A Log-Linearized Gaussian Mixture Network and Its Application to EEG Pattern Classification

Toshio Tsuji, *Associate Member, IEEE*, Osamu Fukuda, Hiroyuki Ichinobe, and Makoto Kaneko, *Member, IEEE*

***Abstract*— The present paper proposes a new probabilistic neural network (NN) that can estimate *a posteriori* probability for a pattern classification problem. The structure of the proposed network is based on a statistical model composed by a mixture of log-linearized Gaussian components. However, the forward calculation and the backward learning rule can be defined in the same manner as the error backpropagation NN. In this paper, the proposed network is applied to the electroencephalogram (EEG) pattern classification problem. In the experiments, two types of a photic stimulation, which are caused by eye opening/closing and artificial light, are used to collect the data to be classified. It is shown that the EEG signals can be classified successfully and that the classification rates change depending on the number of training data and the dimension of the feature vectors.**

***Index Terms*— Electroencephalography, feedforward neural networks, pattern classification, recurrent nerual networks.**

I. Introduction

An electroencephalogram (EEG) signal pattern changes depending on external or internal factors, such as photic stimulation, auditory stimulation, and intentions of movements. These factors may be used as an interface in virtual reality and teleoperation devices, or as a communication tool for handicapped persons, if the operator's intended movement can be estimated from the EEG pattern.

Until now, some investigations of EEG pattern classification using backpropagation neural networks (NN's) have been carried out [1], [2]. Most of them, however, dealt with research on an automatic diagnosis in a clinic, and only few studies were concerned with developing a new interface tool [3].

The NN based on the error backpropagation learning [4] have been frequently employed in various fields, such as pattern classification and learning control. These NN's can represent any nonlinear mapping between input and output patterns. In case of the pattern classification of unclear EEG signals using backpropagation NN's, the networks need a large number of training data, learning iterations, and a large scale of structure. Also, it is very difficult to attain high classification performance.

In order to improve the generalization ability of the NN's, the present paper proposes a new type of the feedforward probabilistic NN based on the mixture model and the log-linear model of the probability density function (pdf).

Generally, an input for a pattern classification problem can be considered as a stochastic variable with a certain distribution. In this case, the pattern classification problem usually reduces to the estimation problem of the pdf since the classification can be performed according to the Bayes' rule if *a posteriori* probability of the input pattern is obtained accurately.

There are two approaches for estimation of the pdf: parametric and nonparametric approaches. In the parametric approach, a specific type of the pdf is assumed. Then the problem reduces to the parameter estimation of the assumed pdf. Linking this approach to the NN's, Bridle [5] proposed a method that estimates parameters of the pdf as synaptic weights of an NN by assuming the normal distribution for each event. Although the method can estimate the pdf from small sample size data, the discrimination ability considerably decreases when complete knowledge of the pdf is lacking. On the other hand, Specht [6] proposed a general regression NN that computes a pdf based on the Parzen window estimation. Nakagawa and Ono [7] showed that the pdf and the *a posteriori* probability can be estimated from sampled data using the vector quantization and the radial basis function network. These methods are based on the nonparametric approach in which the pdf is estimated by making a multidimensional histogram so that accurate estimation of the pdf requires a large number of sampled data.

Tråvén [8] proposed an NN based on the semiparametric estimation of the pdf, which has a flexible structure to represent any distribution and includes a set of parameters of the specific distribution, and applied his method to a speech recognition problem of 18 Swedish consonants. The mixture model of pdf is a key element of the semiparametric method, which approximates an unknown distribution by the weighted sum of a finite number of component densities. The Gaussian mixture model (GMM) using Gaussian component densities has been often used in tandem with NN's: Tråvén [8], Perlovsky and McManus [9], Tsuji *et al.* [10], Lee and Shimoji [11], and Streit and Luginbuhl [12]. For example, Perlovsky and McManus replaced each component of the GMM with units of the NN, and it was shown that the parameters of the GMM can be estimated through learning and the pdf can be approximated accurately enough. Tsuji *et al.* [10] performed the EMG pattern classification using their network. The six motions of forearm and hand were classified using the EMG signals measured from four pairs of electrodes. Also, Perlovsky and McManus [9], Lee and Simoji [11], and Streit and Luginbuhl [12] carried

Manuscript received May 17, 1996; revised March 4, 1997 and December 7, 1997. This work was supported in part by Tateisi Science and Technology Foundation.

T. Tsuji, O. Fukuda, and M. Kaneko are with the Department of the Industrial and Systems Engineering, Hiroshima University, Higashi-Hiroshima, 739 Japan (e-mail: tsuji@huis.hiroshima-u.ac.jp).

H. Ichinobe is with the Customer Equipment Department, NIPPON Telegraph and Telephone Corporation, Chiyoda, Tokyo, 100-8019 Japan.

Publisher Item Identifier S 1094-6977(99)00102-9.

out the simulation experiments that need complicated decision region boundaries, and the high classification performance were achieved.

However, most of these methods only exploited iterative procedure of the maximum likelihood estimation with forward computation of the *a posteriori* probability. As such, the classification ability of the proposed networks was limited to the same level of the GMM. If the number of components increases, the network needs a lot of sample data to estimate the GMM parameters, such as mixture coefficients, mean values, and standard deviations. Therefore, the classification ability often decreases for a high-dimensional classification problem. Also, Lee and Simoji [11] pointed out the local minima problem in the network learning.

On the other hand, Jordan and Jacobs [13] proposed the hierarchical mixture of expert (HME) that incorporated a generalized linear model in the NN, and they applied it to a control simulation of a four-joint robot arm moving in three-dimensional (3-D) space. The generalized linear model was defined by Nelder and Wedderburn [14]. It includes a large class of useful statistical models, in which the dependent variables are distributed according to some exponential family and can be made linear by a monotonous and differentiable transformation. Using the generalized linear model, the HME consists of two kinds of subnetworks: expert and gating networks. The expert network extracts characteristics of the sampled data depending on a specific part of the input vector space. Outputs from all expert networks are integrated through the mixture coefficients regulated by the gating networks. By introducing the gating networks, the internal structure of the HME becomes hierarchical. It includes much more parameters than the ones of the GMM, which may complicate the learning algorithm and lead to the loss of simplicity of the NN model.

The present paper proposes a new type of the feedforward probabilistic NN for pattern classification problems. The NN is based on the GMM and the log-linear model of the pdf. By applying the log-linear model to a product of the mixture coefficient and the mixture component of the GMM, the semiparametric model of the pdf can be incorporated into the feedforward NN and a simple learning algorithm based on the backpropagation is still applicable.

In the proposed network, the weight coefficients regulated in the learning process correspond to the parameters of the log-linearized GMM (LLGMN) that are the nonlinear combination of the GMM parameters, such as the mixture coefficients, mean values, and standard deviations of each component. The weight coefficient has no longer restricted range like a limitation of the statistical parameters included in the GMM: for instance, standard deviation must be positive. Therefore, the representation ability of the proposed network should be higher than that of the GMM. This is a distinctive feature of the proposed network, and it can be expected to achieve higher classification performance than the ones of previous methods. Also, the network's parameters, such as the activation function of each unit and a number of layers and units, can be determined by the corresponding structure of the GMM incorporated into the network.

This paper is organized as follows. The LLGMN and transformation of this model to the NN are explained in Sections II and III. The classification ability is evaluated by simulations in Section IV. The proposed network is applied to the EEG pattern classification problem in Section V. In Section V, neural filters having a recurrent structure are connected to the proposed network to smooth a variability of the *a posteriori* probabilities. It is shown that the EEG signals can be classified successfully. Finally, Section VI concludes the paper.

II. Log-Linearized GMM

A. Gaussian Mixture Model

In the GMM, a pdf is represented by the weighted sum of a finite number of components with Gaussian densities. Then the pdf estimation problem reduces to estimation of the parameters included in each component. Here, a pdf $f(\mathbf{x})$ of the feature vector $\mathbf{x} \in \Re^d$ is represented by GMM with K classes

$$f(\mathbf{x}) = \sum_{k=1}^{K} \sum_{m=1}^{M_k} \alpha_{k,m} g\left(\mathbf{x}; \boldsymbol{\mu}^{(k,m)}, \Sigma^{(k,m)}\right) \tag{1}$$

$$\sum_{k=1}^{K} \sum_{m=1}^{M_k} \alpha_{k,m} = 1 \tag{2}$$

$$g\left(\mathbf{x}; \boldsymbol{\mu}^{(k,m)}, \Sigma^{(k,m)}\right) = (2\pi)^{-\frac{d}{2}} \left|\Sigma^{(k,m)}\right|^{-\frac{1}{2}} \times \exp\left[-\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu}^{(k,m)}\right)^T \times \left(\Sigma^{(k,m)}\right)^{-1}\left(\mathbf{x} - \boldsymbol{\mu}^{(k,m)}\right)\right] \tag{3}$$

where $g(\mathbf{x}; \boldsymbol{\mu}^{(k,m)}, \Sigma^{(k,m)})$ is the d-dimensional Gaussian distribution. M_k $(k = 1, \ldots, K)$ denotes the number of components of the class k; $\alpha_{k,m}$ denotes the mixture coefficient or the mixing proportion of each component $\{k, m\}$; and $\boldsymbol{\mu}^{(k,m)} \in \Re^d$ and $\Sigma^{(k,m)} \in \Re^{d \times d}$ represent, respectively, the mean vector and covariance matrix of each component $\{k, m\}$. Here, $|\cdot|$ stands for the matrix determinant.

