Learning of Three-Layered Neural Networks by Enlarging Domains of Recognition

Niijima, Koichi
Department of Informatics, Kyushu University

Ohkubo, Akito
Fukuoka Institute of Health and Environmental Sciences

Mohamed, Marghny H.
Department of Informatics, Kyushu University

https://hdl.handle.net/2324/3030

出版情報：DOI Technical Report. 172, 2000-02-07. Department of Informatics, Kyushu University
バージョン：
権利関係：

DOI-TR-172

DOI Technical Report

Learning of Three-Layered Neural Networks by Enlarging Domains of Recognition

by

Koichi Niijima, Akito Ohkubo, Marghny H. Mohamed

February 7, 2000

Department of Informatics
Kyushu University
Fukuoka 812-8581, Japan

Email: niijima@i.kyushu-u.ac.jp Phone: +81-92-583-7631

Learning of Three-Layered Neural Networks by Enlarging Domains of Recognition

Koichi Niijima*, Akito Ohkubo†, and Marghny H. Mohamed‡

February 7, 2000

Abstract

The concept of domains of recognition is introduced for three-layered neural networks. The domain lies in the input space and can be represented using connection weights and thresholds of the network. We propose a learning method of the network so as to enlarge the domain of recognition by extending its range mapped into the hidden space and by minimizing the slope of affine transforms in the mapping. Based on the method, we introduce a cost function whose minimization process gives a learning algorithm of the network. A land cover classification problem has been considered in our simulation.

1 Introduction

Many learning methods for three-layered neural networks have been studied from the viewpoints of classification and recognition abilities. The back propagation learning is one of the most popular learning techniques and widely recognized as a powerful tool for learning the input-output mapping. The outputs of the network that was learnt by this method are close to supervised signals for training patterns. However, this method does not guarantee what kind of unknown patterns can be classified in the same category. Several neural network theories and methods for adaptive learning and for dynamic modification of neural network structures have been introduced so far: incremental learning [1]; growing neural networks [2, 6]; pruning neural networks [3, 4, 5]. A problem of these networks is that their theoretical properties are not related directly to the concrete design of the general pattern. In fact, these properties say that the networks have excellent approximation properties, but do not give us any hint about how to predict the category for the unknown pattern.

*K. Niijima is with the Department of Informatics, Kyushu University, Kasuga 816-8580, JAPAN. E-mail: niijima@i.kyushu-u.ac.jp

†A. Ohkubo is with Fukuoka Institute of Health and Environmental Sciences, Dazaifu 818-0135, JAPAN. E-mail: cgg29501@pcvan.or.jp

‡M.H. Mohamed is a doctoral student of Department of Informatics, Kyushu University, Kasuga 816-8580, JAPAN. E-mail: mohamed@i.kyushu-u.ac.jp

Recently, one of the authors derived cone-like domains of attraction, each of which contains a memorized pattern, in autoassociative memory networks in a real-valued vector space [7]. We proposed there a learning method of the network that enlarges the domain of attraction. Any pattern in the domain can be classified in the same category as the memorized pattern in the domain. The enlargement of the domain was done by using the fact that its boundaries consist of some hyperplanes expressed by the weights and thresholds in the network. Such a simple shape of the domain comes from a single layer perceptron. In neural networks with hidden units, the domain of attraction has a complicated form due to nonlinear sigmoid functions in the network.

In this paper, we derive domains in the input space each of which includes several memorized patterns and represents a category in the case of classification. We call the domain a domain of recognition from now on. In this case, pattern recognition can be defined as the categorization of input data into identifiable domains via the extraction of significant features or attributes of the data from a background of irrelevant detail. The boundaries of such a domain contain a nonlinear sigmoid function as well as the weights and thresholds in the network. This makes difficult to learn the network so as to enlarge the domain of recognition. An attention has been paid to the range of the domain mapped into the hidden space. This range is a con-like domain surrounded by a hypercube having zero as a center in the hidden space. However, the enlargement of the range is not sufficient to extend the domain of recognition in the input space. To make large the domain of recognition, we need an additional condition. It is to minimize the slope of affine transforms in the network mapping from the input space into the hidden space. Such two requirements are formulated as a minimization problem of some cost function. The cost function includes supervised conditions as a penalty term. The minimization of the cost function is carried out by applying gradient methods such as the steepest descent method and conjugate gradient method. These minimization processes give our learning algorithm for three-layered neural networks.

