w_O - w)\right]. \qquad (32)$$

In case of regression model (20), MSE is

$$\mathrm{MSE}(w_O) = \sigma^2 tr\left(X^t X\right)^{-1}. \qquad (33)$$

By introducing Tikhonov's regularization, MSE is modified as

$$\mathrm{MSE}(w_\gamma) = E\left[(w_\gamma - w)^t (w_\gamma - w)\right] = \sigma^2 tr\left\{\left(X^t X + \gamma I\right)^{-1} X^t X \left(X^t X + \gamma I\right)^{-1}\right\}. \qquad (34)$$

Hoerl and Kennard [22),23)] show that there always exists a constant $\gamma > 0$ such that

$$\mathrm{MSE}(w_\gamma) < \mathrm{MSE}(w_O). \qquad (35)$$

However, it should be noted that the parameter γ which minimizes MSE depends on unknown parameter w.

(2) Application of Tikhonov's regularization to training algorithm

By using Tikhonov's regularization method, the generalization problem of an LNN can be solved by optimizing function $E + \gamma \cdot z(w)$ analogously.

We adopt the square of the Euclidean norm of weights w_{ih}, w_{kj} as a smoothing function. Then, we can obtain the modified error function as follows

$$E(\gamma) \equiv \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{J}\left\{o_j(x_k, w) - d_j(x_k)\right\}^2 + \gamma \cdot \left(\sum_{h=1}^{H}\sum_{i=1}^{I} w_{ih}^2 + \sum_{j=1}^{J}\sum_{h=1}^{H} w_{hi}^2\right) \quad . \quad (36)$$

The rule of changing the value of weights in the back-propagation algorithm is also modified as [16), 24)]

$$w_{ih}^{new} = w_{ih}^{old} + \eta \cdot \Delta w_{ih} \quad , \quad (37)$$

$$w_{hj}^{new} = w_{hj}^{old} + \eta \cdot \Delta w_{hj} \quad , \quad (38)$$

$$\Delta w_{ih} = -\frac{\partial E(\gamma)}{\partial w_{ih}} = -\frac{\partial E}{\partial w_{ih}} - \gamma \cdot w_{ih} \quad , \quad (39)$$

$$\Delta w_{hj} = -\frac{\partial E(\gamma)}{\partial w_{hj}} = -\frac{\partial E}{\partial w_{hj}} - \gamma \cdot w_{hj} \quad . \quad (40)$$

or (37), (38) and

$$\Delta w_{ih} = -\frac{\partial E_k(\gamma)}{\partial w_{ih}} = -\frac{\partial E_k}{\partial w_{ih}} - \gamma \cdot w_{ih} \quad , \quad (41)$$

$$\Delta w_{hj} = -\frac{\partial E_k(\gamma)}{\partial w_{hj}} = -\frac{\partial E_k}{\partial w_{hj}} - \gamma \cdot w_{hj} \quad , \quad (42)$$

where

$$E_k(\gamma) \equiv \frac{1}{2}\sum_{j=1}^{J}\left\{o_j(x_k, w) - d_j(x_k)\right\}^2 + \gamma \cdot \left(\sum_{h=1}^{H}\sum_{i=1}^{I} w_{ih}^2 + \sum_{j=1}^{J}\sum_{h=1}^{H} w_{hi}^2\right) \quad . \quad (43)$$

A modified back-propagation algorithm based on Tikhonov's regularization for improving the generalization ability of an LNN is called weight decay in some literature [25)].

Needless to say, an ordinary back-propagation algorithm is considered to be a special type of algorithm when the penalty parameter γ is zero. If we adopt an optimal regularization parameter γ, we obtain a better weight so that the expected residual sums of squares might be less. However, as was explained in the previous section, selecting the best regularization parameter depends on information on the value of the true parameter. Thus, there have been some methods proposed to select appropriate regularization parameters. One of the most popular methods among them is to determine the value of a regularization parameter by testing how it performs on the validation data set [26)].

6. CASE STUDY

In this chapter, we will design an LNN based on the suggested procedure and apply it to a practical land cover classification problem.

(1) Study area and data used

The study area is located in Nagakute town, Aichi prefecture in Japan. Airborne Multi-Spectral Scanner digital image data of 256×256 pixels were acquired with 12 bands. A pixel size is 6.25 by 6.25 m.

We classified the area into seven classes based on the land cover by different model sets and tested the possibilities of practical application of the suggested techniques for generalization problem.

(2) Experimental results

Figure 2 shows the LNN based remotely sensed image classification system with 7 nodes in the input layer and 7 nodes in the output layer. The number of hidden layer nodes was chosen based on AIC.