Now, let us briefly discuss the problem of classification of the observed vector $\mathbf{x}$ into one of the K classes. The Bayes' rule determines a specific class if *a posteriori* probability of the vector belonging to the class is larger than the ones of the other classes. Using the GMM, the *a posteriori* probability $P(k \mid \mathbf{x})$ $(k = 1, \ldots, K)$ is given as

$$P(k \mid \mathbf{x}) = \sum_{m=1}^{M_k} P(k, m \mid \mathbf{x}) = \sum_{m=1}^{M_k} \frac{P(k,m) P(\mathbf{x} \mid k, m)}{P(\mathbf{x})} \tag{4}$$

where $P(k, m)$ is *a priori* probability of the class k and the component m, which corresponds to the mixing coefficient $\alpha_{k,m}$; and $P(\mathbf{x} \mid k, m)$ is the pdf of $\mathbf{x}$ agreed upon the class

k and the component m. Then, using (1), the *a posteriori* probability, $P(k,m \mid \mathbf{x})$ can be expressed as

$$P(k,m \mid \mathbf{x}) = \frac{P(k,m)P(\mathbf{x} \mid k,m)}{\sum_{k'=1}^{K}\sum_{m'=1}^{M_{k'}} P(k',m')P(\mathbf{x} \mid k',m')} = \frac{\alpha_{k,m} g\left(\mathbf{x}; \boldsymbol{\mu}^{(k,m)}, \Sigma^{(k,m)}\right)}{\sum_{k'=1}^{K}\sum_{m'=1}^{M_{k'}} \alpha_{k',m'} g\left(\mathbf{x}; \boldsymbol{\mu}^{(k',m')}, \Sigma^{(k',m')}\right)}. \tag{5}$$

In the GMM, the pdf is approximated by the weighted sum of the Gaussian component.

B. Log-Linearization

Using the mean vector $\boldsymbol{\mu}^{(k,m)} = (\mu_1^{(k,m)}, \ldots, \mu_d^{(k,m)})^T$ and the inverse of the covariance matrix $\Sigma^{(k,m)-1} = [s_{ij}^{(k,m)}]$, the numerator of the right side of (5) can be represented as

$$\begin{aligned}
&\alpha_{k,m} g\left(\mathbf{x}; \boldsymbol{\mu}^{(k,m)}, \Sigma^{(k,m)}\right) \\
&\quad = \alpha_{k,m}(2\pi)^{-\frac{d}{2}} \left|\Sigma^{(k,m)}\right|^{-\frac{1}{2}} \\
&\qquad \times \exp\left[-\frac{1}{2}\sum_{j=1}^{d}\sum_{l=1}^{j}(2-\delta_{jl}) s_{jl}^{(k,m)} x_j x_l \right. \\
&\qquad\quad + \sum_{j=1}^{d}\sum_{l=1}^{d} s_{jl}^{(k,m)} \mu_j^{(k,m)} x_l \\
&\qquad\quad \left. - \frac{1}{2}\sum_{j=1}^{d}\sum_{l=1}^{d} s_{jl}^{(k,m)} \mu_j^{(k,m)} \mu_l^{(k,m)} \right]
\end{aligned} \tag{6}$$

where δ_{ij} is the Kronecker delta: $\delta_{ij} = 1$ when $i = j$ and $\delta_{ij} = 0$ when $i \neq j$.

Let us linearize the right side of (6). Taking logarithm of (6), we get

$$\xi_{k,m} \triangleq \log \alpha_{k,m} g\left(\mathbf{x}; \boldsymbol{\mu}^{(k,m)}, \Sigma^{(k,m)}\right) = \boldsymbol{\beta}^{(k,m)^T} \mathbf{X} \tag{7}$$

where $\mathbf{X} \in \Re^H$ and $\boldsymbol{\beta}^{(k,m)} \in \Re^H$ are defined as

$$\mathbf{X} = \left(1, \mathbf{x}^T, x_1^2, x_1x_2, \ldots, x_1x_d, x_2^2, x_2x_3, \ldots, x_2x_d, \ldots, x_d^2\right)^T \tag{8}$$

$$\boldsymbol{\beta}^{(k,m)} = \left(\beta_0^{(k,m)} \sum_{j=1}^{d} s_{j1}^{(k,m)} \boldsymbol{\mu}_j^{(k,m)} \ldots, \sum_{j=1}^{d} s_{jd}^{(k,m)} \boldsymbol{\mu}_j^{(k,m)} - \frac{1}{2} s_{11}^{(k,m)}, -s_{12}^{(k,m)}, \ldots, s_{1d}^{(k,m)}, \ldots, -\frac{1}{2}(2-\delta_{jl}) s_{jl}^{(k,m)}, \ldots, -\frac{1}{2} s_{dd}^{(k,m)} \right)^T \tag{9}$$

$$\beta_0^{(k,m)} = -\frac{1}{2}\sum_{j=1}^{d}\sum_{l=1}^{d} s_{jl}^{(k,m)} \boldsymbol{\mu}_j^{(k,m)} \boldsymbol{\mu}_l^{(k,m)} - \frac{d}{2}\log 2\pi - \frac{1}{2}\log\left|\Sigma^{(k,m)}\right| + \log \alpha_{k,m} \tag{10}$$

and the dimension H is defined as $H = 1 + d(d+3)/2$. We can see that $\xi_{k,m}$ can be expressed as the product of the coefficient vector $\boldsymbol{\beta}^{(k,m)}$ and the modified input vector $\mathbf{X} \in \Re^H$, where the element of the vector $\boldsymbol{\beta}^{(k,m)}$ is a nonlinear combination of the GMM parameters, such as the mixture coefficients, mean values, and standard deviations of the sample data, and the modified input vector $\mathbf{X}$ includes the product of the elements of the input feature vector $\mathbf{x}$.

By regarding the coefficient vector $\boldsymbol{\beta}^{(k,m)}$ as the weight coefficients, the GMM can be incorporated as the network structure. However, the definition of $\boldsymbol{\beta}^{(k,m)}$ (9) indicates that each element of $\boldsymbol{\beta}^{(k,m)}$ is constrained by the statistical properties of the parameter $s_{i,j}^{(k,m)}$. This constraint may cause a difficult problem in the learning procedure: how to satisfy the constraints during the learning of the weight coefficients. In order to remove this difficulty, the new variable $Y_{k,m}$ and the new coefficient vector $\mathbf{w}^{(k,m)}$ are introduced in this paper as

$$\begin{aligned}
Y_{k,m} &\triangleq \xi_{k,m} - \xi_{K,M_k} \\
&= \left(\boldsymbol{\beta}^{(k,m)} - \boldsymbol{\beta}^{(K,M_k)}\right)^T \mathbf{X} = \mathbf{w}^{(k,m)^T} \mathbf{X}
\end{aligned} \tag{11}$$

where the weight coefficient $\mathbf{w}^{(k,m)}$ is defined as the difference between $\boldsymbol{\beta}^{(k,m)}$ and $\boldsymbol{\beta}^{(K,M_K)}$, and $\mathbf{w}^{(K,M_K)} = 0$. Owing to this transformation, the weight coefficient $\mathbf{w}^{(k,m)}$ can have any real number, so that there are no constraints in the learning procedure. Note that this transformation does not result any loss of information in spite of $Y_{K,M_K} = 0$ since the variable $\xi_{k,m}$ in (7) is redundant because of $\sum_{k=1}^{K}\sum_{m=1}^{M_k} P(k,m \mid \mathbf{x}) = 1$. Then the *a posteriori* probability $P(k,m \mid \mathbf{x})$ of (5) can be computed as

$$P(k,m \mid \mathbf{x}) = \frac{\exp[Y_{k,m}]}{\sum_{k'=1}^{K}\sum_{m'=1}^{M_{k'}} \exp[Y_{k',m'}]}. \tag{12}$$

By taking the logarithm of each pdf and using the weight coefficient vector $\mathbf{w}^{(k,m)}$, the *a posteriori* probability can be expressed with the use of the variable $Y_{k,m}$. Note that $Y_{k,m}$ is the linear sum of the modified input vector $\mathbf{X}$ and the coefficient vector $\mathbf{w}^{(k,m)}$. In this paper, coefficient vector $\mathbf{w}^{(k,m)}$ is used as the weight vector and is modified through learning using the teacher vector. Many parameters of the GMM, such as the mixing coefficient $\alpha_{k,m}$, the mean vector $\boldsymbol{\mu}^{(k,m)}$, and the covariance matrix $\Sigma^{(k,m)}$, are replaced by arbitrary parameters $\mathbf{w}^{(k,m)}$. Owing to this replacement, the network can learn $\mathbf{w}^{(k,m)}$ over the structure of the GMM.

It can be seen that, if the number of components M_k for each class is given adequately, the elements of the input vector $\mathbf{X}$ and the weight vector $\mathbf{w}^{(k,m)}$ are determined automatically in correspondence to the characteristics of the pdf. The number of components can be determined arbitrarily, or based on the information on the pdf of the sample data. In the next section, the LLGMN is developed as a feedforward NN.

III. NN Model

A. Network Architecture

The structure of the NN we use in our study is shown in Fig. 1. The network is of feedforward type and contains three layers. First, the input feature vector $\mathbf{x} \in \Re^d$ is preprocessed and converted into the modified input vector $\mathbf{X} \in \Re^H$ according to (8). The first layer consists of H units corresponding

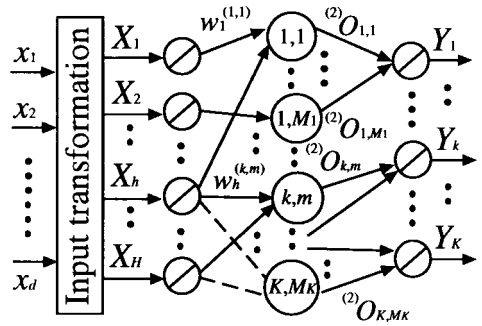


Fig. 1. Structure of an LLGMN.

to the dimension of $\mathbf{X}$, and the identity function is used for activation of each unit. The relationship between input and output of each unit in the first layer is defined as

$$^{(1)}I_j = X_j \tag{13}$$

$$^{(1)}O_j = {}^{(1)}I_j \tag{14}$$

where $^{(1)}I_j$ and $^{(1)}O_j$ denote the input and the output, respectively, of the jth unit in the first layer.

