計測自動制御学会論文集
Vol.50, No.4, 356/365 (2014)

時系列判別成分分析に基づく次元圧縮型リカレント確率ニューラルネット

早志英朗*・島　圭介**・芝軒太郎*
栗田雄一*・辻　敏夫*

A Reduced-dimensional Recurrent Probabilistic Neural Network Based on Time-series Discriminant Component Analysis

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

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 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, the validity of the network was demonstrated for high-dimensional artificial data. The results showed that the network can achieve high discrimination performance in a relatively short learning time. The applicability to biosignal classification was also demonstrated through an EEG discrimination experiment.

Key Words: neural network, dimensionality reduction, pattern classification, hidden Markov model, Gaussian mixture model

1. はじめに

Rumelhartら[1]によって提案された誤差逆伝播型ニューラルネットワーク（Neural Networks: 以下，NNと略記）は高度な並列処理が可能であり，任意の非線形写像を獲得できる強力な学習能力を有するため，さまざまなパターン識別問題に応用されてきた[2)~6)]．たとえばKellyら[2]やHaselsteinerら[3]は誤差逆伝播型NNを用いた筋電位（Electromyogram: 以下，EMGと略記）や脳波（Electroencephalogram: 以下，EEGと略記）の識別を試みている．

EMGやEEGなどの生体信号のパターン識別問題に用いられる入力データはある確率分布に従う確率変数であるため，パターン識別問題は通常，入力データが従う確率密度関数の同定問題に帰着される．確率分布の推定には，あらかじめ特定の分布を仮定してそのパラメータを推定する方法が一般的であり，近年では分布のモデルパラメータをニューラルネットなどの学習器に展開することで，データが従う確率密度関数を学習的に推定する試みがなされている．特にTsujiらは混合正規分布モデル（Gaussian Mixture Model: 以下，GMMと略記）や隠れマルコフモデル（Hidden Markov Model: 以下，HMMと略記）を導入したNNを提案し，EMGやEEGなどの生体信号を高い精度で識別できることを示している[5),6)]．

生体信号の識別問題において，多動作の高精度な識別の実現には生体から識別に有効な情報を取得する必要があるため，信号の計測に多数の電極を用いたり周波数スペクトルなど高次元の特徴量を抽出して識別に用いる方法が試みられてきた[7)~9)]．しかしながら，入力が高次元化すれば学習器の構造が複雑になってモデルパラメータの推定が困難となるだけでなく，学習に莫大な時間が必要となる問題がある．

これに対し，入力データの有する情報を保存したまま入力次元を圧縮して識別する方法が提案されている．たとえば，Gülerら[10]はEMG信号の周波数情報に対し主成分分析（Principle Component Analysis: 以下，PCAと略記）を用いて低次元に圧縮し，多層パーセプトロンやサポートベクターマシンを用いた識別を可能とした．また，Lawrenceら[11]は自己組織化マップによる次元圧縮とNNによる識別を組み合わせ，顔画像認識へ応用した．しかしながら，これらの手法は次元圧

* 広島大学大学院工学研究科　東広島市鏡山 1-4-1
** 横浜国立大学大学院工学研究院
横浜市保土ヶ谷区常盤台 79-5
* Graduate School of Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima
** Faculty of Engineering, Yokohama National University, 79-5 Tokiwadai Hodogaya-ku Yokohama
(Received November 1, 2013)

縮器と学習・識別器を別々に構成する必要があるため，次元圧縮後の入力ベクトルを用いて正しくパターン識別が行なえる保証はない．

一方，Bu らは次元圧縮後の特徴空間において GMM を仮定することで，入力ベクトルの次元圧縮とデータが従う GMM のパラメータを学習的に獲得可能な判別成分分析（local Discriminant Component Analysis: 以下，DCA と略記）[12]を提案しており，さらに Tsuji らは DCA を NN に展開して多チャネル EMG 信号の識別に成功している[13]．ただし，DCA は入力データの時系列特性を考慮して圧縮と識別を行なうことはできないため，EEG のような非定常性の強い信号を識別することは困難である．また，線形判別分析（Linear Discriminant Analysis: 以下，LDA と略記）とリカレント確率 NN を組み合わせて時系列データを圧縮・識別する方法も提案されている[14]が，入力データの線形分離性が期待されないような，非線形なパターン識別問題を取り扱うことはできない．

そこで本論文では，入力データの時系列特性を考慮して圧縮・識別できるように拡張した新しい時系列パターン識別モデル，時系列判別成分分析（Time-series DCA: 以下，TSDCA と略記）を提案するとともに，TSDCA をネットワーク構造に展開した新しいリカレント NN を提案する．提案モデルは，正規直交変換行列を複数用いて入力データを圧縮し，次元圧縮後の特徴ベクトルが GMM を確率密度関数にもつ HMM に従うと仮定することで，入力時系列データの各クラスに対する事後確率を算出することができる．これにより圧縮と識別に関するモデルパラメータを一括的に学習可能となり，非線形特性を有する時系列データに対する高精度な識別が期待できる．

以下，**2.** で提案する TSDCA について述べ，**3.** で提案 NN のネットワーク構造と学習則について説明する．**4.** と **5.** では，提案 NN を用いた高次元人工データの識別実験および脳波信号識別実験により提案 NN の有効性を検討し，最後にまとめとする．