Each pixel consists of 12 spectral measurements to be assigned to one of seven classes. Training data of 1500 pixels and test data of 400 pixels were extracted from digital ground truth data with the same pixel size (6.25 by 6.25 m) for all models.

Starting with a model where the numbers of hidden nodes and output nodes are both 7, eight competing size sets of LNN were compared. In each competing size sets, pruning the connection weights was done based on AIC. The results are shown in **Figure 3.** This indicates that AIC is minimized at the point where the number of hidden nodes is 4 and the difference of AIC between an LNN with 4 hidden nodes and others is more than 30. Consequently, an LNN with 4 hidden nodes was chosen.

Once the size of the nodes in the hidden layer in the LNN was fixed at 4, competing architecture sets of different forms of activation function were applied. Let us stress again that the procedure described in this chapter is only an example and there may be many alternatives. Moreover it is desirable that the procedure is carried out with feedback or simultaneity searching for the optimal architecture. However, as it is computationally too costly, we did not search all possible subsets of the models. The

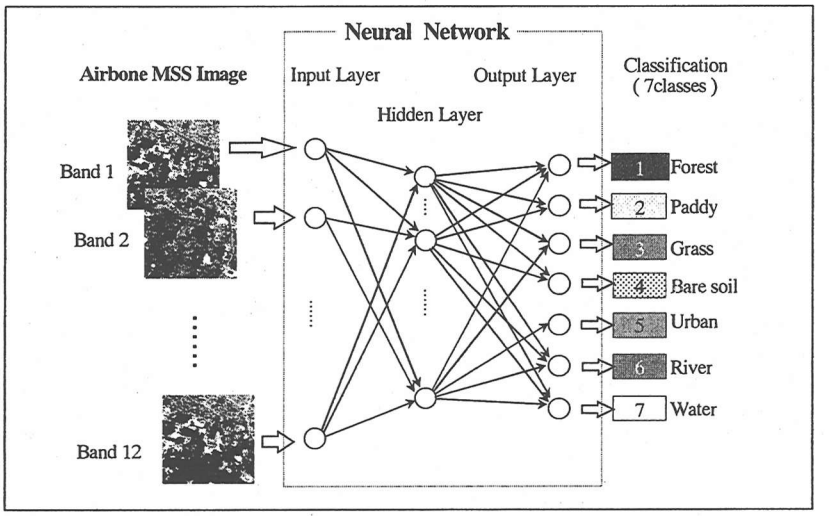

Fig.2 The LNN for remotely sensed images

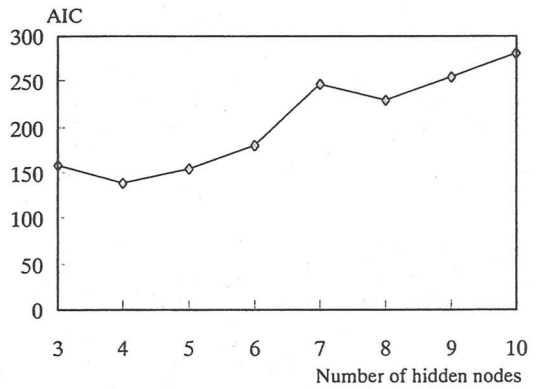

Fig. 3 AIC for competing size sets of LNN.

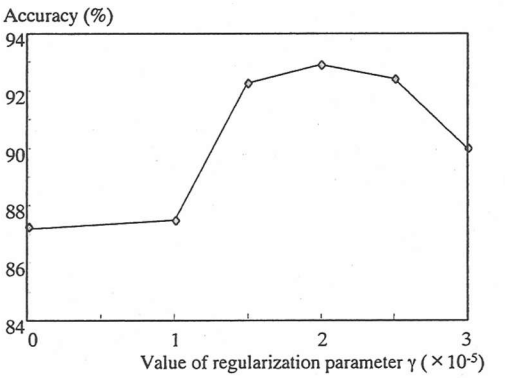

Fig. 5 The regularization penalty parameter γ and the accuracy for validation data.

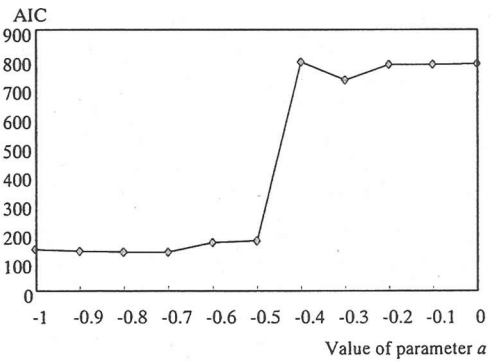

Fig. 4 AIC and the form of activation function.

minimized AIC was 131 for the architecture which adopted the parameter a =-0.8, and 138 for the one which used the normal sigmoid function (a =-1.0) (**Fig. 4**).