The plan of this paper is as follows: In Section 2, domains of recognition for three-layered neural networks are introduced in the same way as in the cone-like domains in a single layer perceptron [7]. The range of the domain of recognition into the hidden space takes the form of cone-like domain, and is surrounded by a hypercube having zero as a center. In Section 3, we derive a condition to make large such a range. The domain of recognition can be enlarged by considering an additional condition for minimizing the slope of affine transforms included in the network between the input and hidden layers. By unifying the above two conditions, we make a functional to be minimized under the supervised conditions. We finally derive a cost function containing these supervised conditions as a penalty term, and minimize the function by various gradient methods. This minimization process gives our learning algorithm. Section 4 is devoted to a computer simulation. We apply our learning method to a land cover classification problem in remote sensing. Section 5 is a conclusion.

2 Three-Layered Neural Network

Let n be the number of input nodes, h the number of hidden units, and l the number of output units. We consider a three-layered neural network:

$$y_i = g(\sum_{j=1}^{h} w_{ij} f(\sum_{k=1}^{n} v_{jk} x_k - \eta_j) - \theta_i), \quad i = 1, 2, ..., l, \tag{1}$$

where v_{jk} and w_{ij} denote weights connecting the k-th input node and the j-th hidden unit, and connecting the j-th hidden unit and the i-th output unit, respectively. The η_j and θ_i indicate thresholds at the j-th hidden unit and at the i-th output unit, respectively. The functions $f(t)$ and $g(t)$ represent sigmoid functions:

$$f(t) = \frac{1 - \exp(-t)}{1 + \exp(-t)}, \quad g(t) = \frac{1}{1 + \exp(-t)}.$$

We introduce the vectors $W_i = {}^t(w_{i1}, w_{i2}, ..., w_{ih})$ and $\psi(x) = {}^t(\psi_1(x), \psi_2(x), ..., \psi_h(x))$ in which $\psi_j(x) = f(V_j \cdot x - \eta_j)$, where $V_j = {}^t(v_{j1}, v_{j2}, ..., v_{jn})$ and $x = {}^t(x_1, x_2, ..., x_n)$ with the transpose symbol ^{t}. Then, (1) is expressed using the inner product symbol $\cdot$ as

$$\begin{aligned} y_i &= g(\, W_i \cdot \psi(x) - \theta_i) \\ &\equiv \varphi_i(x), \qquad\qquad i = 1, 2, ..., l. \end{aligned} \tag{2}$$

Putting $\varphi(x) = {}^t(\varphi_1(x), \varphi_2(x), ..., \varphi_l(x))$ and $y = {}^t(y_1, y_2, ..., y_l)$, we rewrite (2) as

$$y = \varphi(x). \tag{3}$$

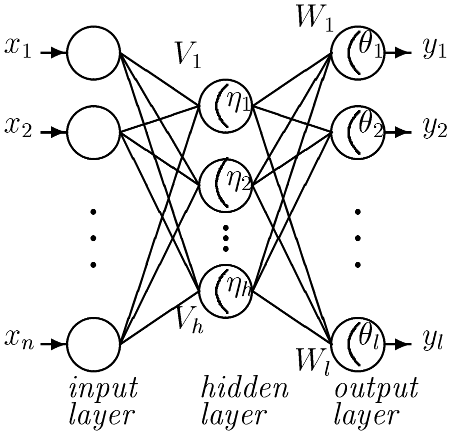

Figure 1: Three-layered neural network

3 Domains of Recognition

We assume that the number of categories to be separated is l. We have m_τ training data for the τ-th category, where $\tau = 1, 2, \cdots, l$. Let us denote the whole training data by x^ν, $\nu = 1, 2, \ldots, m$ with $m = \sum_{\tau=1}^{l} m_\tau$. We introduce the set

$$J_k = \{\nu \mid \sum_{j=0}^{k-1} m_j < \nu \leq \sum_{j=0}^{k} m_j\}, \quad m_0 = 0.$$