2.　時系列判別成分分析（TSDCA）

HMM はマルコフ過程に従って遷移する内部状態と，各状態における出力確率から構成される確率モデルであり，時系列データの識別などによく用いられている．特に HMM の出力確率密度関数に連続関数を仮定した Continuous Density HMM（以下，CD-HMM と略記）が広く用いられている．Baum ら[15]は与えられた観測データから EM アルゴリズムで HMM のパラメータを推定するアルゴリズムを提案した．また，Tsuji らは GMM 型の CD-HMM を NN の構造へ展開し，パラメータを学習的に獲得できることを示した[6]．

しかしながら，通常の CD-HMM では入力次元数の増加にともなって GMM のパラメータ数が増加するため，高次元の時系列データを識別することは困難である．この問題に対し，本論文では Bu らによって提案された DCA を CD-HMM に導入することで高次元の時系列データを低次元に圧縮して識別するモデルを考える．

2.1　高次元時系列データの圧縮

いま，d 次元の時系列データ $\boldsymbol{x}(t) \in \Re^d$ の各クラス c（$c = 1, \cdots, C$; C はクラス数）に対する事後確率を算出することを考え，入力データ $\boldsymbol{x}(t)$ を d' 次元の新たな特徴空間へ写像する問題をとりあげる．一般に CD-HMM では，入力空間において各データが GMM 型の確率密度関数をもつ HMM に従うと仮定する．それに対して提案法では，圧縮後の特徴空間において入力ベクトルがコンポーネント数 $M_{c,k}$，状態数 K_c の CD-HMM に従うと仮定し，新たなベクトル $\boldsymbol{x}'^{(c,k,m)}(t) \in \Re^{d'}$ を次式のように表現する．

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

ただし，$k = 1, \cdots, K_c$，$m = 1, \cdots, M_{c,k}$ であり，$\boldsymbol{\mu}^{(c,k,m)} \in \Re^d$ は入力空間においてコンポーネント $\{c, k, m\}$ に属するデータの平均ベクトル，$\mathbf{V}^{(c,k,m)} \in \Re^{d \times d'}$ は d 次元の特徴空間から d' 次元に圧縮された特徴空間への正規直交変換行列とする．また，$[\cdot]^{\mathrm{T}}$ は行列の転置を表わす．

このとき，圧縮された特徴空間において写像後のデータは以下の確率密度関数に従う．

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

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

ただし，$\Sigma'^{(c,k,m)} \in \Re^{d' \times d'}$ は圧縮後の特徴空間における分散共分散行列である．

2.2　事後確率の推定

写像後のデータが GMM 型の CD-HMM に従うとすれば，入力データ $\boldsymbol{x}(t)$ のクラス c に対する事後確率 $P(c|\boldsymbol{x}(t))$ は以下のように算出できる．

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

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

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

ただし，$\gamma_{k',k}^c$ はクラス c における状態 k' から k の遷移確率，$b_k^c(\boldsymbol{x}(t))$ は $\boldsymbol{x}(t)$ に対応したクラス c の状態 k からの事後確率を表わしており，事前確率 π_k^c は $P(c, k)|_{t=0}$ に等しい．

このとき，事後確率 $b_k^c(\boldsymbol{x}(t))$ が次元圧縮後の空間において (2) 式を 1 つのコンポーネントとする GMM に従うため，(6) 式の $\gamma_{k',k}^c b_k^c(\boldsymbol{x}(t))$ は，

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

となる．ただし $r_{c,k,m}$ は GMM の混合度である．

このように，TSDCA は入力された高次元の時系列データの次元を圧縮し，各識別対象クラスに対する事後確率の算出を可能とする．

2.3 モデルの線形化

前述のモデルに含まれる多数のパラメータは，定義や経験則を基に構造を決定する必要がある．特に HMM の学習には大量の学習データが必要であり，学習データが少ない場合には最適なパラメータの推定が困難となる可能性がある．この問題に対し Tsuji らは対数線形化を行なうことで HMM のパラメータをより少ない係数で表現できることを示した[6]．

ここでは，(1) 式および (7) 式を係数行列と入力ベクトルの線形和で表現することを考える．まず，次元圧縮後の平均ベクトルを

$$\hat{\boldsymbol{\mu}}^{(c,k,m)} = \mathbf{V}^{(c,k,m)^{\mathrm{T}}} \boldsymbol{\mu}^{(c,k,m)} \quad (8)$$

とおき，新たな入力ベクトル $\mathbf{X}(t) = [1, \boldsymbol{x}(t)^{\mathrm{T}}]^{\mathrm{T}} \in \Re^{d+1}$ と以下の係数行列 ${}^{(1)}\mathbf{W}^{(c,k,m)}$ を定義する．

$${}^{(1)}\mathbf{W}^{(c,k,m)} = \begin{bmatrix} -\hat{\mu}_1^{(c,k,m)} & V_{1,1}^{(c,k,m)} & \cdots & V_{1,d}^{(c,k,m)} \\ \vdots & \vdots & \ddots & \vdots \\ -\hat{\mu}_{d'}^{(c,k,m)} & V_{d',1}^{(c,k,m)} & \cdots & V_{d',d}^{(c,k,m)} \end{bmatrix} \quad (9)$$