After choosing the appropriate architecture, the best parameter set of the model could be estimated; this minimizes the modified error function for validation data with the penalty parameter $\gamma \fallingdotseq 2.0\times10^{-5}$. **Figure 5** shows how introduction of the penalty term contributes to the generalization of the LNN and that the improvement of generalization by Tikhonov's regularization was notable.

The classification results of the selected appropriate models at each step were compared with those of the base model, which has seven nodes in the hidden layer trained on the standard back-propagation algorithm. **Table 1** shows the comparison among the results of the competing models. The left column shows the results of Model (a) in which the number of hidden nodes and output nodes are equally 7, and pruning has not been done. The middle column shows the results of Model (b) which achieved the minimization of AIC by reducing

Table 1 The comparison among the results.

		(a)	(b)	(c)
Number of Input Nodes		12		
Number of Hidden Nodes		7	4	
Parameter	a	-1.0	-0.8	
	γ	0		2.0×10^{-5}
AIC		246	131	(225)*
Accuracy(%)		85.4	87.2	92.7

* The parentheses shows that 225 is not the value of AIC properly but the value of corresponding objective function.

the number of hidden nodes and pruning some connection weights. And the right column shows the results of Model (c) obtained by applying Tikhnov's regularization to Model (b).

It is shown that reducing the number of hidden nodes and pruning the connection weights are effective for the decrease of the number of parameters so that the minimization of AIC can be achieved. The improvement of generalization by Tikhonov's regularization is also notable.

7. CONCLUSION

In this paper, we introduced techniques for the generalization of Layered Neural Networks (LNNs) and proposed LNN design in the classification of remotely sensed images.

We discussed the generalization of LNN classifiers, a controversial and often vague term in the neural network literature, and introduced some techniques based on information statistics. Akaike's Information Criterion (AIC) was introduced for LNNs taking into consideration the fact that the output of the LNN, which has been trained with a sufficient number of training data, is considered as an approximated estimate of a Baysian posterior probability. Then, we gave a clear description of LNN generalization from the viewpoint of architecture design and learning paradigm based on AIC.

Concerning the architecture design, the size of network (the number of layers and nodes) and the type of activation functions are important factors. We proposed LNN architecture design based on the minimization of AIC. Concretely, different sized sets of LNN were trained with pruning, and the number of hidden nodes and the connections weights between nodes were determined based on the minimization of AIC.

Once the architecture is fixed, the behavior of the trained model depends on the values of the connection weights. It is known, however, that AIC has a large variance, so that due to the limited number of training data and the presence of noise, over-training often presents problems. We introduced Tikhonov's regularization to overcome the problem of over-training.

Finally, we designed an LNN classifier based on the proposed procedure and applied it to a land cover classification problem. Our experimental results illustrate the potential of the proposed design techniques. We believe that the insight gained from this study is complementary to a more general analysis for the generalization of feed-forward layered neural networks based on information statistics.

ACKNOWLEDGMENT: The authors would like to thank the anonymous referees for their valuable comments.

REFERENCES

1) Curran, P.J., and Hay, A.M.: The importance of measurement error for certain procedures in remote sensing at optical wavelengths, *Photogramm. Eng. Remote Sensing*, Vol.52, pp.229-241, 1986.

2) Yool, S.R., Star, J.L., Estes, J.E., Botkin, D.B, Eckhardt, D.W. and Davis, F.W.: Performance analysis of image processing algorithms for classification of natural vegetation in the mountains of Southern California, *Int. J. Remote Sensing*, Vol.7, pp.683-702, 1986.

3) Wan, E. A.: Neural network classification: a Bayesian interpretation. *IEEE Trans. Neural Networks*, Vol.1, No.4, pp.303-305, 1990.

4) Gallant, A.R. and White, H.: There exists a neural network that does not make avoidable mistakes, *Proc. Int. Conf. Neural Networks*, Vol.1, pp.657-666, 1988.

5) Funahashi, K.: On the approximate realization of continuous mapping by neural networks, *Neural Networks*, Vol.2, pp.183-192, 1989.

6) Cybenko, G.: Approximation by superpositions of a sigmoidal function, *Mat. of Contr., Signals and System*, Vol.2, pp.304-314, 1989.

7) Hornik, K., Stinchcombe, M., and White, H.: Multilayer feedforward networks are universal approximators, *Neural Networks*, Vol.2, No.5, pp.359-366, 1989.

