35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

Bioelectric Signal Classification Using a Recurrent Probabilistic Neural Network with Time-series Discriminant Component Analysis

Hideaki HAYASHI, *Student Member, IEEE,* Keisuke SHIMA, *Member, IEEE,*
Taro SHIBANOKI, *Member, IEEE,* Yuichi KURITA, *Member, IEEE,* and Toshio TSUJI, *Member, IEEE*

***Abstract*— This paper outlines a probabilistic neural network developed on the basis of time-series discriminant component analysis (TSDCA) that can be used to classify high-dimensional time-series patterns. TSDCA involves the compression of high-dimensional time series into a lower-dimensional space using a set of orthogonal transformations and the calculation of posterior probabilities based on a continuous-density hidden Markov model that incorporates a Gaussian mixture model expressed in the reduced-dimensional space. The analysis can be incorporated into a neural network so that parameters can be obtained appropriately as network coefficients according to backpropagation-through-time-based training algorithm. The network is considered to enable high-accuracy classification of high-dimensional time-series patterns and to reduce the computation time taken for network training. In the experiments conducted during the study, the validity of the proposed network was demonstrated for EEG signals.**

I. Introduction

Bioelectric signals such as electromyograms (EMGs) and electroencephalograms (EEGs) strongly reflect human internal states and intentions, and many studies have investigated interfaces controlled with these signals [1], [2], [3]. To support the development of high-performance interfaces, accurate pattern classification of bioelectric signals is required.

In previous work, neural networks (NNs) have been applied to bioelectric signal pattern classification. In particular, Tsuji *et al*. proposed NNs based on a Gaussian mixture model (GMM) and a hidden Markov model (HMM), and demonstrated their effectiveness for such classification [4], [5].

However, the dimensionality of input with NNs increases if high-dimensional features (e.g., signals measured with numerous electrodes and frequency spectra) are used. Such increased dimensionality results in network structural complexity, parameter learning difficulty and longer computation time.

This paper outlines a novel time-series pattern classification model called time-series discriminant component analysis (TSDCA) that can be used to reduce and classify input data in consideration of the time context. A probabilistic neural network has also been developed based on TSDCA, which allows a reduction in the amount of input data required based on the use of several orthogonal transformation matrices and enables calculation of posterior probabilities for classification under the assumption that the reduced feature vectors obey an HMM with a GMM for probabilistic density. In this way, the parameters of dimensional reduction and classification can be obtained together, thereby supporting the accurate classification of time-series data with high dimensionality.

H. Hayashi, T. Shibanoki, Y. Kurita, and T. Tsuji are with Graduate School of Engineering, Hiroshima University, Higashi-hiroshima, 739-8527 Japan hayashi@bsys.hiroshima-u.ac.jp

K. Shima is with Graduate School of Engineering, Yokohama National University, Yokohama, 240-8501 Japan

II. Time-series discriminant component analysis (TSDCA)

A. Model structure

TSDCA consists of several orthogonal transformation matrices and an HMM that incorporates a GMM for the probabilistic density function. The model allows a reduction in the dimensionality of input data and enables calculation of posterior probabilities for each class.

In regard to classifying a d-dimensional time-series vector $\boldsymbol{x}(t) \in \Re^d$ into one of the given C classes, the posterior probability $P(c|\boldsymbol{x}(t))$ $(c = 1, \cdots, C)$ is examined. First, $\boldsymbol{x}(t)$ is projected into d'-dimensional vector $\boldsymbol{x'}^{(c,k,m)}(t) \in \Re^{d'}$ using several orthogonal transformation matrices $\mathbf{V}^{(c,k,m)}$. This can be described as follows:

$$\boldsymbol{x'}^{(c,k,m)}(t) = \mathbf{V}^{(c,k,m)^{\mathrm{T}}}(\boldsymbol{x}(t) - \boldsymbol{\mu}^{(c,k,m)}), \quad (1)$$

where $\boldsymbol{\mu}^{(c,k,m)} \in \Re^d$ is the mean vector of the component $\{c, k, m\}$ $(k = 1, \cdots, K_c$; K_c is the number of states, $m = 1, \cdots, M_{c,k}$; $M_{c,k}$ is the number of components), and $\mathbf{V}^{(c,k,m)} \in \Re^{d \times d'}$ is the orthogonal transformation matrix that projects from d into d'.