これにより，$\boldsymbol{x}'^{(c,k,m)}(t)$ を係数行列 ${}^{(1)}\mathbf{W}^{(c,k,m)}$ と入力ベクトル $\mathbf{X}(t)$ の積で表わすことができる．

$$\begin{aligned} &\boldsymbol{x}'^{(c,k,m)}(t) \\ &= \mathbf{V}^{(c,k,m)^{\mathrm{T}}}(\boldsymbol{x}(t) - \boldsymbol{\mu}^{(c,k,m)}) \\ &= \mathbf{V}^{(c,k,m)^{\mathrm{T}}}\boldsymbol{x}(t) - \hat{\boldsymbol{\mu}}^{(c,k,m)} \\ &= {}^{(1)}\mathbf{W}^{(c,k,m)}\mathbf{X}(t) \end{aligned} \quad (10)$$

つぎに，

$$\xi^c_{k',k,m}(t) = \gamma^c_{k',k} r_{c,k,m} g(\boldsymbol{x}(t); c, k, m) \quad (11)$$

とおき，$\xi^c_{k',k,m}(t)$ の対数をとることで新たなベクトル $\mathbf{X}'^{(c,k,m)}(t)$ に対して線形化する．

$$\log \xi^c_{k',k,m}(t) = {}^{(2)}\mathbf{W}^{(c,k',k,m)}\mathbf{X}'^{(c,k,m)}(t) \quad (12)$$

ここで，係数ベクトル ${}^{(2)}\mathbf{W}^{(c,k',k,m)}$ は

$$\begin{aligned} &{}^{(2)}\mathbf{W}^{(c,k',k,m)} \\ &= [\log \gamma^c_{k',k} + \log r_{c,k,m} - \frac{d'}{2}\log 2\pi - \frac{1}{2}\log|\Sigma'^{(c,k,m)}|, \\ &\quad -\frac{1}{2}s'^{(c,k,m)}_{1,1}, -s'^{(c,k,m)}_{1,2}, \cdots, s'^{(c,k,m)}_{1,d'}, \cdots, \\ &\quad -\frac{1}{2}(2-\delta_{i,j})s'^{(c,k,m)}_{i,j} \cdots, -\frac{1}{2}s'^{(c,k,m)}_{d',d'}] \end{aligned} \quad (13)$$

であり，$s'^{(c,k,m)}_{1,1}, \cdots, s'^{(c,k,m)}_{d',d'}$ は逆行列 $(\Sigma'^{(c,k,m)})^{-1}$ の各要素，$\delta_{i,j}$ はクロネッカのデルタで，$i = j$ のとき 1，$i \neq j$ のとき 0 の値をとる．また，$\mathbf{X}'^{(c,k,m)}(t) \in \Re^H (H = 1 + \frac{d'(d'+1)}{2})$ は，

$$\begin{aligned} &\mathbf{X}'^{(c,k,m)}(t) \\ &= [1, x_1'^{(c,k,m)}(t)^2, x_1'^{(c,k,m)}(t)x_2'^{(c,k,m)}(t), \cdots, \\ &\quad x_1'^{(c,k,m)}(t)x_{d'}'^{(c,k,m)}(t), x_2'^{(c,k,m)}(t)^2, \\ &\quad x_2'^{(c,k,m)}(t)x_3'^{(c,k,m)}(t), \cdots, \\ &\quad x_2'^{(c,k,m)}(t)x_{d'}'^{(c,k,m)}(t), \cdots, x_{d'}'^{(c,k,m)}(t)^2]^{\mathrm{T}} \end{aligned} \quad (14)$$

である．ただし，$t = 1$ のときは

$$\xi^c_{k',k,m}(1) = \pi^c_k r_{c,k,m} g(\boldsymbol{x}(1); c, k, m) \quad (15)$$

とおき同様に線形化を行なうことで

$$\log \xi^c_{k',k,m}(1) = {}^{(2)}\mathbf{W}^{(c,k',k,m)}(1)\mathbf{X}'^{(c,k,m)}(1) \quad (16)$$

が得られる．このとき，${}^{(2)}\mathbf{W}^{(c,k',k,m)}(1)$，${}^{(2)}\mathbf{W}^{(c,k',k,m)}$ ともに無制約の重み係数とみなすことで ${}^{(2)}\mathbf{W}^{(c,k',k,m)}(1) = {}^{(2)}\mathbf{W}^{(c,k',k,m)}$ とすることができる[6]．

以上のようにモデルを線形化することにより，TSDCA に含まれるパラメータをより少ない係数 ${}^{(1)}\mathbf{W}^{(c,k,m)}$，${}^{(2)}\mathbf{W}^{(c,k',k,m)}$ で表現できる．これらの係数を適切に設定できれば，モデルパラメータおよび構造を決定でき，高次元時系列データの各クラスに対する事後確率を計算可能である．

次節では ${}^{(1)}\mathbf{W}^{(c,k,m)}$，${}^{(2)}\mathbf{W}^{(c,k',k,m)}$ を NN の重み係数として展開して学習的に獲得する方法について示す．