8) Hill, T., Marquez, L., O'Connor, M. and Remus, W.: Artificial neural network models for forecasting and decision making, *Int. Jour. Forecasting*, Vol.10, pp.5-15, 1994.

9) Bose, N. K. and Liang, P.: *Neural Network Fundamentals with Graphs, Algorithms, and Applications*, NcGraw-Hill, 1996.

10) Forgel, D. B.: An information criterion for optimal neural network selection, *IEEE Trans. Neural Networks*, Vol.2, No.5, pp.490-497, 1991.

11) Akaike, H.: A new look at the statistical model identification, *IEEE Trans. Automat. Contr.*, Vol. 19, No.6, pp.716-723, 1974.

12) Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M. E. and Suter, B. W.: The multilayer perceptron as an approximation to a Bayes optimal discriminant function, *IEEE Trans. Neural Networks*, Vol.1, No.4, pp.296-298, 1990.
13) Shimohira, H.: A Model selection procedure based on the information criterion with its variance, *METR*, 93-16, University of Tokyo, 1993.
14) Mehrotra, K.G., Mohan, C.K. and Ranka, S.: Bounds on the number of samples needed for neural learning, *IEEE Trans. Neural Networks*, Vol. 6, pp.548-558, 1991.
15) Weigend, A.S. and Rumelhart, D. E.: The effective dimension of the space of hidden units. *In Proc. IEEE Int. Joint Conf. Neural Network, Singapore*, Vol. 3, pp. 2069-2074, 1991.
16) Amirikian, B. and Nishimura, H.: What size network is good for generalization of a specific task of interest, *Neural Networks*, Vol. 7, No.2, pp.321-329, 1995.
17) Sietsma, J. and Dow, R. J.F.: Creating artificial neural networks that generalize, *IEEE Trans. Neural Networks*, Vol. 4, pp.67-79, 1990.
18) Karnin, E. D.: A Simple Procedure for pruning back-propagation trained neural networks, *IEEE Trans. Neural Networks*, Vol. 2, pp.239-242, 1990.
19) Hassibi, B. and Stork, D.G. : Second order derivatives for network pruning: Optimal brain surgeon, *Advances in Neural Information Processing Systems*, Vol. 5, pp.164-171, 1993.
20) Shimizu, E.: A theoretical interpretation for layered neural network classifier, *Jour. JSPRS*, Vol.35, No.4, pp.4-8, 1996.
21) Tikhonov, A.N., Goncharsky, A.V., Stepanov, V.V, and Yagola, A.G.: *Numerical Methods for the Solution of Ill-posed Problems. Mathematics and Its Applications*, Kluwer Academic Publishers, 1990.
22) Hoerl, A. E. and Kennard, R. W.:Ridge regression : Biased estimation for nonorthogonal problems, *Technometrics*, Vol. 12, No. 1, pp. 55-67, 1970.
23) Hoerl, A. E. and Kennard, R. W.:Ridge regression : Applications to nonorthogonal problems, *Technometrics*, Vol. 12, No. 1, pp. 69-82, 1970.
24) Mehrotra, K., Mohan, C. K. and Ranka, S.: *Elements of Artificial Neural Networks*, MIT Press, 1997.
25) Krogh, A. and Hertz, A. J.: A simple weight decay can iprove generalization, *Advances in Nueral Information Processing Systems 4*, Moody, J. E., Hanson, S. J. and Lippmann, R. P. eds., Morgan Kaufman Publishers, 1992.
26) Golub, G. H., Heath, M. and Wahba, G.: Generalized cross-validation as a method for choosing a good paramter, *Technometrics*, Vol.21, pp.215-223, 1979.

(Received June 2, 1998)

リモートセンシング画像分類のための階層型ニューラルネットワークの汎化手法

清水英範・堤盛人・Le Van TRUNG

リモートセンシングデータの分類問題へのノンパラメトリックな接近法の一つとして，近年，階層型ニューラルネットワーク（LNN）を用いた分類モデルが注目され，従来の最尤法による手法等と比較しても高い推定能力を有していることが確認されている．しかし，この高い推定能力は，LNNの関数としての複雑さとその学習方法に支えられているのであり，安易な適用はモデルとしての汎化性（未知データに対する推定能力）を著しく損ねることが指摘されている．本研究ではLNN分類モデルの汎化性問題について，LNN分類モデルの出力がベイズの事後確率を近似することを利用し，情報量統計学の視点から整理を行った．次に，情報量基準の一つである赤池の情報量基準を用いた汎化性向上の設計手法を示し，実際の土地被覆分類問題に適用してその有効性について検討した．