In the compressed feature space, the projected data obey a probabilistic density function as follows:

$$g(\boldsymbol{x}(t); c,k,m) = (2\pi)^{-\frac{d'}{2}} |\Sigma'^{(c,k,m)}|^{-\frac{1}{2}} \exp[\psi^{(c,k,m)}(t)], \quad (2)$$

$$\psi^{(c,k,m)}(t) = -\frac{1}{2}\boldsymbol{x'}^{(c,k,m)}(t)^{\mathrm{T}}(\Sigma'^{(c,k,m)})^{-1}\boldsymbol{x'}^{(c,k,m)}(t), \quad (3)$$

where $\Sigma'^{(c,k,m)} \in \Re^{d' \times d'}$ is the covariance matrix in the compressed feature space.

Assuming that the projected data obey an HMM, the posterior probability of $\boldsymbol{x}(t)$ is calculated as

$$P(c|\boldsymbol{x}(t)) = \sum_{k=1}^{K_c} \frac{\alpha_k^c(t)}{\sum_{c'=1}^{C}\sum_{k'=1}^{K_{c'}} \alpha_{k'}^{c'}(t)}, \quad (4)$$

$$\alpha_k^c(1) = \pi_k^c b_k^c(\boldsymbol{x}(1)), \quad (5)$$

$$\alpha_k^c(t) = \sum_{k'=1}^{K_c} \alpha_{k'}^c(t-1)\gamma_{k',k}^c b_k^c(\boldsymbol{x}(t)), \quad (1 < t \leq T) \quad (6)$$

where $\gamma_{k',k}^c$ is the probability of a state change from k' to k in class c, $b_k^c(\boldsymbol{x}(t))$ is defined as the posterior probability

for the state k in class c corresponding to $\boldsymbol{x}(t)$, and the prior probability π_k^c is equal to $P(c,k)|_{t=0}$. Here, $\gamma_{k',k}^c b_k^c(\boldsymbol{x}(t))$ can be derived with the form

$$\gamma_{k',k}^c b_k^c(\boldsymbol{x}(t)) = \sum_{m=1}^{M_{c,k}} \gamma_{k',k}^c r_{c,k,m} g(\boldsymbol{x}(t); c, k, m) \quad (7)$$

where $r_{c,k,m}$ represents the mixture proportion.

B. Log-linearization

The parameters of TSDCA need to be empirically determined based on specific definitions. In particular, a large quantity of training data is needed to train an HMM. Tsuji *et al.* showed that the parameters of an HMM can be expressed with a smaller number of coefficients using log-linearization as a way of addressing this problem [5].

In this paper, (1) and (7) are considered based on a linear combination of coefficient matrices and input vectors. First, $\boldsymbol{x'}^{(c,k,m)}(t)$ is transformed as follows:

$$\begin{aligned}
&\boldsymbol{x'}^{(c,k,m)}(t) \\
&= \mathbf{V}^{(c,k,m)^{\mathrm{T}}}(\boldsymbol{x}(t) - \boldsymbol{\mu}^{(c,k,m)}) \\
&\triangleq \mathbf{V}^{(c,k,m)^{\mathrm{T}}}\boldsymbol{x}(t) - \hat{\boldsymbol{\mu}}^{(c,k,m)} \\
&= \begin{bmatrix} -\hat{\mu}_1^{(c,k,m)} & V_{1,1}^{(c,k,m)} & \cdots & V_{1,d}^{(c,k,m)} \\ \vdots & \vdots & \ddots & \vdots \\ -\hat{\mu}_{d'}^{(c,k,m)} & V_{d',1}^{(c,k,m)} & \cdots & V_{d',d}^{(c,k,m)} \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{x}(t) \end{bmatrix} \\
&\triangleq {}^{(1)}\mathbf{W}^{(c,k,m)}\mathbf{X}(t), \quad (8)
\end{aligned}$$