3. リカレント確率ニューラルネットへの展開

3.1 ネットワーク構造

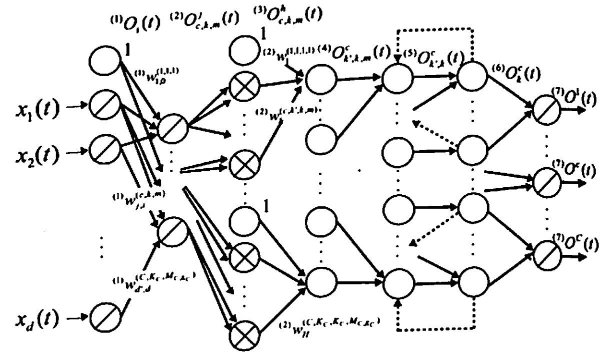


Fig. 1 Structure of the proposed neural network

Fig. 1 に提案 NN の構造を示す．ここで，図中の ⊘ は入力信号に対する線形和ユニット，⊗ は積ユニットを表わす．提案 NN は 7 層構造をとっており，第 1 層と第 2 層の間，および第 3 層と第 4 層の間にそれぞれ重み係数 ${}^{(1)}\mathbf{W}^{(c,k,m)}$，${}^{(2)}\mathbf{W}^{(c,k',k,m)}$ が存在し，第 5 層と第 6 層の間にリカレント結合を有する．

d 次元の時系列データ $\boldsymbol{x}(t)$ $(t = 1, 2, \cdots, T)$ をネットワークの入力とするとき，第 1 層は $d+1$ 個のユニットから構成され，その入出力関係は

$$^{(1)}I_i(t) = \begin{cases} 1 & (i=0) \\ x_i(t) & (i=1,\cdots,d) \end{cases} \tag{17}$$

$$^{(1)}O_i(t) = {}^{(1)}I_i(t) \tag{18}$$

となる．$^{(1)}I_i(t)$ と $^{(1)}O_i(t)$ は i 番目のユニットの入出力を表わす．

第 2 層は $C \times K_c \times M_{c,k} \times d'$ 個のユニットから構成され，第 1 層のユニットからの出力を重み係数 $^{(1)}w_{j,i}^{(c,k,m)}$ を介して受け取る．ユニット $\{j,c,k,m\}$ $(j=1,\cdots,d',\ c=1,\cdots,C,\ k=1,\cdots,K_c,\ m=1,\cdots,M_{c,k})$ の入力 $^{(2)}I_{c,k,m}^{j}(t)$ と出力 $^{(2)}O_{c,k,m}^{j}(t)$ の関係は

$$^{(2)}I_{c,k,m}^{j}(t) = \sum_{i=0}^{d} {}^{(1)}O_i(t)^{(1)}w_{j,i}^{(c,k,m)} \tag{19}$$

$$^{(2)}O_{c,k,m}^{j}(t) = {}^{(2)}I_{c,k,m}^{j}(t) \tag{20}$$

となる．ただし，重み係数 $^{(1)}w_{j,i}^{(c,k,m)}$ は次式の行列 $^{(1)}\mathbf{W}^{(c,k,m)}$ の各要素を表わす．

$$^{(1)}\mathbf{W}^{(c,k,m)} = \begin{bmatrix} ^{(1)}w_{1,0}^{(c,k,m)} & \cdots & ^{(1)}w_{1,d}^{(c,k,m)} \\ \vdots & \ddots & \vdots \\ ^{(1)}w_{d',0}^{(c,k,m)} & \cdots & ^{(1)}w_{d',d}^{(c,k,m)} \end{bmatrix} \tag{21}$$

第 3 層は $C \times K_c \times M_{c,k} \times H$ 個 $(H = 1 + \frac{d'(d'+1)}{2})$ のユニットから構成される．ユニット $\{h,c,k,m\}$ $(h=1,\cdots,H)$ の入力を $^{(3)}I_{c,k,m}^{h}(t)$，出力を $^{(3)}O_{c,k,m}^{h}(t)$ とするとき，その入出力関係は

$$^{(3)}I_{c,k,m}^{h}(t) = \begin{cases} 1 & (h=1) \\ ^{(2)}O_{c,k,m}^{j}(t)^{(2)}O_{c,k,m}^{j'}(t) & \\ \quad (h = j' - \frac{1}{2}j^2 + (d' + \frac{1}{2})j - d' + 1) & \end{cases} \tag{22}$$

$$^{(3)}O_{c,k,m}^{h}(t) = {}^{(3)}I_{c,k,m}^{h}(t) \tag{23}$$

である．ただし，$j \leq j' (j'=1,\cdots,d')$ であり，(22) 式は (14) 式の非線形変換に対応する．

第 4 層は $C \times K_c^2 \times M_{c,k}$ 個のユニットから構成される．ユニット $\{c,k',k,m\}(k'=1,\cdots,K_c)$ は第 3 層のユニットからの出力を重み係数 $^{(2)}w_h^{(c,k',k,m)}$ を介して受け取る．入力 $^{(4)}I_{k',k,m}^{c}(t)$ と出力 $^{(4)}O_{k',k,m}^{c}(t)$ の関係は以下のようになる．

$$^{(4)}I_{k',k,m}^{c}(t) = \sum_{h=1}^{H} {}^{(3)}O_{c,k,m}^{h}(t)^{(2)}w_h^{(c,k',k,m)} \tag{24}$$

$$^{(4)}O_{k',k,m}^{c}(t) = \exp\left({}^{(4)}I_{k',k,m}^{c}(t)\right) \tag{25}$$

ただし，重み係数 $^{(2)}w_h^{(c,k',k,m)}$ はベクトル $^{(2)}\mathbf{W}^{(c,k',k,m)}$ の各要素を表わす．

$$^{(2)}\mathbf{W}^{(c,k',k,m)} = \left[{}^{(2)}w_1^{(c,k',k,m)}, \cdots, {}^{(2)}w_H^{(c,k',k,m)}\right] \tag{26}$$