For latter convenience, we define the index function $q(\nu)$ by

$$q(\nu) = k, \quad \nu \in J_k, \quad k = 1, 2, \cdots, l.$$

On the output value $\varphi_i(x^\nu)$ at the i-th output unit, we impose the following supervised condition

$$\begin{aligned} \varphi_i(x^\nu) &\geq 1 - \varepsilon, & i = q(\nu), \quad (4)\\ \varphi_i(x^\nu) &\leq \varepsilon, & i \neq q(\nu) \quad (5) \end{aligned}$$

with a sufficiently small ε satisfying $0 < \varepsilon < 1/2$. Since the function $s = f(t)$ is monotonically increasing and has the inverse function $t = \ln(s/(1-s))$, we can rewrite (4) and (5) as follows:

$$\begin{aligned} W_i \cdot \psi(x^\nu) - \theta_i &\geq \ln\frac{1-\varepsilon}{\varepsilon}, & i = q(\nu), \quad (6)\\ W_i \cdot \psi(x^\nu) - \theta_i &\leq -\ln\frac{1-\varepsilon}{\varepsilon}, & i \neq q(\nu). \quad (7) \end{aligned}$$

We define a domain $D_\rho(x^\nu)$ by

$$\begin{aligned} D_\rho(x^\nu) = \{x \in R^n \mid\ & W_i{\cdot}(\psi(x) - \psi(x^\nu)) \\ & \leq \rho \mid W_i{\cdot}\psi(x^\nu) - \theta_i \mid,\ i{\neq}q(\nu), \\ & W_i{\cdot}(\psi(x) - \psi(x^\nu)) \\ & \geq -\rho \mid W_i{\cdot}\psi(x^\nu) - \theta_i \mid,\ i = q(\nu)\} \end{aligned}$$

in the input space. Although this domain has a complicated form because its boundaries contain the nonlinear function $f(t)$, we can prove the following result.

Theorem 1: For any x in $D_\rho(x^\nu)$, we have

$$\begin{aligned} \varphi_i(x) &\geq 1 - \varepsilon^{1-\rho}, & i = q(\nu), \quad (8)\\ \varphi_i(x) &\leq \varepsilon^{1-\rho}, & i \neq q(\nu). \quad (9) \end{aligned}$$

Proof: Let x be in $D_\rho(x^\nu)$. First, we consider the case $i = q(\nu)$. Then, since $W_i \cdot \psi(x^\nu) - \theta_i > 0$, we have

$$W_i \cdot (\psi(x) - \psi(x^\nu)) \geq -\rho(W_i \cdot \psi(x^\nu) - \theta_i).$$

Using this inequality and (6), we can derive

$$\begin{aligned} W_i \cdot \psi(x) - \theta_i &= W_i \cdot (\psi(x) - \psi(x^\nu)) \\ &\qquad + W_i \cdot \psi(x^\nu) - \theta_i \\ &\geq (1-\rho) \ln \frac{1-\varepsilon}{\varepsilon}. \end{aligned}$$

A further use of the inequality

$$\left(\frac{1-\varepsilon}{\varepsilon}\right)^{1-\rho} \geq \frac{1-\varepsilon^{1-\rho}}{\varepsilon^{1-\rho}}$$

leads us to

$$W_i \cdot \psi(x) - \theta_i \geq \ln \frac{1-\varepsilon^{1-\rho}}{\varepsilon^{1-\rho}}$$

which implies (8). Similarly, we can prove (9).

This theorem means that any x belonging to $D_\rho(x^\nu)$ can be recognized as the training pattern x^ν. We call $D_\rho(x^\nu)$ a domain of recognition.

As a result of Theorem 1. we can prove

Theorem 2: We define l unions of $D_\rho(x^\nu)$ by

$$S_k = \cup_{\nu \in J_k} D_\rho(x^\nu), \quad k = 1, 2, \cdots, l.$$

Then, S_k are mutually disjoint.