The second layer consists of the same number of units as the total component number of the GMM $\sum_{k=1}^{K} M_k$. Each unit receives the output of the first layer weighted by the coefficient $w_h^{(k,m)}$ and outputs the *a posteriori* probability of each component according to (12). The input to the unit $\{k,m\}$ in the second layer $^{(2)}I_{k,m}$ and the output $^{(2)}O_{k,m}$ are defined as

$$^{(2)}I_{k,m} = \sum_{h=1}^{H} {}^{(1)}O_h w_h^{(k,m)} \tag{15}$$

$$^{(2)}O_{k,m} = \frac{\exp\left[{}^{(2)}I_{k,m}\right]}{\sum_{k'=1}^{K}\sum_{m'=1}^{M_K}\exp\left[{}^{(2)}I_{k',m'}\right]} \tag{16}$$

where $w_h^{(K,M_k)} = 0$ $(h = 1,\ldots,H)$. It should be noted that (16) can be considered as a kind of the generalized sigmoid functions.

Finally, the third layer consists of K units corresponding to the number of classes and outputs the *a posteriori* probability of the class k $(k = 1,\ldots,K)$. The unit k integrates the outputs of M_k units $\{k,m\}$ $(m = 1,\ldots,M_k)$ in the second layer. The relationship between the input and the output is defined as

$$^{(3)}I_k = \sum_{m=1}^{M_k} {}^{(2)}O_{k,m} \tag{17}$$

$$Y_k = {}^{(3)}I_k. \tag{18}$$

In the LLGMN defined above, the *a posteriori* probability of each class is defined as outputs of the last layer. Note that, the log-linearized Gaussian mixture structure is incorporated in the network through learning only the weight coefficient $w_h^{(k,m)}$.

B. Learning Rule

Now, let us consider the supervised learning with the teacher vector $\mathbf{T}^{(n)} = (T_1^{(n)},\ldots,T_k^{(n)},\ldots,T_K^{(n)})^T$ for the nth input vector $\mathbf{x}^{(n)}$. When the teacher provides perfect classification, $T_k^{(n)} = 1$ for the particular class k and $T_k^{(n)} = 0$ for all the other classes. Using the *a posteriori* probability $P(k \mid \mathbf{x}^{(n)})$, the probability that the teacher vector $\mathbf{T}^{(n)}$ is observed for the input vector $\mathbf{x}^{(n)}$ is given by

$$P(\mathbf{T}^{(n)}) = \prod_{k=1}^{k} P(k \mid \mathbf{x}^{(n)})^{T_k^{(n)}}. \tag{19}$$

The network is trained using a given set of N data $\mathbf{x}^{(n)}(n = 1,\ldots,N)$. Using the training data, a log-likelihood function L can be derived from (18) and (19) as

$$L = \sum_{n=1}^{N}\sum_{k=1}^{K} T_k^{(n)} \log Y_k^{(n)} \tag{20}$$

where the nth output $Y_k^{(n)}$ of the LLGMN corresponds to $P(k \mid \mathbf{x}^{(n)})$. As an energy function for the network, we use

$$J = \sum_{n=1}^{N} J_n = -\sum_{n=1}^{N}\sum_{k=1}^{K} T_k^{(n)} \log Y_k^{(n)} \tag{21}$$

and the learning is performed to minimize it, that is, to maximize the likelihood. For $\mathbf{x}^{(n)}$, the weight modification $\Delta w_h^{(k,m)}$ of the corresponding weight $w_h^{(k,m)}(h = 1,\ldots,H)$ is defined as

$$\Delta w_h^{(k,m)} = -\eta \frac{\partial J_n}{\partial w_h^{(k,m)}} \tag{22}$$

in a sequential learning scheme, and

$$\Delta w_h^{(k,m)} = -\eta \sum_{n=1}^{N} \frac{\partial J_n}{\partial w_h^{(k,m)}} \tag{23}$$

in a collective learning scheme. Here, $\eta > 0$ is the learning rate and the partial derivative in (24) and (25) can be obtained using the chain rule in the same manner as the error backpropagation rule [4]

$$\begin{aligned}\frac{\partial J_n}{\partial w_h^{(k,m)}} &= \frac{\partial}{\partial w_h^{(k,m)}}\left(-\sum_{k=1}^{K} T_k^{(n)} \log Y_k^{(n)}\right) \\ &= -\sum_{k'=1}^{K} \frac{\partial T_{k'}^{(n)} \log Y_{k'}^{(n)}}{\partial Y_{k'}^{(n)}} \\ &\quad \times \sum_{m'=1}^{M_k} \frac{\partial Y_{k'}^{(n)}}{\partial {}^{(2)}O_{k',m'}^{(n)}} \frac{\partial {}^{(2)}O_{k',m'}^{(n)}}{\partial {}^{(2)}I_{k,m}^{(n)}} \frac{\partial {}^{(2)}I_{k,m}^{(n)}}{\partial w_h^{(k,m)}} \\ &= \left(Y_k^{(n)} - T_k^{(n)}\right) \frac{{}^{(2)}O_{k,m}^{(n)}}{Y_k^{(n)}} X_h^{(n)}.\end{aligned} \tag{24}$$

If the teacher signal for each component is available, the partial derivative of (24) can be replaced as

$$\begin{aligned}\frac{\partial J_n}{\partial w_h^{(k,m)}} &= \frac{\partial}{\partial w_h^{(k,m)}}\left(-\sum_{k=1}^{K}\sum_{m=1}^{M_K} T_{k,m}^{(n)} \log {}^{(2)}O_{k,m}^{(n)}\right) \\ &= -\sum_{k'=1}^{K}\sum_{m'=1}^{M_k} \frac{\partial T_{k',m'}^{(n)} \log {}^{(2)}O_{k',m'}^{(n)}}{\partial {}^{(2)}O_{k',m'}^{(n)}} \\ &\quad \times \frac{\partial {}^{(2)}O_{k',m'}^{(n)}}{\partial {}^{(2)}I_{k,m}^{(n)}} \frac{\partial {}^{(2)}I_{k,m}^{(n)}}{\partial w_h^{(k,m)}} \\ &= \left({}^{(2)}O_{k,m}^{(n)} - T_{k,m}^{(n)}\right) X_h^{(n)} \end{aligned} \tag{25}$$

where $T_{k,m}^{(n)}$ is the teacher signal for the component $\{k,m\}$. Note that $T_{k,m}^{(n)} = 1$ if the sample $\mathbf{x}^{(n)}$ belongs to the component $\{k,m\}$ and $T_{k,m}^{(n)} = 0$, otherwise. In the pattern classification problems, the teacher signal usually defines assignment of different classes. It can be seen from the learning rule (24) that the teacher signals given for the classes are propagated backward into each component according to the ratio of the *a posteriori* probability ${}^{(2)}O_{k,m}^{(n)}$ of each component to the *a posteriori* probability $Y_k^{(n)}$ of the class k.

The learning rule derived in this paper can be applied using, not only the discrete teacher signal of $\{0,1\}$, but the probabilistic or fuzzy teacher signal that takes continuous values of $[0,1]$. It should be noted that the conditions of $\sum_{k=1}^{K} T_k^{(n)} = 1$ and $T_{k,m}^{(n)} \geq 0$ must be held. Now, when the teacher signal vector $\mathbf{T}^{(n)}$ and the output vector $\mathbf{Y}^{(n)} = (Y_1^{(n)}, \ldots, Y_k^{(n)}, \ldots, Y_K^{(n)})^T$ of the LLGMN are given, the energy function for learning is changed as

$$\begin{aligned} J &= \sum_{n=1}^{N} I(\mathbf{T}^{(n)}; \mathbf{Y}^{(n)}) = \sum_{n=1}^{N}\sum_{k=1}^{K} T_k^{(n)} \log \frac{T_k^{(n)}}{Y_k^{(n)}} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} T_k^{(n)} \log T_k^{(n)} - \sum_{n=1}^{N}\sum_{k=1}^{K} T_k^{(n)} \log Y_k^{(n)} \\ &\geq 0 \end{aligned} \tag{26}$$

where $I(\cdot;\cdot)$, which arises here as a difference between $\mathbf{T}^{(n)}$ and $\mathbf{Y}^{(n)}$, was introduced by Kullback and Leibler as a measure of directed divergence between two probability distributions [15]. We have with equality if, and only if, $T_k^{(n)} = Y_k^{(n)}$ for all k, n.

Note that the first term in (26) is constant. Thus, the second term should be minimized to make $\mathbf{Y}^{(n)}$ close to $\mathbf{T}^{(n)}$. Since the second term is equal to the energy function J of (21), the learning rules (22)–(25) derived for the perfect teacher signal also minimize the Kullback information.

IV. Simulation Experiments

A. Generalization Ability

To compare the generalization ability of the LLGMN with that of the error backpropagation NN (BPN), pattern classification experiments are carried out using two-dimensional (2-D) data $(d = 2, H = 6)$ for two classes $(K = 2)$. The data are artificially generated using the Gaussian mixture pdf with two components $(M_1 = M_2 = 2)$. Table I indicates the parameters used in the GMM.

TABLE I
Parameters of the Gaussian Mixture Model Used in the Experiments

		$\alpha_{k,m}$		$\mu^{(k,m)\mathrm{T}}$		$\Sigma^{(k,m)}$	
component, m		1	2	1	2	1	2
class, k	1	0.3	0.2	[0.0,0.0]	[5.0,7.0]	$\begin{bmatrix} 9.0 & 0.63 \\ 0.63 & 0.09 \end{bmatrix}$	$\begin{bmatrix} 1.0 & -0.5 \\ -0.5 & 1.0 \end{bmatrix}$
	2	0.25	0.25	[2.0,2.0]	[−6.0,2.0]	$\begin{bmatrix} 1.0 & 2.7 \\ 2.7 & 9.0 \end{bmatrix}$	$\begin{bmatrix} 0.09 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$

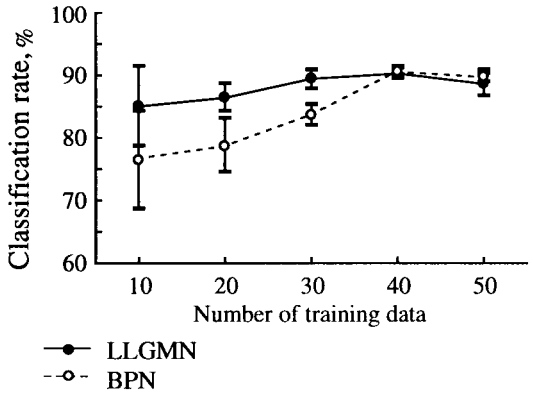


Fig. 2. Effect of the number of training data on classification ability.