where $\hat{\boldsymbol{\mu}}^{(c,k,m)} \in \Re^{d'}$ corresponds to an image of the mean vector mapped onto the compressed space from the input space. Hence, $\boldsymbol{x'}^{(c,k,m)}(t)$ is expressed by multiplication of the coefficient matrix ${}^{(1)}\mathbf{W}^{(c,k,m)}$ and the novel input vector $\mathbf{X}(t) = [1, \boldsymbol{x}(t)^{\mathrm{T}}]^{\mathrm{T}} \in \Re^{d+1}$. Secondly, setting

$$\xi_{k',k,m}^c(t) = \gamma_{k',k}^c r_{c,k,m} g(\boldsymbol{x}(t); c, k, m) \quad (9)$$

and taking the log-linearization of $\xi_{k',k,m}^c(t)$ gives

$$\begin{aligned}
&\log \xi_{k',k,m}^c(t) \\
&= [\log \gamma_{k',k}^c + \log r_{c,k,m} - \frac{d'}{2}\log 2\pi - \frac{1}{2}\log|\Sigma'^{(c,k,m)}|, \\
&\quad -\frac{1}{2}s'^{(c,k,m)}_{1,1}, -s'^{(c,k,m)}_{1,2}, \cdots, s'^{(c,k,m)}_{1,d'}, \cdots, \\
&\quad -\frac{1}{2}(2-\delta_{i,j})s'^{(c,k,m)}_{i,j}\cdots, -\frac{1}{2}s'^{(c,k,m)}_{d',d'}]\mathbf{X'}^{(c,k,m)}(t) \\
&\triangleq {}^{(2)}\mathbf{W}^{(c,k',k,m)}\mathbf{X'}^{(c,k,m)}(t), \quad (10)
\end{aligned}$$

where $s'^{(c,k,m)}_{1,1}, \cdots, s'^{(c,k,m)}_{d',d'}$ are elements of the inverse matrix $(\Sigma'^{(c,k,m)})^{-1}$, and $\delta_{i,j}$ is a Kronecker delta, which is 1 if $i = j$ and otherwise 0. Additionally, $\mathbf{X'}^{(c,k,m)}(t) \in \Re^H (H = 1 + \frac{d'(d'+1)}{2})$ is defined as

$$\begin{aligned}
&\mathbf{X'}^{(c,k,m)}(t) \\
&= [1, x_1'^{(c,k,m)}(t)^2, x_1'^{(c,k,m)}(t)x_2'^{(c,k,m)}(t), \cdots, \\
&\quad x_1'^{(c,k,m)}(t)x_{d'}'^{(c,k,m)}(t), x_2'^{(c,k,m)}(t)^2, \\
&\quad x_2'^{(c,k,m)}(t)x_3'^{(c,k,m)}(t), \cdots, \\
&\quad x_2'^{(c,k,m)}(t)x_{d'}'^{(c,k,m)}(t), \cdots, x_{d'}'^{(c,k,m)}(t)^2]. \quad (11)
\end{aligned}$$

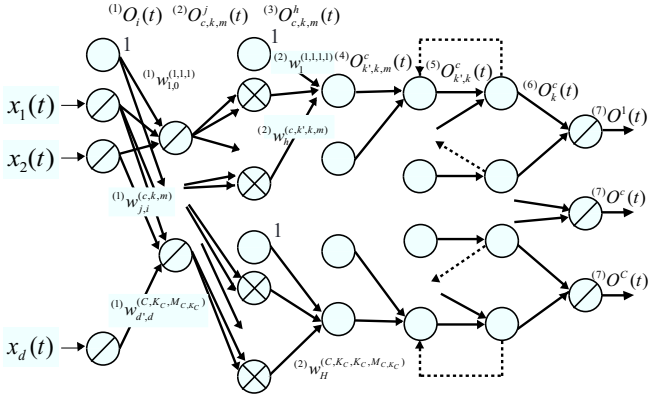


Fig. 1. Structure of the proposed neural network

As stated above, the parameters of TSDCA can be expressed with a smaller number of coefficients ${}^{(1)}\mathbf{W}^{(c,k,m)}$ and ${}^{(2)}\mathbf{W}^{(c,k',k,m)}$ using log-linearization. If these coefficients are appropriately obtained, the parameters and the structure of the model can be defined and the posterior probability of high-dimensional time-series data for each class can be calculated.