Proof: The proof is done by proof of contradiction. Suppose that $S_k \cap S_{k'} \neq \phi$ for $k \neq k'$, where ϕ denotes an empty set. Then there exists x^* belonging to $D_\rho(x^\nu)$ for $\nu \in J_k$ and $D_\rho(x^\mu)$ for $\mu \in J_{k'}$. Since J_k and $J_{k'}$ are different, there exists i such that $\varphi_i(x^*) \geq 1 - \varepsilon^{1-\rho}$ and $\varphi_i(x^*) \leq \varepsilon^{1-\rho}$. This leads us to contradiction.

Theorem 2 implies that if a different firing condition per each different category is given as a supervised condition, the neural network has classification ability. This fact will be applied to land cover classification problems to be described in Section IV.

4 Learning Method

In the previous section, we defined the domain of recognition $D_\rho(x^\nu)$. This domain is desirable to be as large as possible. So we want to determine the connection weights and the thresholds in the network so as to enlarge the domain $D_\rho(x^\nu)$.

If there is no hidden layer in the network (1), this domain takes the form

$$\begin{aligned} D_\rho(x^\nu) = \{x \in R^n \mid \ & W_i{\cdot}(x - x^\nu) \\ & \leq \rho \mid W_i \cdot x^\nu - \theta_i \mid, \ i{\neq}q(\nu), \\ & W_i \cdot (x - x^\nu) \\ & \geq -\rho \mid W_i \cdot x^\nu - \theta_i \mid i = q(\nu)\} \end{aligned}$$

which has a very simple form because the boundaries of the domain are hyperplanes. Using this fact, we derived in [7] a learning algorithm of a single layer perceptron.

In the neural network having hidden layers, the boundaries of the domain $D_\rho(x^\nu)$ has a complicated shape, which makes difficult a learning of the neural network. So we consider a mapping of the domain $D_\rho(x^\nu)$ into the hidden space R^h. It can be represented as

$$\begin{aligned} E_\rho(\psi(x^\nu)) = \{u \in R^h \mid \ & W_i{\cdot}(u - \psi(x^\nu)) \\ & \leq \rho \mid W_i{\cdot}\psi(x^\nu) - \theta_i \mid, \ i{\neq}q(\nu), \\ & W_i{\cdot}(u - \psi(x^\nu)) \\ & \geq -\rho \mid \ W_i{\cdot}\psi(x^\nu) - \theta_i \mid, \ i = q(\nu)\}. \end{aligned} \tag{10}$$

We see from (10) that the boundaries of $E_\rho(\psi(x^\nu))$ consist of hyperplanes. But it should be noticed that the range $E_\rho(\psi(x^\nu))$ is in the h-dimensional cube $[-1, 1]^h$. For latter use, we define l unions of $E_\rho(\psi(x^\nu))$ by

$$H_k = \cup_{\nu \in J_k} E_\rho(\psi(x^\nu)), \quad k = 1, 2, \cdots, l.$$

We illustrate the relation between the unions S_k and the unions H_k in Fig.2.

Concerning the union H_k, we have the following result.

Corollary 1: At most one H_k includes zero in the hidden space R^h.

Proof: The proof is obvious from Theorem 2.

Corollary 1 will be used to enlarge $E_\rho(\psi(x^\nu))$ in the cube $[-1, 1]^h$.

Although $D_\rho(x^\nu)$ has a complicated form, $E_\rho(\psi(x^\nu))$ is a region whose boundaries consist of hyperplanes. To enlarge $D_\rho(x^\nu)$, we first make large the region $E_\rho(\psi(x^\nu))$. We decompose $E_\rho(\psi(x^\nu))$ into a cone $Cone(\psi(x^\nu))$ and a strip $Str(\psi(x^\nu))$ as