The LLGMN includes four units in the second layer that correspond to the total component number, six in the input layer, and two in the output layer. The BPN includes two units with the identity activation functions in the input layer; ten units with the sigmoid functions in each of the two hidden layers; and two units with the sigmoid functions in the output layer. Learning for both the networks is carried out until the energy function J_n of (21) becomes less than 0.1 for all training data. The teacher signal is given for each class [see (22) and (24); $\eta = 0.0001$]. For the BPN, two outputs are normalized to make the sum of the outputs one.

Fig. 2 indicates changes in the classification rate as the number of the training data increases. Both the networks are trained independently using five sets of the training data. The numbers of the training data are 10–50. After learning, 2000 data that are not used in learning are prepared for classification. Then, the ratio of the correct classification to 2000 data (1000 for each class) is computed. In Fig. 2, the mean values and the standard deviations of the classification rate for ten kinds of initial weights that are randomly chosen are plotted. Although both networks can achieve a high classification rate for a large number of training data, the difference becomes clear as the number of the training data decreases. The LLGMN keeps the classification rate high enough even for a small sample size of the training data, whereas the classification rate of the BPN significantly decreases.

Decision region boundaries, on which the *a posteriori* probabilities of both classes become equal, are shown in Fig. 3. In the BPN, the decision region boundaries are varied largely with the number of the training data. On the other hand, similar

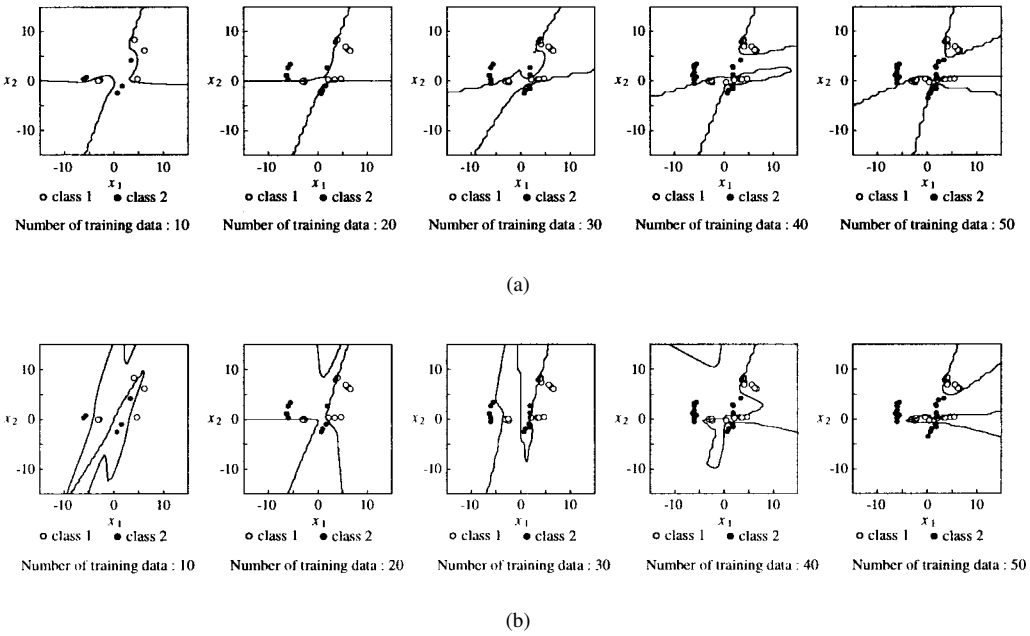


Fig. 3. Scatter diagram of training data and decision region boundaries: (a) LLGMN and (b) BPN.

decision region boundaries can be obtained by the LLGMN in spite of changes in the number of the training data.

B. Representation Ability

The parameters of the GMM, such as the mixing coefficient $\alpha_{k,m}$, the mean vector $\boldsymbol{\mu}^{(k,m)}$, and the covariance matrix $\Sigma^{(k,m)}$, include several constraints. For example, the covariance matrix must be invertible, the mixing coefficient $\alpha_{k,m}$ must be positive, and the total sum of the mixing coefficients must be one. On the other hand, the weight coefficients $w_h^{(k,m)}$ used in the LLGMN have no such constraints and are mutually independent. To evaluate this difference, the LLGMN is compared with the maximum likelihood artificial neural system (MLANS) [9] that was developed by the direct use of the GMM.

The classification capability of two networks are evaluated in the 2-D feature space $(d = 2, H = 6)$ for three classes $(K = 3)$, A, B, C. In the feature space, the classes A and B are represented by a single rectangular region and class C has two such regions. The pdf is constant in every region, and the *a priori* probabilities of three classes are the same. Also, two regions belonging to class C have the same *a priori* probabilities. An example of the training data with four rectangular regions is shown in Fig. 4.

The LLGMN includes six units in the input layer and three in the output layer. In the second layer, the number of units is equal to the total number of MLANS's components. The teacher signal is given for each class [see (22) and (24); $\eta = 0.5$], and learning is carried out until the mean value of the energy function J_n of (21) for all training data becomes less than 0.5. Note that, for three training sets, including 210, 270, and 330 data, dynamics of the terminal attractor [18] are incorporated in the learning rule to speed the learning procedure. The terminal attractor is based on the concept that

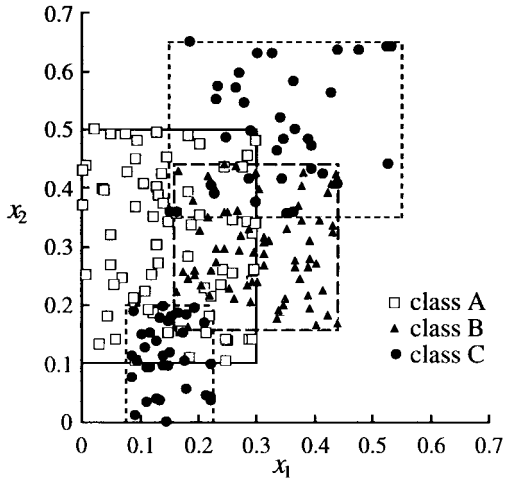


Fig. 4. Scatter diagram of 210 training data.

the Lipschitz conditions are violated at the equilibrium point. The network learning converges to the equilibrium point, that is, the global minimum or one of local minima, in a finite specified time [16].

On the other hand, in MLANS, the learning procedure is continued until the change of the Bhattacharyya distance [9] of the *a posteriori* probability with one iteration becomes less than 0.0001. For evaluating the classification ability, 3000 data (1000 for each class), which are different from the training data, are artificially generated.

Fig. 5 shows the classification result when the number of the training data is varied from 30 to 330, where the mean values and the standard deviations of the classification rates for ten kinds of initial weights are plotted. The solid line and the dashed line show the results of the LLGMN and the MLANS, respectively. Note that both the number of

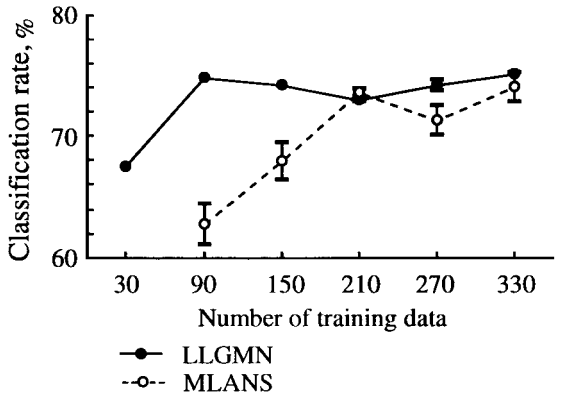


Fig. 5. Effect of the number of training data on classification ability.

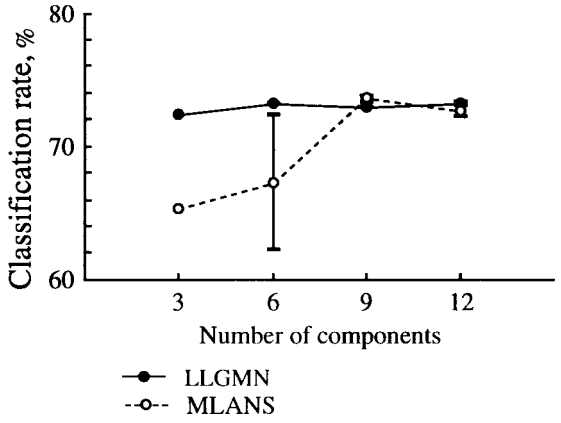


Fig. 6. Effect of the number of components on classification ability.

components used in the MLANS and the number of units in the second layer of the LLGMN are nine, i.e., three for each class. When the number of the training data is sufficiently large, the classification rates of both the networks are almost the same. However, as the number of the training data decreases, the classification rate of the MLANS becomes worse than the one of the LLGMN. Note that the covariance matrices included in the MLANS cannot be estimated in the case of 30 training data because the number of the data belonging to each component decreases remarkably when the number of the training data is small.

Next, Fig. 6 shows the classification result when the total number of components is varied from three to 12. The number of the training data is 210, i.e., 70 for each class, and the same convergence conditions as in the previous experiment are used. By using the MLANS (the dotted line), classification is performed successfully when the number of components is large enough. For the small number of components, however, it becomes difficult to represent the data distribution adequately, and the classification rate decreases. The standard deviation of the classification rate of the MLANS becomes large when the number of components is six since different learning results have been obtained, depending on the initial weights. On the other hand, with the use of the LLGMN, classification remains satisfactory even if the number of components is small.