The next section describes how ${}^{(1)}\mathbf{W}^{(c,k,m)}$ and ${}^{(2)}\mathbf{W}^{(c,k',k,m)}$ are acquired as weight coefficients of a NN through learning.

III. Proposed Neural Network

A. Network structure

Figure 1 shows the structure of the proposed NN, which is a seven-layer recurrent type with the weight coefficients ${}^{(1)}\mathbf{W}^{(c,k,m)}$ and ${}^{(2)}\mathbf{W}^{(c,k',k,m)}$ between the first/second and third/fourth layers, respectively, and a feedback connection between the fifth and sixth layers.

The first layer consists of $d+1$ units corresponding to the dimensions of the input data $\boldsymbol{x}(t)$ $(t = 1, 2, \cdots, T) \in \Re^d$. The relationships between the input and the output are defined as

$$\begin{aligned}
{}^{(1)}I_i(t) &= \begin{cases} 1 & (i=0) \\ x_i(t) & (i = 1, \cdots, d) \end{cases}, \quad (12) \\
{}^{(1)}O_i(t) &= {}^{(1)}I_i(t), \quad (13)
\end{aligned}$$

where ${}^{(1)}I_i(t)$ and ${}^{(1)}O_i(t)$ are the input and output of the ith unit, respectively.

The second layer is composed of $C \times K_c \times M_{c,k} \times d'$ units, each receiving the output of the first layer weighted by the coefficient ${}^{(1)}w^{(c,k,m)}_{j,i}$. The relationships between the input ${}^{(2)}I^j_{c,k,m}(t)$ and the output ${}^{(2)}O^j_{c,k,m}(t)$ of the unit $\{j, c, k, m\}$ $(j = 1, \cdots, d',\ c = 1, \cdots, C,\ k = 1, \cdots, K_c,\ m = 1, \cdots, M_{c,k})$ are described as

$$\begin{aligned}
{}^{(2)}I^j_{c,k,m}(t) &= \sum_{i=0}^{d} {}^{(1)}O_i(t)\,{}^{(1)}w^{(c,k,m)}_{j,i}, \quad (14) \\
{}^{(2)}O^j_{c,k,m}(t) &= {}^{(2)}I^j_{c,k,m}(t), \quad (15)
\end{aligned}$$

where the weight coefficient ${}^{(1)}w_{j,i}^{(c,k,m)}$ is for each element of the matrix ${}^{(1)}\mathbf{W}^{(c,k,m)}$ described as follows:

$${}^{(1)}\mathbf{W}^{(c,k,m)} = \begin{bmatrix} {}^{(1)}w_{1,0}^{(c,k,m)} & \cdots & {}^{(1)}w_{1,d}^{(c,k,m)} \\ \vdots & \ddots & \vdots \\ {}^{(1)}w_{d',0}^{(c,k,m)} & \cdots & {}^{(1)}w_{d',d}^{(c,k,m)} \end{bmatrix}. \quad (16)$$

The third layer is comprised of $C \times K_c \times M_{c,k} \times H$ ($H = 1 + \frac{d'(d'+1)}{2}$) units. The relationships between the input ${}^{(3)}I_{c,k,m}^{h}(t)$ and the output ${}^{(3)}O_{c,k,m}^{h}(t)$ of the units $\{h, c, k, m\}$ $(h = 1, \cdots, H)$ are defined as

$${}^{(3)}I_{c,k,m}^{h}(t) = \begin{cases} 1 \ \ (h=1) \\ {}^{(2)}O_{c,k,m}^{j}(t)\,{}^{(2)}O_{c,k,m}^{j'}(t) \\ \quad (h = j' - \frac{1}{2}j^2 + (d' + \frac{1}{2})j - d' + 1) \end{cases}, \quad (17)$$

$${}^{(3)}O_{c,k,m}^{h}(t) = {}^{(3)}I_{c,k,m}^{h}(t), \quad (18)$$

where $j \leq j'(j' = 1, \cdots, d')$, and (17) corresponds to the nonlinear conversion shown in (11).