$$E_\rho(\psi(x^\nu)) = Cone(\psi(x^\nu)) \cup Str(\psi(x^\nu)),$$

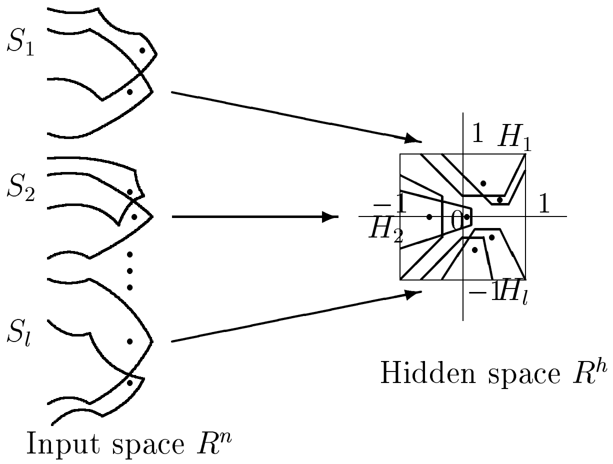

Figure 2: Relation of S_k and H_k

where

$$
\begin{aligned}
Cone(\psi(x^\nu)) = \{u \in R^h \mid & W_i{\cdot}(u - \psi(x^\nu)) \leq 0,\ i \neq q(\nu), \\
& W_i{\cdot}(u - \psi(x^\nu)) \geq 0,\ i = q(\nu)\}
\end{aligned}
$$

and

$$
\begin{aligned}
Str(\psi(x^\nu)) = \{u \in R^h \mid & 0 < W_i{\cdot}(u - \psi(x^\nu)) \\
& \leq \rho \mid W_i{\cdot}\psi(x^\nu) - \theta_i \mid,\ i \neq q(\nu), \\
& 0 > W_i{\cdot}(u - \psi(x^\nu)) \\
& \geq -\rho \mid W_i{\cdot}\psi(x^\nu) - \theta_i \mid,\ i = q(\nu)\}.
\end{aligned}
$$

As was done in [7], we make large the width of $Str(\psi(x^\nu))$ which can be expressed as

$$\frac{\rho|W_i \cdot \psi(x^\nu) - \theta_i|}{||W_i||}.$$

Since this width depends on the number ν of training data, we maximize the squared summation on ν,

$$\frac{\sum_{\nu=1}^{m}(W_i \cdot \psi(x^\nu) - \theta_i)^2}{||W_i||^2},$$

that is, we minimize

$$\frac{||W_i||^2}{\sum_{\nu=1}^{m}(W_i \cdot \psi(x^\nu) - \theta_i)^2}.$$

Concerning the $Cone(\psi(x^\nu))$, we maximize the angle γ where the hyperplanes $W_i{\cdot}(u - \psi(x^\nu)) = 0$ and $W_j \cdot (u - \psi(x^\nu)) = 0$ cross. To do so, it suffices to minimize $\cos\gamma = W_i \cdot W_j / ||W_i||||W_j||$.

It is insufficient to extend only $E_\rho(\psi(x^\nu))$ itself. We must enlarge the region $E_\rho(\psi(x^\nu))$ in the cube $[-1, 1]^h$. From Corollary 1, we see that it is desirable for

$\|\psi(x^\nu)\|$ to be as close to zero as possible. So we minimize

$$\sum_{\nu=1}^{m} \|\psi(x^\nu)\|^2.$$

To make large the domain $D_\rho(x^\nu)$, we also need to consider the mapping from $D_\rho(x^\nu)$ into $E_\rho(\psi(x^\nu))$. The essence of this mapping lies in an affine transform

$$a_j = V_j \cdot x - \eta_j$$

in the function $\psi_j(x)$. Since the sigmoid function $f(s)$ contained in $\psi_j(x)$ is monotonically increasing, the domain $D_\rho(x^\nu)$ is larger if the norm of V_j is smaller. Hence, we minimize

$$\sum_{j=1}^{h} \|V_j\|^2.$$

Finally, we consider the supervised conditions (4) and (5). Let us introduce the following function

$$z_+^2 = z^2, \quad z > 0, \qquad z_+^2 = 0, \quad z \le 0.$$

Then, by bounding the term

$$\sum_{i \ne q(\nu)} \left(\ln \frac{1-\varepsilon}{\varepsilon} + W_i \cdot \psi(x^\nu) - \theta_i\right)_+^2 + \sum_{i = q(\nu)} \left(\ln \frac{1-\varepsilon}{\varepsilon} - W_i \cdot \psi(x^\nu) + \theta_i\right)_+^2$$

from above, we can realize the supervised conditions (4) and (5).