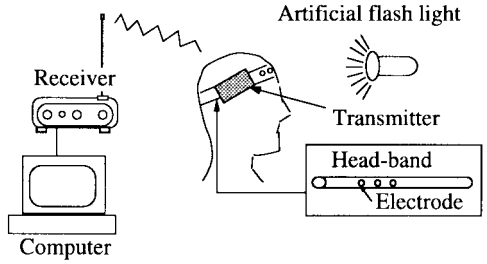


Fig. 7. Overview of the experimental apparatus.

V. EEG Pattern Classification

A. Experiments

The pattern classification of EEG signals is carried out under the photic stimulation by eye opening/closing and artificial light, as shown in Fig. 7.

1) Experimental Apparatus: To evaluate the possibility of the EEG signals as a human interface tool, simple and handy electroence phalograph (IBVA, Random ELECTRONICS DESIGN) is used. This enables us to measure EEG signals in usual environments. The experimental system consists of the headband, transmitter, and receiver.

The transmitter is attached to the headband. The EEG signals measured from the electrodes are digitized by an A/D converter (the sampling frequency $= 120$ Hz, quantization $= 8$ bits) after they are amplified and filtered out through low-cut (3 Hz) and high-cut (40 Hz) analogue filters. The size of the transmitter is quite compact ($93 \times 51 \times 25$ mm). The personal computer, which is connected to the receiver, collects data. The surface electrodes are located at Fp1 and Fp2, which are specified by the International 10–20 Electrode System. The noise in the EEG signals can be removed significantly by the bipolar derivation between the two electrodes at Fp1 and Fp2.

2) Experimental Conditions: The EEG signals are measured under the two following conditions.

a) Photic stimulation by opening and closing eyes:
Subjects are seated in a well-lit room. First, EEG signals are measured during both eye opening and closing (60 s for each). The measured signals are used as training data. Next, subjects are asked to switch their eye states alternatively according to the pseudorandom series for 450 s.

b) Photic stimulation by an artificial light:
Subjects are seated in a dark room and open their eyes. A flash light (xenon, illuminating power: 0.176 [J]) is set at the distance of 50 cm apart from their eyes. The light turning on and off with the frequency 4 Hz is used as the artificial photic stimulation.

The electroence phalograph used in the experiments has one pair of the electrodes, so that the spatial information of the EEG signals on the location of the electrodes cannot be utilized. The frequency characteristics of the EEG signals, however, significantly change depending on the eye states. Therefore, the spectral information of the measured EEG signals are used as follows. The power spectral density

TABLE II
FREQUENCY RANGE USED IN THE CLASSIFICATION EXPERIMENTS

d	Frequency ranges (Hz)					
2	0 ~ 8	9 ~ 35	-	-	-	-
3	0 ~ 8	9 ~ 20	21 ~ 35	-	-	-
4	0 ~ 8	9 ~ 12	13 ~ 20	21 ~ 35	-	-
5	0 ~ 4	5 ~ 8	9 ~ 12	13 ~ 20	21 ~ 35	-
6	0 ~ 2	3 ~ 4	5 ~ 8	9 ~ 12	13 ~ 20	21 ~ 35
	x_1	x_2	x_3	x_4	x_5	x_6

d : Dimension of the input vector

function of the measured EEG signal is estimated using the Fast Fourier transform (FFT) for every 128 sampled data. The function is divided into several ranges (from 0 to 35 Hz). The frequency bands of this range are determined based on the clinical use of the brain wave (delta, theta, alpha, beta). Time series of the mean values of the power spectral density function within each frequency ranges are calculated and normalized between $[0, 1]$ in each range. Thus, the multidimensional data $(x_1, x_2, \ldots, x_d)$ are obtained and used as the input vector to the networks. Here, d denotes the number of the frequency ranges. The frequency ranges used in the experiments are shown in Table II.

To compare the LLGMN with other NN's, the pattern classification experiments are conducted using four types of networks: the LLGMN, MLANS, and two types of the BPN's (with one or two hidden layers). In the LLGMN, the first layer consists of H units and the third layer consists of two units corresponding to the number of classes. The second layer consists of the six units (three for each class) corresponding to the total component number of the GMM. On the other hand, in the BPN's, the first layer consists of d units and each hidden layer consists of 15 units. In the MLANS, the learning procedure is continued until the change of the Bhattacharyya distance [9] of the *a posteriori* probabilities with one iteration becomes less than 0.0001. On the other hand, in BPN's, the learning procedure is continued until the mean square error becomes less than 0.1. However, if the mean square error after 50 000 iterations does not become less than 0.2, the learning procedure is stopped.

B. Classification Results

1) EEG Classification of Eye States: To examine the classification ability of the networks, experiments are performed for five subjects (A, B, C, D: males, E: female). Each network is trained using 112 data (56 for each class). Then, the ratio of the correct classification to 422 data that are not used in learning is computed.

Fig. 8 shows the classification result of the LLGMN (subject A). In this figure, the 2-D input vector is used as shown in Table II $(d = 2, H = 6, K = 2)$.

In the figure, the timing of switching eye states, the input pattern to the LLGMN (x_1, x_2), the output of the network $({}^{(3)}O_1, {}^{(3)}O_2)$, and the classification results are shown. As can be seen, the LLGMN achieves considerably high performance with 91.1% of the classification rate. The misclassified data are observed immediately after switching eye states from opening to closing.

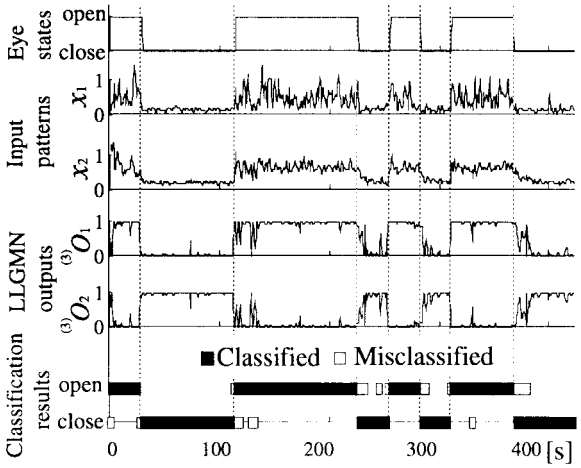


Fig. 8. Classification results of eye states by the LLGMN.

TABLE III
CLASSIFICATION RESULTS OF EYE STATES

Subject	Performance	BPN with 1 hidden layer	BPN with 2 hidden layers	MLANS	LLGMN
A	*Rcla*	72.6	84.3	85.5	91.1
(male)	*Rsd*	12.1	3.4	2.1	0.4
	Rcon	53.3	86.7	100.0	100.0
B	*Rcla*	76.2	84.4	83.7	83.3
(male)	*Rsd*	6.1	3.1	0.7	0.6
	Rcon	73.3	83.3	100.0	100.0
C	*Rcla*	81.6	88.7	89.9	88.6
(male)	*Rsd*	5.4	2.7	0.4	1.4
	Rcon	80.0	83.3	100.0	100.0
D	*Rcla*	73.4	78.4	80.6	81.3
(male)	*Rsd*	5.7	3.7	0.5	1.1
	Rcon	56.7	60.0	100.0	100.0
E	*Rcla*	86.3	89.7	90.6	93.2
(female)	*Rsd*	6.7	3.1	1.6	0.6
	Rcon	56.7	60.0	100.0	100.0

Rcla : Classification rate(%), *Rsd* : Standard deviation(%)
Rcon: Convergence rate(%)

Table III shows classification results for all five subjects. The mean values and the standard deviations of the classification rate for 30 kinds of initial weights, which are randomly chosen, are shown. The convergence rate is defined as the ratio of the number of the converged learnings to 30 trials.

As can be expected for all subjects, the convergence rates of the MLANS and the LLGMN are greater than the ones of the BPN's. In the BPN's, the mean values of the convergence rate are always less than that of the LLGMN, where the convergence rates of the MLANS and the LLGMN are 100%. Also, the standard deviations of the classification rates of the LLGMN are quite small.

Next, we examine the changes of the classification rates with the number of the training data N and the dimension of the input vector d taken from Table II. For each input vector, the number of training data N is changed from ten to 100.

Here, the pattern classification results carried out using the LLGMN and the MLANS are shown. Both the networks are trained using 50 sets of the training data $(N = 10, 20, \ldots, 100, d = 2, 3, \ldots, 6)$. Then the ratio of the correct classification to 422 data, which are not used in learning, is computed. Figs. 9

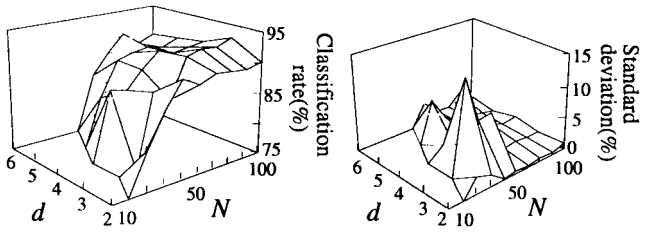


Fig. 9. Effect of the training data on classification results of eye states by MLANS.

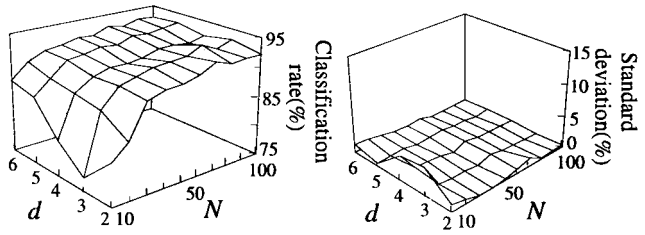


Fig. 10. Effect of the training data on classification results of eye states by LLGMN.

and 10 show the mean values and the standard deviations of the classification rate for ten kinds of the initial weights. Although both networks can achieve a high classification rate for a large number of training data, the difference becomes clear as the number of the training data decreases. The LLGMN keeps the classification rate high even for a small sample size of the training data, whereas the classification rate of the MLANS decreases. Note that the covariance matrices included in the MLANS cannot be estimated in some cases $(N = 10, \ldots, 40)$ because the number of the data belonging to each component decreases remarkably when the number of the training data is small. The statistical structure incorporated in the LLGMN realizes considerably high classification ability for even a small sample size of the training data.