The fourth layer is comprised of $C \times K_c^2 \times M_{c,k}$ units. Unit $\{c, k', k, m\}(k' = 1, \cdots, K_c)$ receives the output of the third layer weighted by the coefficient ${}^{(2)}w_{h}^{(c,k',k,m)}$. The input ${}^{(4)}I_{k',k,m}^{c}(t)$ and the output ${}^{(4)}O_{k',k,m}^{c}(t)$ are defined as

$${}^{(4)}I_{k',k,m}^{c}(t) = \sum_{h=1}^{H} {}^{(3)}O_{c,k,m}^{h}(t)\,{}^{(2)}w_{h}^{(c,k',k,m)}, \quad (19)$$

$${}^{(4)}O_{k',k,m}^{c}(t) = \exp\left({}^{(4)}I_{k',k,m}^{c}(t)\right), \quad (20)$$

where the weight coefficient ${}^{(2)}w_{h}^{(c,k',k,m)}$ corresponds to each element of the vector ${}^{(2)}\mathbf{W}^{(c,k',k,m)}$.

$${}^{(2)}\mathbf{W}^{(c,k',k,m)} = \left[{}^{(2)}w_{1}^{(c,k',k,m)}, \cdots, {}^{(2)}w_{H}^{(c,k',k,m)}\right] \quad (21)$$

The fifth layer consists of $C \times K_c^2$ units. The output of the fourth layer is added up and input into this layer. The one-time-prior output of the sixth layer is also fed back to the fifth layer. These are expressed as follows:

$${}^{(5)}I_{k',k}^{c}(t) = \sum_{m=1}^{M_{c,k}} {}^{(4)}O_{k',k,m}^{c}(t), \quad (22)$$

$${}^{(5)}O_{k',k}^{c}(t) = {}^{(6)}O_{k'}^{c}(t-1)\,{}^{(5)}I_{k',k}^{c}(t), \quad (23)$$

where ${}^{(6)}O_{k'}^{c}(0) = 1.0$ for the initial phase.

The sixth layer has $C \times K_c$ units. The relationships between the input ${}^{(6)}I_{k}^{c}(t)$ and the output ${}^{(6)}O_{k}^{c}(t)$ of the unit $\{c, k\}$ are described as

$${}^{(6)}I_{k}^{c}(t) = \sum_{k'=1}^{K_c} {}^{(5)}O_{k',k}^{c}(t), \quad (24)$$

$${}^{(6)}O_{k}^{c}(t) = \frac{{}^{(6)}I_{k}^{c}(t)}{\sum_{c'=1}^{C}\sum_{k'=1}^{K_{c'}} {}^{(6)}I_{k'}^{c'}(t)}. \quad (25)$$

Finally, the seventh layer consists of C units, and its input ${}^{(7)}I^{c}(t)$ and output ${}^{(7)}O^{c}(t)$ are

$${}^{(7)}I^{c}(t) = \sum_{k=1}^{K_c} {}^{(6)}O_{k}^{c}(t), \quad (26)$$

$${}^{(7)}O^{c}(t) = {}^{(7)}I^{c}(t). \quad (27)$$

${}^{(7)}O^{c}(t)$ corresponds to the posterior probability for class c $P(c|\boldsymbol{x}(t))$. Here, the posterior probability $P(c|\boldsymbol{x}(t))$ based on TSDCA can be calculated if the NN coefficients ${}^{(1)}\mathbf{W}^{(c,k,m)}$, ${}^{(2)}\mathbf{W}^{(c,k',k,m)}$ are appropriately established.

B. Learning algorithm

A set of vector streams $\boldsymbol{x}^{(n)}(t)$ is given for training with the teacher vector $\boldsymbol{Q}^{(n)} = [Q_1^{(n)}, \cdots, Q_c^{(n)}, \cdots, Q_C^{(n)}]^{\mathrm{T}}$ $(n = 1, \cdots, N)$ for the nth input at T. The training process of the proposed NN involves minimization of an energy function J defined as

$$J = \sum_{n=1}^{N} J_n = -\sum_{n=1}^{N}\sum_{c=1}^{C} Q_c^{(n)} \log {}^{(7)}O^{c}(T)^{(n)}, \quad (28)$$