The expression for the cost function F, in terms of the above conditions, is given by

$$\begin{aligned} F = \sum_{i=1}^{l} \frac{\|W_i\|^2}{\sum_{\nu=1}^{m}(W_i \cdot \psi(x^\nu) - \theta_i)^2} + C_1 \sum_{i \ne j} \frac{W_i \cdot W_j}{\|W_i\|\,\|W_j\|} \\ + C_2 \sum_{\nu=1}^{m} \|\psi(x^\nu)\|^2 + C_3 \sum_{j=1}^{h} \|V_j\|^2 \\ + C_4 \left[\sum_{i \ne q(\nu)} \left(\ln \frac{1-\varepsilon}{\varepsilon} + W_i \cdot \psi(x^\nu) - \theta_i\right)_+^2 \right. \\ \left. + \sum_{i = q(\nu)} \left(\ln \frac{1-\varepsilon}{\varepsilon} - W_i \cdot \psi(x^\nu) + \theta_i\right)_+^2 \right], \end{aligned} \tag{11}$$

where C_i denote penalty constants. Our learning algorithm is given as a minimizing process for this cost function. The training of the network takes care of the task of minimization of F with respect to W_i, V_j and θ_i which is performed by gradient-descent technique.

In actual computation, however, we minimize the following functional in place of (11) to avoid numerical instability:

$$
\begin{aligned}
F = \sum_{i=1}^{l} \frac{1}{\sum_{\nu=1}^{m}(U_i \cdot \psi(x^\nu) - \xi_i)^2} + C_1 \sum_{i \neq j} U_i \cdot U_j \\
+ C_2 \sum_{\nu=1}^{m} \sum_{j=1}^{h} (V_j \cdot x^\nu - \eta_j)^2 + C_3 \sum_{j=1}^{h} \|V_j\|^2 \\
+ C_4 \Bigg[\sum_{i \neq q(\nu)} (\alpha + \beta_i(U_i \cdot \psi(x^\nu) - \xi_i))_+^2 \\
+ \sum_{i = q(\nu)} (\alpha - \beta_i(U_i \cdot \psi(x^\nu) - \xi_i))_+^2 \Bigg] \\
+ C_5 \sum_{i=1}^{l} \left(\|U_i\|^2 - 1\right)^2,
\end{aligned}
\tag{12}
$$

where we have put $U_i = W_i/\|W_i\|$, $\xi_i = \theta_i/\|W_i\|$, $\beta_i = \|W_i\|$, and $\alpha = \ln((1-\varepsilon)/\varepsilon)$. We minimize finally the cost function (12) by the gradient method. This minimization process gives our learning algorithm.

5 Land Cover Classification

5.1 Data Description

Data for land cover classification are obtained by the Thematic Mapper (TM) installed in Landsat-5 and the Active Microwave Instrument (AMI) installed in ERS-2. Each of the data is 7 band values which are from visible to infrared reflections observed by TM, or 8 band values consisting of these values and 1 band value of microwave scattering observed by AMI. The TM value of each band has a 8 bits expression. However, since the AMI value I has a 16 bits expression, the value I is converted into a 8 bits value D_n by the following equations:

$$
\begin{aligned}
\sigma_0 &= 20 \times \log_{10}(I) - 68.5 \quad (dB), \\
D_n &= 5(\sigma_0 + 30)
\end{aligned}
$$

in which σ_0 is called a backscattering coefficient. In Simulation I, 7 band data of TM will be treated to learn a neural network. Simulation II deals with 7 band data of TM and 1 band data of AMI for training a neural network. In both simulations, an object area to be classified is Chikushi plain in Kyushu Island, Japan, shown in Fig.3. Categories to be classified are paddy, field and orchard, forest, urban and residential area, bare soil, and sea and river. Furthermore, forest is divided into two subcategories to get good classification results. Therefore, the total number of categories is 7.

Figure 3: Landsat 5 TM false color composite image in Chikushi plain, Japan, acquired on April 24, 1997. The image displays band 5 as red, band 4 as green and band 3 as blue. Bird's eye view.

5.2 Simulation I

Input data of the neural network are orthogonal components obtained by applying the principal component analysis to 7-dimensional real vectors corresponding to 7 band data of TM. The training data were chosen from the data whose categories are known in advance.