Also, the classification rates of the LLGMN with a sufficient number of the input vector d are relatively high even if the number of the training data decreases. On the other hand, the classification rate of the MLANS decreases considerably in those cases and the standard deviations of the classification rates are much greater than those of the LLGMN.

2) EEG Classification of the Artificial Photic Stimulation: Next, the pattern classification experiments are carried out using the artificial photic stimulation. An example of the classification result is shown in Fig. 11(a). In this case, the classification performance decreases considerably compared to Fig. 8. Two-dimensional learning data ($d = 2$ in Table I) are shown in Fig. 11(b), in which 100 data (50 for each class) are plotted. Although the EEG patterns changed largely, depending on the opening and closing of eyes, as shown in Fig. 8, the evoked potential is not observed clearly from the EEG signal measured by only a pair of the electrodes for the artificial photic stimulation. Also, uncomfortable feelings of the subjects may act as an artifact. As a result, part of the distributions overlapped each other, so that it seems to be difficult to classify the EEG data into the different classes without any consideration on the temporal properties of the EEG signals.

Table IV shows experimental results for five subjects. The dimensions of the input vector $d = 2, 6$ and the number of the training data $N = 50\,100$ are used in the learning procedure. The ratio of the correct classification to 422 data, which are not used in learning, is computed. Compared to the classification result of the eye states, the classification rates under the artificial photic stimulation decrease. Although the difference among the individuals can be observed, the classification rates tend to improve with the increase of the number of the training data from $N = 50$ to $N = 100$ and the dimension of the input vector from $d = 2$ to $d = 6$. Also, the standard deviations of the classification rates tend to decrease.

Fig. 12 shows the effect of the training data on the classification results of subject A. The classification rates slightly improve with the increase of the dimension of the input vector. On the other hand, any improvement of the classification rates, depending on the number of the training data, is not observed. Compared to the classification results of the eye states, the classification rates under the artificial photic stimulation decrease. This is because the EEG patterns change considerably, depending on the time of the artificial photic stimulation. To remedy this problem, we propose to use a kind of neural filter (NF) [19]. They should be connected to the third layer of the LLGMN to take into account the history of the EEG patterns.

C. Introduction of Neural Filter

The NF is introduced to cope with time-varying characteristics of the EEG signals and to classify them accurately. A number of NN structures may be suitable for the NF. In the proposed scheme, the NF deals with a single-input/single-output signal processing and a simple and compact structure is desirable. Lo [19] proposed the NF with one hidden layer of fully interconnected neurons for filtering signals, including nonlinear input/output relationships. He reported that the NF with only a few hidden neurons consistently outperforms the extended Kalman filter in the simulation experiments. This type of the NF is incorporated into the proposed network.

Using the NF, the present paper proposes the following two-step approach. First, the *a posteriori* probability of the input vector belonging to each class is calculated with the use of the LLGMN. Next, the NF's that are connected to the LLGMN receive this *a posteriori* probability, and they make it smoother. The characteristics of the NF can be changed flexibly through the learning. This makes the NF different from the conventional digital filters.

Fig. 13 shows the structure of the proposed network combining the LLGMN with the NF. First, the EEG signal is preprocessed. Then, the first layer of the LLGMN receives the input vector $\mathbf{x} \in \Re^d$. The third layer outputs $\mathbf{Y} \in \Re^K$ to the NF. The outputs of the NF are normalized and considered as the *a posteriori* probability. Finally, the Bayes' rule is used to determine the specific class.

1) Structure of the Neural Filter: Fig. 14 shows the structure of the NF. The unit in the first layer receives the input

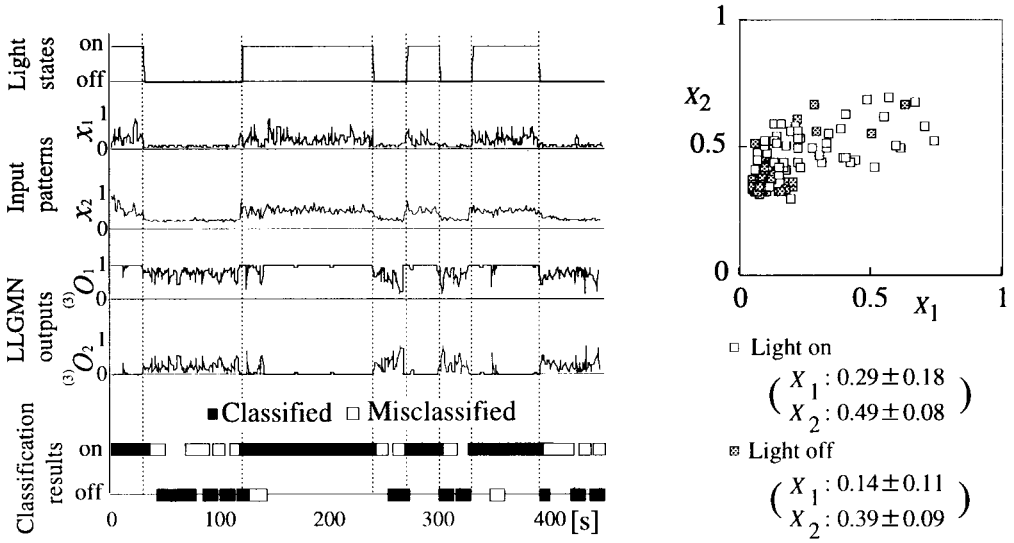
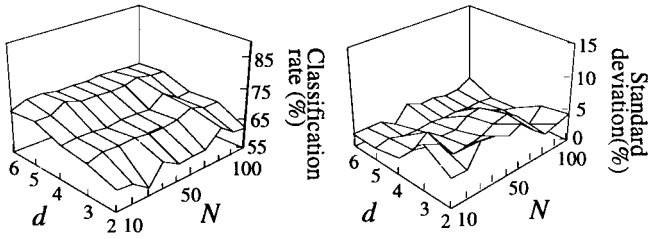


Fig. 11. Classification results under the artificial photic stimulation by LLGMN.

TABLE IV
CLASSIFICATION RESULTS UNDER THE ARTIFICIAL PHOTIC STIMULATION

Number of the learning data		N =50		N =100	
Dimension of the input vector		d =2	d =6	d =2	d =6
subject A (male)	R_{cla}	60.1	67.6	62.4	69.9
	R_{sd}	6.0	2.9	4.4	2.6
subject B (male)	R_{cla}	81.1	82.6	83.7	84.5
	R_{sd}	3.2	3.2	3.4	1.7
subject C (male)	R_{cla}	62.5	64.9	70.7	71.9
	R_{sd}	1.8	2.2	1.2	1.7
subject D (male)	R_{cla}	67.4	72.4	74.4	76.7
	R_{sd}	3.0	1.2	2.1	1.7
subject E (male)	R_{cla}	67.7	71.6	73.8	75.5
	R_{sd}	4.2	1.7	2.5	1.2

R_{cla} : Classification rate(%), R_{sd} : Standard deviation(%)



d : Number of the input vector, N : Number of the training data

Fig. 12. Effect of the training data on classification results of artificial photic stimulation.

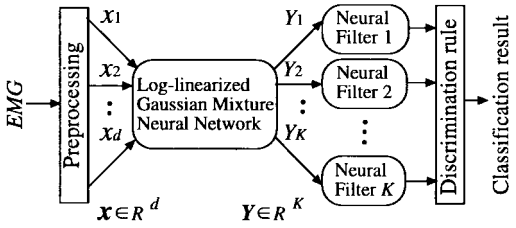


Fig. 13. Structure of the network. The neural filters are serially connected to the LLGMN.

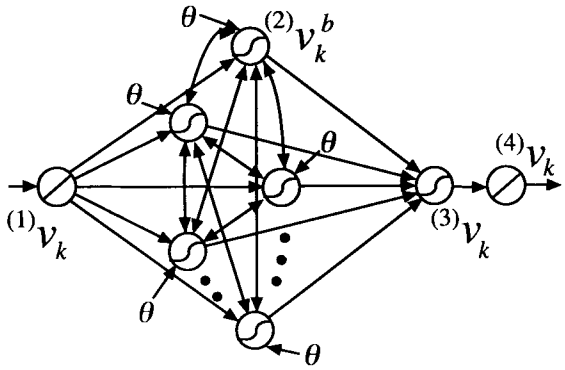


Fig. 14. Structure of the neural filter.

$^{(1)}r_k^{(n)}$ corresponding to the nth outputs $Y_k^{(n)}$ of the LLGMN, and sends $^{(1)}v_k^{(n)}$ to the second layer. The identity function is used for the activation function in the first layer.