第 5 層は $C \times K_c^2$ 個のユニットから構成され，ユニット $\{c,k',k\}$ への入力 $^{(5)}I_{k',k}^{c}(t)$ は，第 4 層の出力を統合したものであり，その入力に 1 時刻前の第 6 層の出力 $^{(6)}O_{k'}^{c}(t-1)$ を乗じた値が第 5 層の出力 $^{(5)}O_{k',k}^{c}(t)$ となる．

$$^{(5)}I_{k',k}^{c}(t) = \sum_{m=1}^{M_{c,k}} {}^{(4)}O_{k',k,m}^{c}(t) \tag{27}$$

$$^{(5)}O_{k',k}^{c}(t) = {}^{(6)}O_{k'}^{c}(t-1)^{(5)}I_{k',k}^{c}(t) \tag{28}$$

ただし，初期状態は $^{(6)}O_{k'}^{c}(0) = 1.0$ である．

第 6 層は $C \times K_c$ 個のユニットから構成される．ユニット $\{c,k\}$ の入力 $^{(6)}I_k^c(t)$ と出力 $^{(6)}O_k^c(t)$ の関係は

$$^{(6)}I_k^c(t) = \sum_{k'=1}^{K_c} {}^{(5)}O_{k',k}^{c}(t) \tag{29}$$

$$^{(6)}O_k^c(t) = \frac{^{(6)}I_k^c(t)}{\sum_{c'=1}^{C}\sum_{k'=1}^{K_{c'}} {}^{(6)}I_{k'}^{c'}(t)} \tag{30}$$

で与えられる．

第 7 層は C 個のユニットで構成され，入力を $^{(7)}I^c(t)$，出力を $^{(7)}O^c(t)$ として

$$^{(7)}I^c(t) = \sum_{k=1}^{K_c} {}^{(6)}O_k^c(t) \tag{31}$$

$$^{(7)}O^c(t) = {}^{(7)}I^c(t) \tag{32}$$

と表わされる．この $^{(7)}O^c(t)$ がクラス c に対する事後確率 $P(c|\boldsymbol{x}(t))$ に相当する．このとき，NN の重み係数 $^{(1)}\mathbf{W}^{(c,k,m)}$，$^{(2)}\mathbf{W}^{(c,k',k,m)}$ を適切に設定できれば，モデルに基づく事後確率 $P(c|\boldsymbol{x}(t))$ の計算が可能となる．

3.2 学習則

いま，時刻 T において n 番目の入力ベクトル $\boldsymbol{x}^{(n)}(t)$ に対応する教師ベクトル $Q^{(n)} = [Q_1^{(n)},\cdots,Q_c^{(n)},\cdots,Q_C^{(n)}]^{\mathrm{T}} (n=1,\cdots,N)$ が与えられたとし，重み係数 $^{(1)}\mathbf{W}^{(c,k,m)}$，$^{(2)}\mathbf{W}^{(c,k',k,m)}$ を獲得する問題について考える．提案 NN は次式で定義する評価関数 J を最小化，すなわち対数尤度を最大化するように学習を行なう．

$$J = \sum_{n=1}^{N} J_n = -\sum_{n=1}^{N}\sum_{c=1}^{C} Q_c^{(n)} \log {}^{(7)}O^c(T)^{(n)} \tag{33}$$

ここで，$^{(7)}O^c(T)^{(n)}$ は時刻 T における入力ベクトル $\boldsymbol{x}^{(n)}(t)$ に対する出力である．このとき，2 つの重み係数 $^{(1)}w_{j,i}^{(c,k,m)}$ と $^{(2)}w_h^{(c,k',k,m)}$ の修正量を勾配法に基づいて以下の式で定義する．

$$\Delta^{(1)}w_{j,i}^{(c,k,m)} = -\gamma\sum_{n=1}^{N}\frac{\partial J_n}{\partial^{(1)}w_{j,i}^{(c,k,m)}} \tag{34}$$

$$\Delta^{(2)}w_h^{(c,k',k,m)} = -\gamma\sum_{n=1}^{N}\frac{\partial J_n}{\partial^{(2)}w_h^{(c,k',k,m)}} \tag{35}$$

ただし γ は学習率である．

重みの更新には通時誤差逆伝播法 (BPTT) [16] を用い，時系列中の誤差勾配を蓄積して，重みの修正量を計算する．こ

こで *chain rule* を用いると (34), (35) 式は以下の式で展開される.