to maximize the log-likelihood. Here, ${}^{(7)}O^{c}(T)^{(n)}$ is the output for an input vector at T. The weight modification for ${}^{(1)}w_{j,i}^{(c,k,m)}$ and ${}^{(2)}w_{h}^{(c,k',k,m)}$ based on the gradient method is defined as

$$\Delta^{(1)}w_{j,i}^{(c,k,m)} = -\gamma \sum_{n=1}^{N} \frac{\partial J_n}{\partial {}^{(1)}w_{j,i}^{(c,k,m)}}, \quad (29)$$

$$\Delta^{(2)}w_{h}^{(c,k',k,m)} = -\gamma \sum_{n=1}^{N} \frac{\partial J_n}{\partial {}^{(2)}w_{h}^{(c,k',k,m)}}, \quad (30)$$

where γ is the learning rate.

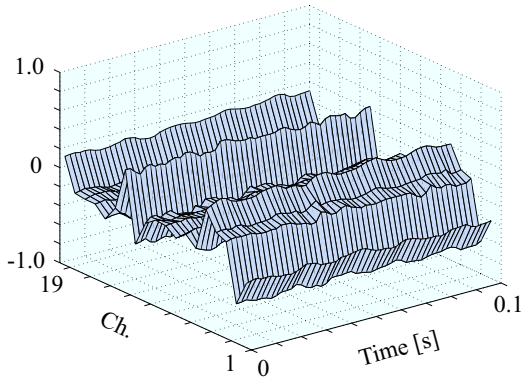
The backpropagation-through-time (BPTT) algorithm [6] is used for the weight modification. Based on this, the error gradient within a stream is accumulated and the weight modification is calculated.

To maintain orthogonality, orthonormalization using the Gram-Schmidt process is applied to $\mathbf{V}^{(c,k,m)}$ in ${}^{(1)}\mathbf{W}^{(c,k,m)}$ every time the weight coefficient is modulated. Using the above algorithm, collective training is applied in relation to the weight coefficients for dimensional reduction and discrimination.

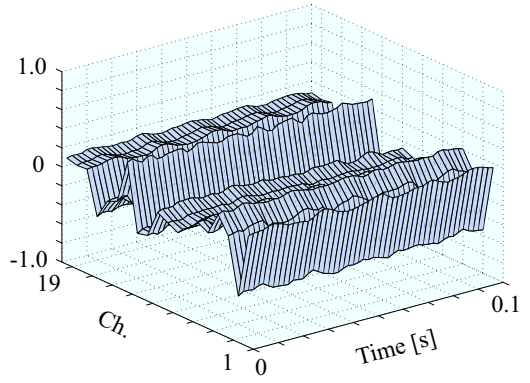
IV. Experiment

A. Method

To evaluate the validity of the proposed NN for real biological data, a classification experiment was conducted using 19-channel ($d = 19$) EEG data downloaded from Project BCI [7] (see Fig. 2 for an example). The EEG signals were recorded from a healthy subject who performed two tasks ($C = 2$: (a) right-hand movement, (b) left-hand movement) for about 128.6 seconds each. In the experiment, with 0.1 seconds of data as a sample for each class, 50 samples were treated as training data, and the remaining 1,236 were used as test data. Average classification rates were then calculated by changing the combination of training/test data sets randomly 10 times and resetting the initial weight coefficients 10 times for each combination. The parameters of the proposed NN were $K_c = 1$, $M_{c,k} = 1$, $d' = 1$. The average classification rate was also compared with those of the R-LLGMN [5], a method developed by combining PCA and the R-LLGMN

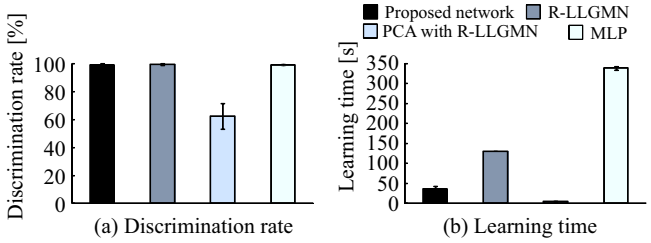


(a) Class 1: right-hand movement



(b) Class 2: left-hand movement

Fig. 2. Examples of EEG signals



(a) Discrimination rate (b) Learning time

Fig. 3. Average discrimination rate and learning time for each method

(PCA with the R-LLGMN), and a multi-layered perceptron (MLP). The number of states and the number of components of the R-LLGMN were identical to those of the proposed NN, and the number of compression dimensions with PCA was 1. The MLP had 5 layers (3 hidden), and the number of units in the input layer, the hidden layer and the output layer were 19, 20 and 2, respectively.