The second layer consists of B units. Each unit receives the nth output of the first layer and the $(n-1)$th output of the second layer. Also, this layer has the bias input $(\theta = 1)$. The fully interconnected units keep the internal representation, so that the time history of the input data can be considered. The input to the unit b in the second layer $^{(2)}r_k^{b(n)}$ and the output $^{(2)}v_k^{b(n)}$ are defined as

$$^{(2)}r_k^{b(n)} = \sum_{a=1}^{B} {}^{(2,2)}u_k^{a,b}\, {}^{(2)}v_k^{a(n-1)} + {}^{(1,2)}u_k^{b}\, {}^{(1)}v_k^{(n)} + {}^{(\theta)}u_k^{b} \tag{27}$$

$$^{(2)}v_k^{b(n)} = g\left({}^{(2)}r_k^{b(n)}\right) \tag{28}$$

where $^{(2,2)}u_k^{a,b}$, $^{(1,2)}u_k^{b}$ and $^{(\theta)}u_k^{b}$ denote the weight coefficients between the ath and the bth unit in the second layer, between the first layer and the bth unit in the second layer, and between the bias input and the bth unit in the second layer, respectively. The unit in the third layer is connected to all the units in the second layer, and the relationship between the

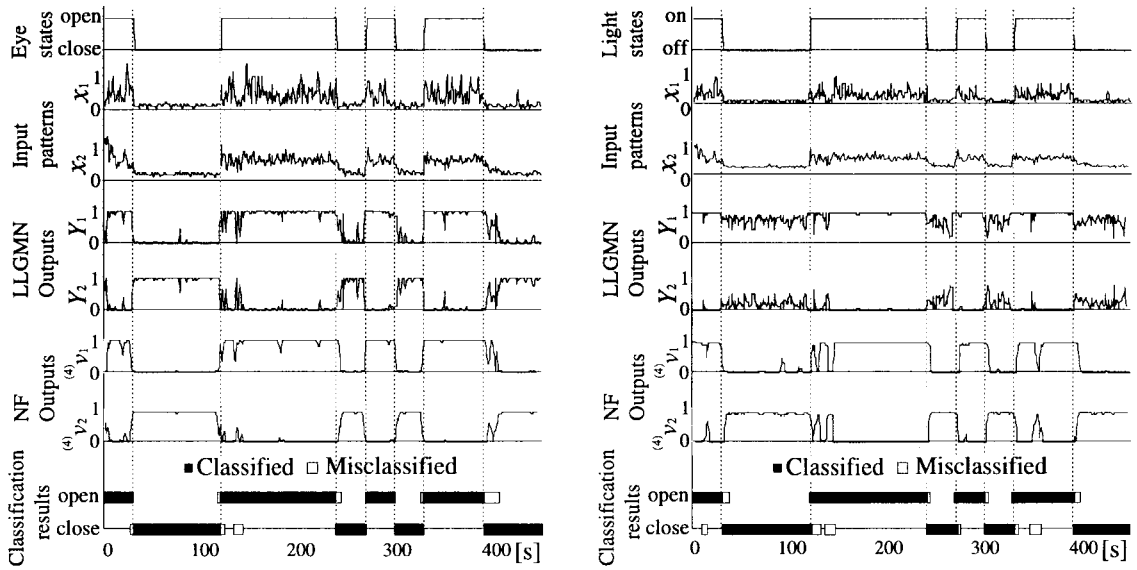


Fig. 15. Classification results.

input and the output of each unit is defined as

$$ {}^{(3)}r_k^{(n)} = \sum_{a=1}^{B} {}^{(2,3)}u_k^a {}^{(2)}v_k^{a(n)} \tag{29} $$

$$ {}^{(3)}v_k^{(n)} = g\left({}^{(3)}r_k^{(n)}\right) \tag{30} $$

where ${}^{(3)}r_k^{(n)}$ and ${}^{(3)}v_k^{(n)}$ denote the input and the output in the third layer, respectively, and ${}^{(2,3)}u_k^a$ denotes the weight coefficient between the ath unit in the second and third layers.

The identity function is used as the activation function in the fourth layer, and the output is defined as

$$ {}^{(4)}v_k^{(n)} = {}^{(3,4)}u_k {}^{(3)}v_k^{(n)} \tag{31} $$

where ${}^{(3,4)}u_k$ denotes the weight coefficient between the third and fourth layers. Note that the weight coefficient ${}^{(3,4)}u_k$ functions as a gain.

2) Learning Schedule: If the teacher signal is given only to the output unit in the NF, the error may backpropagate from the NF to the LLGMN, so that the learning is performed for both networks at the same time. However, the appropriate error backpropagation between the NF and the LLGMN could not be guaranteed because of the redundancy of the network structure.

Therefore, we introduce the following two-step learning schedule that divides the learning into the LLGMN and the NF. First, the LLGMN is trained using the training data to represent the statistical model. Then another set of the input data $\mathbf{x}^{(n)} \in \Re^d$ is given and the LLGMN outputs the *a posteriori* probability $Y_k^{(n)}$ $(k = 1, \ldots, K)$. Next, the NF are trained using this output data and the teacher signal $T_k^{(n)}$ $(k = 1, \ldots, K)$ are given for each output unit. The learning of the NF is performed according to the learning rule based on the backpropagation through time [4] because of the presence of the interconnection in the second layer.

3) Effect of the Neural Filter on Classification Result: To examine the effect of the NF on the classification result, the following experiments are carried out. In the experiments, the 2-D input vector $(d = 2, H = 6, K = 2)$, shown in Table II, is used. The NF, which includes eight units in the second layer, is trained using $N = 168$ data series according to the pseudorandom series for 180 s. Then, the ratio of the correct classification to 422 data, which are not used in the learning, is computed.

TABLE V
EFFECT OF THE NF ON CLASSIFICATION RESULTS

Type of the network		Eye opening and closing		Artificial photic stimulation	
		LLGMN	LLGMN with NF	LLGMN	LLGMN with NF
subject A	R_{cla}	91.1	94.5	62.4	84.4
(male)	R_{sd}	0.4	0.3	4.4	1.4
subject B	R_{cla}	83.3	90.4	83.7	92.3
(male)	R_{sd}	0.6	0.8	3.4	1.1
subject C	R_{cla}	88.6	93.8	70.7	79.5
(male)	R_{sd}	1.4	1.1	1.2	0.5
subject D	R_{cla}	81.3	89.7	74.4	80.2
(male)	R_{sd}	1.1	0.5	2.1	0.2
subject E	R_{cla}	93.2	93.4	73.8	82.4
(male)	R_{sd}	0.6	0.1	2.5	0.9

R_{cla} : Classification rate(%), R_{sd} : Standard deviation(%)

Fig. 15 shows the effect of the NF on the classification result (subject A). In the figure, the timing of switching eye states (or artificial flash light), the input pattern of the LLGMN (x_1, x_2), the output of the LLGMN $({}^{(3)}O_1, {}^{(3)}O_2)$, the output of the NF $({}^{(4)}v_1, {}^{(4)}v_2)$, and the classification results are shown. The outputs of the LLGMN, especially in the case of the artificial photic stimulation, are varied considerably, depending on time, and accurate classification is not realized. The statistical processing by the LLGMN is not enough in this case. It can be seen from Fig. 15 that the NF makes the output of the LLGMN considerably smooth and the high classification performance is obtained. This is the effect of the NF on the classification results.

Table V shows classification results for five subjects. The mean values and the standard deviations of the classification rate for 30 kinds of randomly chosen initial weights are shown. In both cases, the classification rates tend to improve with the use of NF.

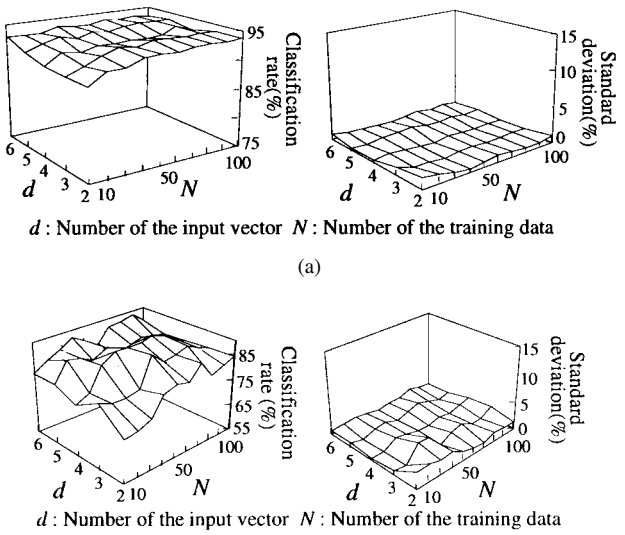


Fig. 16. Effects of the dimension of the input vector and number of the training data on the classification results.

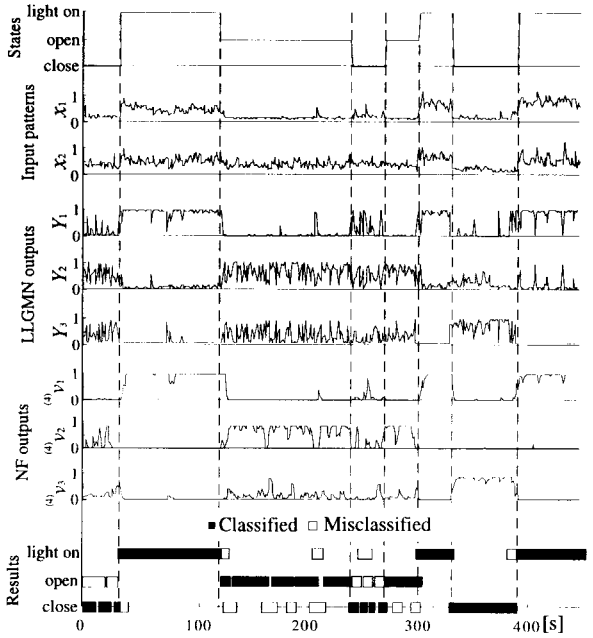


Fig. 17. Classification results of three types of the photic stimulation by LLGMN.

Then, we examine the changes of the classification rates depending on the number of the training data N and the dimension of the input vector d. Fig. 16 shows the classification result. Compared to the use of the LLGMN only (see Figs. 10 and 12), the classification rates improve considerably and the standard deviations of the classification rates keep very small values.

Finally, additional experiments for three classes of the EEG patterns are performed. Subjects are seated in a dark room and the following three states are used for the classification: closing eyes, opening eyes, and opening eyes with an artificial light. Experimental conditions are the same as the ones used in Figs. 8 and 11. Fig. 17 shows an example of the classification results. Although it seems to be considerably difficult to classify the EEG patterns into three different classes, the classification rate is about 80% in these experiments. The output of the LLGMN are smoothed out by the NF, taking the time-varying characteristics into consideration.