$$
\begin{aligned}
&\frac{\partial J_n}{\partial^{(1)} w_{j,i}^{(c,k,m)}} \\
&= -\sum_{t=0}^{T-1}\sum_{c'=1}^{C}\sum_{k''=1}^{K_{c'}} \Delta_{k''}^{c'}(t) \frac{\partial^{(6)} O_{k''}^{c'}(T-t)}{\partial^{(6)} I_k^c(T-t)} \\
&\times \sum_{k'=1}^{K_c} \frac{\partial^{(6)} I_k^c(T-t)}{\partial^{(5)} O_{k',k}^c(T-t)} \frac{\partial^{(5)} O_{k',k}^c(T-t)}{\partial^{(5)} I_{k',k}^c(T-t)} \\
&\times \frac{\partial^{(5)} I_{k',k}^c(T-t)}{\partial^{(4)} O_{k',k,m}^c(T-t)} \frac{\partial^{(4)} O_{k',k,m}^c(T-t)}{\partial^{(4)} I_{k',k,m}^c(T-t)} \\
&\times \sum_{h=1}^{H} \frac{\partial^{(4)} I_{k',k,m}^c(T-t)}{\partial^{(3)} O_{c,k,m}^h(T-t)} \frac{\partial^{(3)} O_{c,k,m}^h(T-t)}{\partial^{(2)} I_{c,k,m}^j(T-t)} \\
&\times \frac{\partial^{(2)} I_{c,k,m}^j(T-t)}{\partial^{(1)} w_{j,i}^{(c,k,m)}(T-t)} \\
&= -\sum_{t=0}^{T-1}\sum_{c'=1}^{C}\sum_{k''=1}^{K_{c'}} \Delta_{k''}^{c'}(t) \\
&\times (\delta_{(c',k''),(c,k)} - {}^{(6)}O_{k''}^{c'}(T-t)) \frac{{}^{(6)}O_{k''}^{c'}(T-t)}{{}^{(6)}I_{k''}^{c'}(T-t)} \\
&\times \sum_{k'=1}^{K_c} {}^{(6)}O_{k'}^{c}(T-t-1)\,{}^{(4)}O_{k',k,m}^{c}(T-t) \\
&\times \sum_{h=1}^{H} {}^{(2)}w_h^{(c,k',k,m)} \frac{\partial^{(3)} O_{c,k,m}^h(T-t)}{\partial^{(2)} I_{c,k,m}^j(T-t)} \\
&\times {}^{(1)}O_i(T-t)
\end{aligned}
\qquad (36)
$$

$$
\begin{aligned}
&\frac{\partial J_n}{\partial^{(2)} w_{h}^{(c,k',k,m)}} \\
&= -\sum_{t=0}^{T-1}\sum_{c'=1}^{C}\sum_{k''=1}^{K_{c'}} \Delta_{k''}^{c'}(t) \frac{\partial^{(6)} O_{k''}^{c'}(T-t)}{\partial^{(6)} I_k^c(T-t)} \\
&\times \frac{\partial^{(6)} I_k^c(T-t)}{\partial^{(5)} O_{k',k}^c(T-t)} \frac{\partial^{(5)} O_{k',k}^c(T-t)}{\partial^{(5)} I_{k',k}^c(T-t)} \\
&\times \frac{\partial^{(5)} I_{k',k}^c(T-t)}{\partial^{(4)} O_{k',k,m}^c(T-t)} \frac{\partial^{(4)} O_{k',k,m}^c(T-t)}{\partial^{(4)} I_{k',k,m}^c(T-t)} \\
&\times \frac{\partial^{(4)} I_{k',k,m}^c(T-t)}{\partial^{(2)} w_h^{(c,k',k,m)}} \\
&= -\sum_{t=0}^{T-1}\sum_{c'=1}^{C}\sum_{k''=1}^{K_{c'}} \Delta_{k''}^{c'}(t) \\
&\times (\delta_{(c',k''),(c,k)} - {}^{(6)}O_{k''}^{c'}(T-t)) \frac{{}^{(6)}O_{k''}^{c'}(T-t)}{{}^{(6)}I_{k''}^{c'}(T-t)} \\
&\times {}^{(6)}O_{k'}^{c}(T-t-1)\,{}^{(4)}O_{k',k,m}^{c}(T-t) \\
&\times {}^{(2)}O_{c,k,m}^h(T-t)
\end{aligned}
\qquad (37)
$$

なお, $\Delta_{k''}^{c'}(t)$ は J_n の ${}^{(6)}O_{k''}^{c'}(T-t)$ による偏微分として次式で与えられる.

$$
\Delta_{k''}^{c'}(t) = \frac{\partial J_n}{\partial^{(6)} O_{k''}^{c'}(T-t)} \qquad (38)
$$

ただし,

$$
\begin{aligned}
&\Delta_{k''}^{c'}(0) \\
&= \frac{\partial(Q_{c'} \log {}^{(7)}O^{c'}(T))}{\partial^{(7)} O^{c'}(T)} \frac{\partial^{(7)} O^{c'}(T)}{\partial^{(6)} O_{k''}^{c'}(T)} \\
&= \frac{Q_{c'}}{{}^{(7)}O^{c'}(T)}
\end{aligned}
\qquad (39)
$$

$$
\begin{aligned}
&\Delta_{k''}^{c'}(t+1) \\
&= \sum_{c''=1}^{C}\sum_{k'''=1}^{K_{c''}} \Delta_{k'''}^{c''}(t) \sum_{k''''=1}^{K_{c''}} \frac{\partial^{(6)} O_{k'''}^{c''}(T-t)}{\partial^{(6)} I_{k''''}^{c'}(T-t)} \\
&\times \frac{\partial^{(6)} I_{k''''}^{c'}(T-t)}{\partial^{(5)} O_{k'',k''''}^{c'}(T-t)} \frac{\partial^{(5)} O_{k'',k''''}^{c'}(T-t)}{\partial^{(6)} O_{k''}^{c'}(T-(t+1))} \\
&= \sum_{c''=1}^{C}\sum_{k'''=1}^{K_{c''}} \Delta_{k'''}^{c''}(t) \\
&\times \sum_{k''''=1}^{K_{c''}} (\delta_{(c'',k'''),(c',k'''')} - {}^{(6)}O_{k'''}^{c''}(T-t)) \\
&\times \frac{{}^{(6)}O_{k'''}^{c''}(T-t)}{{}^{(6)}I_{k'''}^{c''}(T-t)} {}^{(5)}I_{k'',k''''}^{c'}(T-t)
\end{aligned}
\qquad (40)
$$