The structure of the neural network is as follows: the number n of input units of the network is 7, the number h of hidden units is 5, and the number l of output units is 7 which corresponds to the number of categories. The number of training data for each category was chosen as $m_1 = 13$, $m_2 = 12$, $m_3 = 9$, $m_4 = 16$, $m_5 = 16$, $m_6 = 16$, and $m_7 = 11$. Therefore, the total number m of training data is 93. The penalty constants in the cost function (12) were selected as $C_1 = 5$, $C_2 = 15$, $C_3 = 1$, $C_4 = 50$ and $C_5 = 0.5$. As initial values of the weights W_i and V_j, random numbers were chosen. Initial values of thresholds θ_i and η_j, and ε were selected as $\theta_i = 5$, $\eta_j = 0$ and $\varepsilon = 10^{-2}$. We used the steepest descent method as a minimizing process of (12). By the learning of neural network, we can obtain 7 unions each of which consists of several domains of recognition.

Fig.4 illustrates a land cover classification map constructed by estimating which union each pixel of the image data in Chikushi plain is contained in. We succeeded to classify 90.5% pixels in the image data ([8, 9]).

Table 1 shows ratios of land cover in Fig.4 and in the map constructed using Digital National Land Information.

We see from Table 1 that our classification results are close to those in the Land use.

5.3 Simulation II

We use 7 band data of TM and 1 band data of AMI. AMI data in the area where the altitude is higher than $70m$ are prone to be influenced by mountains. So, we cut the mountain area in our analysis. Fig.5 shows an AMI image.

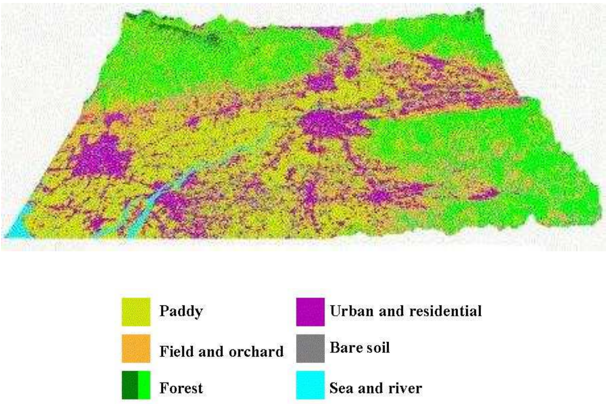

Figure 4: Land cover classification map produced by LDR method for TM image in Fig. 3, displaying by bird's eye view.

Table 1: Ratios of land cover in Fig.4 and in the map constructed using Digital National Land Information

	Category	Our method	Land use[1)2)]
1	Paddy	39.3	34.5
2	Field and orchard	10.5	8.4
3	Forest	30.0	37.7
4	Urban and residential area	16.4	14.1
5	Bare soil	3.0	4.3
6	Sea and river	0.8	1.0
	Total	100.0 [3)]	100.0

1) Digital National Land Information.
2) The values for 1,3,4,5 and 6 indicate total valuesof some subcategories.
3) Unclassified pixels are not included.

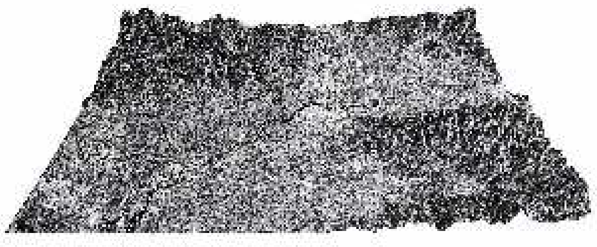

Figure 5: ERS-2 AMI image in Chikushi plain, Japan, acquired on January 17, 1997. The image displays back scatter with a gray scale. Bird's eye view.

The 8 band data are converted into the orthogonal components by the principal component analysis, which are used as input data of a neural network. The number of training data in each category is the same as in Simulation I.

The structure of the neural network is also the same as in Simulation I except for the number 8 of input nodes. We chose the penalty constants in (12) in the same way as in Simulation I. As initial values of weights and thresholds, random numbers were chosen. We employed the steepest descent method as in Simulation I to learn a neural network. By the learning of neural network, we can obtain 7 unions each of which consists of several domains of recognition. Fig.6 illustrates a land cover classification map made by estimating which union each pixel of the image data in Chikushi plain is contained in. We succeeded to classify 92.3% pixels in the image data ([8, 9]). This result is better than in Simulation I.