TABLE VI
CLASSIFICATION RESULTS OF THREE TYPES OF THE PHOTIC STIMULATIONS

Type of the network		LLGMN	LLGMN with NF
subject A (male)	R_{cla}	75.8	84.4
	R_{std}	0.5	0.1
subject B (male)	R_{cla}	83.8	91.6
	R_{std}	0.3	0.3
subject C (male)	R_{cla}	69.1	78.3
	R_{std}	4.5	2.3
subject D (male)	R_{cla}	65.8	74.6
	R_{std}	1.7	1.8
subject E (male)	R_{cla}	77.2	87.1
	R_{std}	1.9	0.9

R_{cla} : Classification rate(%)
R_{std} : Standard deviation(%)

Table VI shows the classification results for five subjects. Although the high classification performance is not realized for subjects C and D, the NF improves the classification rates for all subjects. In the future, the feature extraction method and experimental apparatus for EEG measurements should be reconsidered to improve the classification performance.

VI. CONCLUSION

The present paper has proposed a new NN based on the LLGMN that can estimate the *a posteriori* probability for pattern classification problems. The parameters of the network, such as an activation function of each unit, number of layers, and number of units, can be determined easily in correspondence to the GMM incorporated in the network. Also, the output from the LLGMN can be interpreted as a probability. The forward calculation and backward learning rule, which is based on the maximum likelihood estimation, can be defined in the same manner as the one of the feedforward NN model. To examine the classification ability of the proposed network, simulation and experiments have been performed. The results obtained are summarized as follows.

- In the simulation, the statistical structure incorporated in the LLGMN realized the smooth decision region boundaries even for a small sample size of the training data.
- Weight coefficients used in the LLGMN have no constraints and are mutually independent, so that the LLGMN achieved higher classification performance than that of the MLANS even for a small sample size of the training data.
- In the EEG pattern classification experiments of eye states and artificial photic stimulation for five subjects, the LLGMN classified the EEG patterns with about 85% and 75% of the classification rates, respectively.
- LLGMN achieved effective learning and relatively high classification performance, while the learning of the BPN converged local minima frequently and that of the MLANS needed a large sample size of the training data.

- In order to cope with time-varying characteristics of the EEG patterns, a new network structure that combines LLGMN with NF was introduced in three kinds of experiments: the eye opening and closing, the artificial flash light, and the eye opening and closing with an artificial flash light. This network achieved considerably high classification performance with 92.4, 83.8, and 83.2% of the mean values of the classification rates, respectively.

Our future research will be directed toward developing some techniques to incorporate a dynamic statistical model into the NN.

REFERENCES

[1] F. Y. Wu, J. D. Slater, L. S. Honih, and R. E. Ramsay, "A neural network for event-related potential diagnosis," *Comput. Biol. Machine*, vol. 23, no. 3, pp. 251–264, 1993.

[2] N. Schaltenbrand, R. Lengelle, and J. P. Macher, "Neural network model: Application to automatic analysis of human sleep," *Comput. Biomed. Res.*, vol. 26, no. 2, pp. 157–171, 1993.

[3] M. Peltoranta and G. Pfurtscheller, "Neural network based classification of nonaveraged event-related EEG responses," *Med. Biol. Eng. Comput.*, vol. 32, no. 2, pp. 189–196, 1994.

[4] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, vol. 1. Cambridge, MA: MIT Press, 1986, chap. 8.

[5] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," *Neurocomputing: Algorithms, Architectures, and Applications*, S. F. Fogelman and J. Hault, Eds. New York: Springer-Verlag, 1989, pp. 227–236.

[6] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Networks*, vol. 2, pp. 568–576, Nov. 1991.

[7] S. Nakagawa and Y. Ono, "Estimation of probability density function and a posteriori probability by neural networks, and vowel recognition," *Trans. IEICE Jpn.*, vol. J76-D-II, no. 6, pp. 1081–1089, 1993 (in Japanese).

[8] H. G. C. Tråvén, "A neural network approach to statistical pattern classification by "semiparametric" estimation of probability density functions," *IEEE Trans. Neural Networks*, vol. 2, pp. 366–377, May 1991.

[9] L. I. Perlovsky and M. M. McManus, "Maximum likelihood neural networks for sensor fusion and adaptive classification," *Neural Networks*, vol. 4, pp. 89–102, 1991.

[10] T. Tsuji, D. Mori, and T. Ito, "Motion discrimination method from EMG signals using statistically structured neural networks," *Trans. IEE Jpn.*, vol. 112-C, no. 8, pp. 465–473, 1992 (in Japanese).

[11] S. Lee and S. Shimoji, "Self-organization of Gaussian mixture model for learning class pdfs in pattern classification," in *Proc. IEEE Int. Joint Conf. Neural Networks 1993*, vol. III, pp. 2492–2495.

[12] R. L. Streit and T. E. Luginbuhl, "Maximum likelihood training of probabilistic neural networks," *IEEE Trans. Neural Networks*, vol. 5, pp. 764–783, Sept. 1994.

[13] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," in *Proc. IEEE Int. Joint Conf. Neural Networks 1993*, vol. II, pp. 1339–1344.

[14] J. A. Nelder and W. M. Wedderburn, "Generalized linear models," *J. R. Stat. Soc. A*, vol. 135, part 3, pp. 370–384, 1972.

[15] J. M. Bernardo and A. F. M. Smith, *BAYESIAN THEORY*. New York: Wiley, 1994, pp. 75–76.

[16] O. Fukuda, T. Tsuji, and M. Kaneko, "Pattern classification of EEG signals using a log-linearized Gaussian mixture neural networks," in *Proc. IEEE Int. Conf. Neural Networks 1995*, vol. V, pp. 2479–2484.

[17] J. H. Wolfe, "Pattern clustering by multivariate mixture analysis," *Multivariate Behav. Res.*, vol. 5, pp. 329–350, 1970.

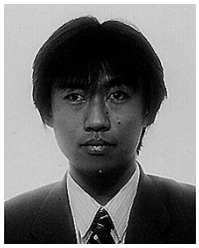
[18] M. Zak, "Terminal attractors for addressable memory in neural networks," *Phys. Lett. A*, vol. 133, pp. 218–222, 1988.

[19] J. T. H. Lo, "Synthetic approach to optimal filtering," *IEEE Trans. Neural Networks*, vol. 5, pp. 803–811, Sept. 1994.

Toshio Tsuji (A'88) was born in Kyoto, Japan, on December 25, 1959. He received the B.E. degree in industrial engineering in 1982, the M.E. degree in systems engineering in 1985, and the Dr.Eng. degree in systems engineering in 1989, all from Hiroshima University, Higashi-Hiroshima, Japan.

He was a Research Associate in Faculty of Engineering, Hiroshima University, from 1985 to 1994, and he was a Visiting Researcher at the University of Genova, Genova, Italy, from 1992 to 1993. He is currently an Associate Professor in the Department of Industrial and Systems Engineering, Hiroshima University. He has been interested in various aspects of motor control in robot and human movements. His current research interests include control of EMG-controlled prostheses and computational neural sciences, in particular, biological motor control.

Dr. Tsuji is a member of the Japanese Society of Mechanical Engineers, Robotics Society of Japan, and Japanese Society of Instrumentation and Control Engineers.



Osamu Fukuda was born in Fukuoka, Japan, on September 30, 1969. He received the B.E. degree in mechanical engineering from Kyushu Institute of Technology, Kyushu, Japan, in 1993 and the M.E. degree in information engineering from Hiroshima University, Higashi-Hiroshima, Japan, in 1997. He is currently pursuing the Ph.D. degree in information engineering at Hiroshima University.

His main research interest includes human interface and the neural network.

Mr. Fukuda is a Research Fellow of the Japanese Society for the Promotion of Science.

Hiroyuki Ichinobe was born on January 17, 1970. He received the B.E. degree in systems engineering in 1992 and the M.E. degree in information engineering in 1994 from Hiroshima University, Higashi-Hiroshima, Japan, where he studied the neural network and its application to the EMG-controlled artificial hand.

He is currently with the Customer Equipment Department, NIPPON Telegraph and Telephone Corporation, Chiyoda, Tokyo, Japan.

Makoto Kaneko (A'84–M'87) received the B.S. degree in mechanical engineering from Kyushu Institute of Technology, Kyushu, Japan, in 1976, and the M.S. and Ph.D. degrees in mechanical engineering from Tokyo University, Tokyo, Japan, in 1978 and 1981, respectively.

He was a Researcher at the Mechanical Engineering Laboratory (MEL), Ministry of International Trade and Industry (MITI), Tsukuba Science City, Tsukuba, Japan. from 1981 to 1990. From 1988 to 1989, he was a Postdoctoral Fellow at the Technical University of Darmstadt, Darmstadt, Germany, where he joined a space robotics project. From 1990 to 1993, he was an Associate Professor of computer science and system engineering at Kyushu Institute of Technology. From November 1991 to January 1992, he received an Invited Professorship at Technical University of Darmstadt. Since October 1993, he has been a Professor in the Industrial Engineering Department, Hiroshima University, Higashi-Hiroshima, Japan. His research interests include tactile-based active sensing, grasping strategy, sensor applications, and experimental robotics.

Dr. Kaneko received the Outstanding Young Engineer Award in 1983 from the Japanese Society of Mechanical Engineers, the Best Paper awards from the Robotics Society of Japan in 1994 and the Japanese Society of Instrumentation and Control Engineers in 1996. He also received the Humboldt Research Award in 1997. He served as an Associate Editor of IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION from 1990 to 1994. He has been a program committee member for the IEEE International Conference on Intelligent Robots and Systems since 1991. He has also worked as a program committee member for the 1995, 1996, and 1998 IEEE International Conference on Robotics and Automation. He is a member of the IEEE Robotics and Automation Society, the IEEE Systems, Man, and Cybernetics Society, and the IEEE Industrial Electronics Society. He is also a member of the Japanese Society of Mechanical Engineers, Robotics Society of Japan, and Japanese Society of Instrumentation and Control Engineers.