である. また, ${}^{(3)}O_{c,k,m}^h(T-t)$ の ${}^{(2)}I_{c,k,m}^j(T-t)$ による偏微分は以下の式で求められる.

$$
\frac{\partial^{(3)} O_{c,k,m}^h(T-t)}{\partial^{(2)} I_{c,k,m}^j(T-t)}
= \begin{cases}
2\,{}^{(2)}I_{c,k,m}^j(T-t) & (j=s=l) \\
{}^{(2)}I_{c,k,m}^l(T-t) & (j=s,\ j\neq l) \\
{}^{(2)}I_{c,k,m}^s(T-t) & (j\neq s,\ j=l) \\
0 & (otherwise)
\end{cases}
\qquad (41)
$$

ここで整数 h, s, l は以下の関係をみたす.

$$
h = -\frac{1}{2}s^2 + (d' + \frac{1}{2})s + l - d' + 1 \qquad (1 \le s \le l \le d') \quad (42)
$$

このとき, 変換行列 $\mathbf{V}^{(c,k,m)}$ の直交性を保持するため, 重みの更新のたびに ${}^{(1)}\mathbf{W}^{(c,k,m)}$ の $\mathbf{V}^{(c,k,m)}$ 成分に対し, (43), (44) 式に示すグラムシュミットの直交化を行なう.

$$
\boldsymbol{v'}_n = \boldsymbol{v}_n^{\mathrm{old}} - \sum_{i=1}^{n-1} (\boldsymbol{v}_i^{\mathrm{new}} \cdot \boldsymbol{v}_n^{\mathrm{old}}) \boldsymbol{v}_i^{\mathrm{new}} \qquad (43)
$$

$$
\boldsymbol{v}_n^{\mathrm{new}} = \frac{\boldsymbol{v'}_n}{\| \boldsymbol{v'}_n \|_2} \qquad (n = 1, \cdots, d') \quad (44)
$$

$\boldsymbol{v}_n^{\mathrm{old}}$ と $\boldsymbol{v}_n^{\mathrm{new}}$ はそれぞれ変換前および変換後における $\mathbf{V}^{(c,k,m)}$ の n 番目の列ベクトルである.

以上の学習則を用いることで, 圧縮に関する重み係数と識別に関する重み係数を一括して学習可能である.

4. 人工データ識別実験

高次元時系列データに対する提案 NN の識別能力を確認するため, 人工的に生成したデータの識別実験を行なった. こ

こではまず，(i) 提案 NN の状態数 K_c，コンポーネント数 $M_{c,k}$，圧縮後の次元数 d' が高次元時系列データの識別精度および学習時間に与える影響を明らかにする．そして，(ii) 従来の次元圧縮を行なわないニューラルネットと比較し同等以上の識別精度で学習時間を削減できることを示すとともに，(iii) 提案 NN の次元圧縮能力を従来の次元圧縮法とニューラルネットを組み合わせた識別法と比較する．以下，4.1 節で提案 NN のパラメータ検証，4.2 節，4.3 節で識別・圧縮能力の比較実験について述べる．なお，各実験に用いた PC は Intel Core(TM) i7-3770K（クロック周波数：3.5 GHz），メモリ：16.0 GB である．

4.1　ネットワークパラメータの検証

4.1.1　方法

識別対象とする $\boldsymbol{y}(t) \in \Re^d$ はクラスごとに HMM に基づいて生成した d 次元のデータ $\boldsymbol{x}(t) \in \Re^d$ と，平均 0，標準偏差 1 のホワイトガウスノイズ $\boldsymbol{\eta}(t) \in \Re^d$ を次式を用いて結合したデータとする．

$$\boldsymbol{y}(t) = (1-a)\boldsymbol{x}(t) + a\boldsymbol{\eta}(t) \tag{45}$$

ここで，a は $[0, 1)$ 間の係数とし，$\boldsymbol{x}(t) = [x_1(t), \cdots, x_d(t)]^{\mathrm{T}}$，$\boldsymbol{\eta}(t) = [\eta_1(t), \cdots, \eta_d(t)]^{\mathrm{T}}$ である．**Fig. 2** に $\boldsymbol{y}(t)$ の一例を示す．ここで，$\boldsymbol{y}(t)$ の生成に用いたパラメータは $C = 3$，$d = 30$，$T = 50$，$a = 0.8$ とし，図はそれぞれ横軸が時間，縦軸が各次元の信号の値を示している．$\boldsymbol{x}(t)$ の生成に用いた HMM のパラメータは，状態数・混合数をともに 2 とし，混合度と初期状態確率および状態遷移確率を和が 1 となるように正規化した $[0, 1]$ 間の一様乱数で，各コンポーネントの平均ベクトルおよび共分散行列を $[-1, 1]$ 間の一様乱数でそれぞれ設定した．このデータを 1 サンプルとし，$\boldsymbol{x}(t)$ および $\boldsymbol{\eta}(t)$ をサンプルごとに変更して各クラス 55 サンプル作成した（a は全サンプル共通）．そして 5 サンプルを学習データ，残りの 50 サンプルを識別データとして学習・識別を行なった．このとき，識別において $\boldsymbol{y}(t)$ の最終時刻における事後確率が最大となるクラスを識別結果とした．これを 1 セットとし，$\boldsymbol{\eta}(t)$ およびネットワークの初期重みを毎回変更して 100 セット計算した際の平均値を算出した．提案 NN の各パラメータは $K_c = 1, 2, 3$，$M_{c,k} = 1, 2, 3$，$d' = 5, 4, \cdots, 1$ とした．また，$\boldsymbol{y}(t)$ に含まれるノイズの割合を決めるパラメータ a は予備実験より $a = 0.8$ とした．なお，学習の高速化を行なうため Tsuji ら [6] の手法を参考にしてターミナルアトラクタ [17] を導入した．