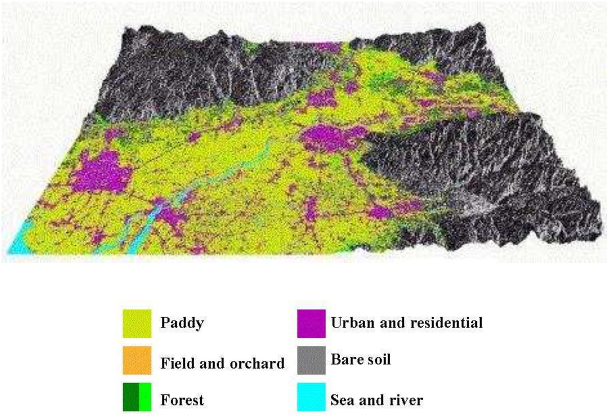

Figure 6: Land cover classification map produced by LDR method for TM and AMI images in Fig. 3 and Fig. 5, displaying by bird's eye view

Table 2 shows ratios of land cover in Fig.6 and in the map constructed using Digital National Land Information.

Table 2 shows that the ratios in paddy, and sea and river are extremely close to those in Land use. The cover ratios in other areas except for forest are almost in the same situation as in Table 1 when comparing with the values in Land use.

6 Conclusions

We proposed a new learning method for three-layered neural networks based on the concept of domains of recognition. The feature of our method is to enlarge the domain by expanding its range in the hidden space and to minimize slopes of affine transforms in the network mapping between the input and hidden layers.

We designed two neural networks: one of them was learnt using 7 band data of TM, and the other learnt using 7 band data of TM and 1 band data of AMI for

Table 2: Ratios of land cover in Figure 6 and in the map constructed using Digital National Land Information

	Category	Our method	Land use[1)2)]
1	Paddy	56.2	57.5
2	Field and orchard	4.9	6.3
3	Forest	15.3	5.8
4	Urban and residential area	19.5	23.7
5	Bare soil	2.6	5.3
6	Sea and river	1.5	1.6
	Total	100.0 [3)]	100.0

1) Digital National Land Information.
2) The values for 1,3,4,5 and 6 indicate total valuesof some subcategories.
3) Unclassified pixels are not included.

land cover classification. As a result, we could get two land cover classification maps. Unclassified pixels on the map constructed using 8 band data decreased in comparison with those made using 7 band data.

Classification ability of neural networks depends on the choice of penalty constants in the cost function and training data. It is a future work how to select the penalty constants and training data in order to improve the classification ability.

References

[1] T.M. Heskes and B. Kappen, "On-line learning processes in artificial neural networks," in *Math. foundations of neural networks.* Amsterdam: Elsevier, 1993. pp.199-233.

[2] B. Fritzke, "Growing cell structure – a self-organizing neural network for unsupervised and supervised learning," *Neural Networks*, vol.7, pp.1441-1460, 1994.

[3] R. Reed, "Pruning algorithms – a survey," *IEEE Trans. Neural Networks*, vol.4, pp.740-747, 1993.

[4] N. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering.* Cambridge: MIT Press, 1996.

[5] M.H. Mohamed and K. Niijima, "Extracting rules from neural networks by removing unnecessary connections," Accepted in Neural Computation 2000 (NC'2000).

[6] M.H. Mohamed, T. Minamoto and K. Niijima, "Convergence rate of minimization learning for neural networks," Lecture Notes in Artificial Intelligence, 1398, Springer (ECML'98), pp.412-417, 1998.

[7] K. Niijima, "Learning of associative memory networks based upon cone-like domains of attraction," *Neural Networks*, vol.10, pp.1649–1658, 1997.

[8] A. Ohkubo and K. Niijima, "A new supervised learning method of neural networks and its application to the land cover classification," *Proc. of IEEE International Geoscience and Remote Sensing Symposium*, June 1999.

[9] A. Ohkubo and K. Niijima, "New supervised learning of neural networks for satellite image classification," *Proc. of IEEE International Conference on Image Processing*, Oct. 1999.