B. Results and discussion

Figure 3 shows the average classification rate and learning time for each method. It can be seen that the proposed NN, the R-LLGMN and the MLP achieved high classification rates (99.2 ± 1.0 [%], 99.6 ± 0.6 [%] and 99.3 ± 1.0 [%], respectively; see Fig. 3 (a)).

Meanwhile, the learning times for the R-LLGMN and the MLP were long (130.4 ± 0.1 [s] and 339.0 ± 4.3 [s], respectively; see Fig. 3 (b)) because all dimensions were used for classification with these NNs. Learning time was significantly reduced for the PCA with the R-LLGMN (4.3 ± 0.4 [s]), although its classification rate was low (62.6 ± 9.3 [%]) because the features necessary for classification could not be extracted with this combination. In contrast, the proposed NN achieved higher-level classification performance and a shorter learning time than conventional NNs (36.3 ± 6.6 [s]). This is presumably because compression and classification can be realized simultaneously with the proposed NN, and training can be implemented to achieve mapping to the lower dimensional space where classification is effective. These results demonstrate the validity of the proposed NN for EEG classification.

V. Conclusion

This paper outlines a novel time-series classification model called time-series discriminant component analysis (TSDCA) and a recurrent probabilistic NN with dimensional reduction based on TSDCA. This analysis involves several orthogonal transformation matrices and an HMM that includes a GMM for probabilistic density, thereby allowing dimensional reduction of input data and calculation of posterior probabilities for each class. TSDCA is incorporated into a NN structure using log-linearization so that the parameters are obtained as weight coefficients of the NN.

High-dimensional EEG classification experiments showed that the proposed NN demonstrated a high level of classification performance and relatively fast learning (average classification rate: 99.2 ± 1.0 [%], average learning time: 36.3 ± 6.6 [s]).

In future research, the authors plan to apply this approach to brain-computer interfaces and other interfaces involving the use of biosignals. Theoretical analysis of the proposed NN and improvement of the learning algorithm will also be conducted.

References

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, T. M. Vaughan, Brain-computer interfaces for communication and control, Clinical Neurophysiology, Vol. 113, pp. 767-791, 2002

[2] J. N. Mal, D. J. McFarland, T. M. Vaughan, L. M. McCane, P. Z. Tsui, D. J. Zeitlin, E. W. Sellers and J. R. Wolpaw, EEG correlates of P300-based brain-computer interface (BCI) performance in people with amyotrophic lateral sclerosis, Journal of Neural Engineering, Vol. 9, No. 2, 2012

[3] F. Zaccone, S. Micera, M. C. Carrozza, On the Shared Control of an EMG-Controlled Prosthetic Hand: Analysis of User-Prosthesis Interaction, IEEE Transactions on Robotics, Vol. 24, No. 1, pp. 170-184, 2008

[4] T. Tsuji, O. Fukuda, H. Ichinobe, and M. Kaneko: A Log-Linearized Gaussian Mixture Network and Its Application to EEG Pattern Classification, IEEE Transactions on Systems, Man, and Cybernetics-Part C, Vol. 29, No. 1, pp. 60-72, 1999

[5] T. Tsuji, N. Bu, M. Kaneko and O. Fukuda: A Recurrent Log-linearized Gaussian Mixture Network, IEEE Transactions on Neural Networks, Vol. 14, No. 2, pp. 304-316, 2003

[6] P. J. Werbos: Backpropagation through time: What it does and how to do it, Proceedings of the IEEE, Vol. 78, No. 10, pp. 1550-1560, 1990

[7] A. Midhat. Project BCI - EEG motor activity data set [Online]. Available: https://sites.google.com/site/projectbci/