4.1.2　結果

Fig. 3 は $\boldsymbol{y}(t)$ の作成に用いた a を決定するための予備実験の結果であり，$a = 0$ から $a = 0.9$ まで 0.1 刻みで変化させた際の提案 NN（$K_c = 1$，$M_{c,k} = 1$ および $K_c = 2$，$M_{c,k} = 2$）と HMM の平均識別率を示している．なお HMM は，平均ベクトルおよび共分散行列の初期値を k-means 法を用いて推定した値とし，残りの初期パラメータを乱数で設定して，EM アルゴリズムを用いて学習を行なった [15]．このとき，各手法の識別率に対して有意水準 1%で Holm 法に基づく多重比較検定を行なったところ，$a = 0.6, 0.7$ のとき提案 NN（$K_c = 1$，$M_{c,k} = 1$）と HMM，提案 NN（$K_c = 2$，$M_{c,k} = 2$）と HMM の各群間で有意差があり，$a = 0.8, 0.9$ のときすべての群間において有意な差を確認した．

Table 1 は状態数 $K_c = 1, 2, 3$ およびコンポーネント数 $M_{c,k} = 1, 2, 3$ のそれぞれの組み合わせに対する平均識別率と学習時間を示す．このとき，$d' = 1$ とした．この結果に対し，有意水準 1%で Holm 法に基づく多重比較検定を行なったところ，識別率に関しては ($K_c = 1$, $M_{c,k} = 1$) と ($K_c = 2$, $M_{c,k} = 2$)，($K_c = 1$, $M_{c,k} = 1$) と ($K_c = 2$, $M_{c,k} = 3$)，

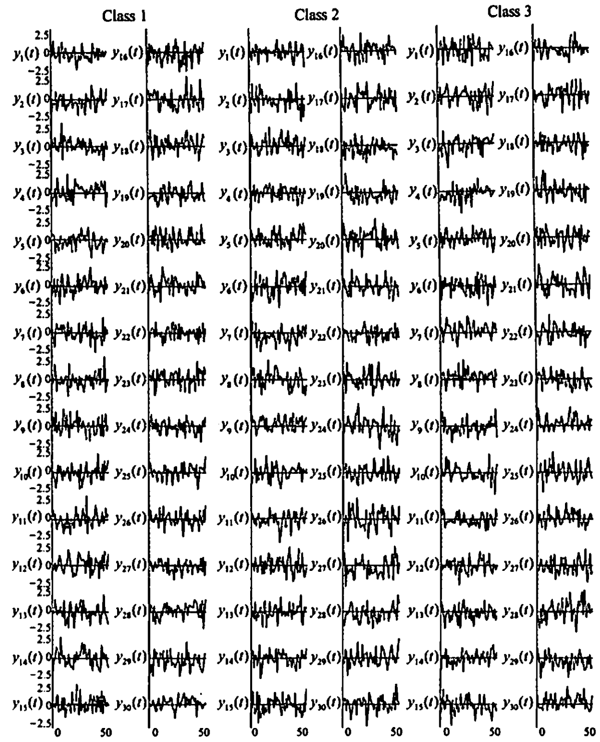


Fig. 2　Example set of artificial time series signals $\boldsymbol{y}(t)$

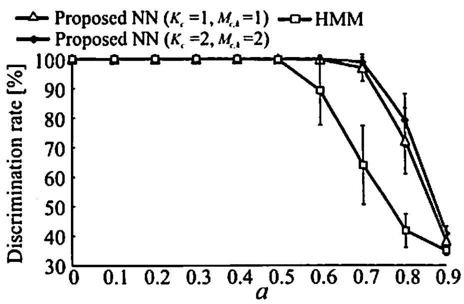


Fig. 3　Avarage discrimination rates for each a

Table 1　Average discrimination rates and learning time for each K_c and $M_{c,k}$

(a) Discrimination rates

$M_{c,k}$ \ K_c	1	2	3
1	72.3±11.2	76.8±9.4	79.2±9.1
2	76.3±9.8	79.3±9.0	78.7±8.2
3	77.0±9.9	79.3±8.3	79.1±8.1

Average ± Standard deviation [%]

(b) Learning time

$M_{c,k}$ \ K_c	1	2	3
1	1.0±0.1	5.2±1.0	15.1±1.5
2	1.9±0.1	10.5±1.6	30.7±3.6
3	2.8±0.2	15.5±2.1	48.7±7.0

Average ± Standard deviation [s]

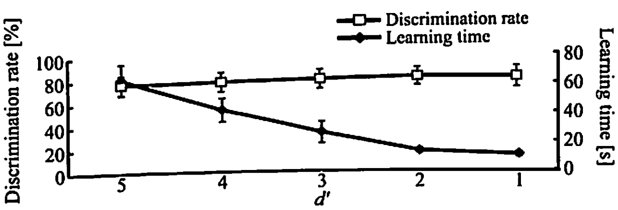


Fig. 4 Average discrimination rates and learning time for each d'

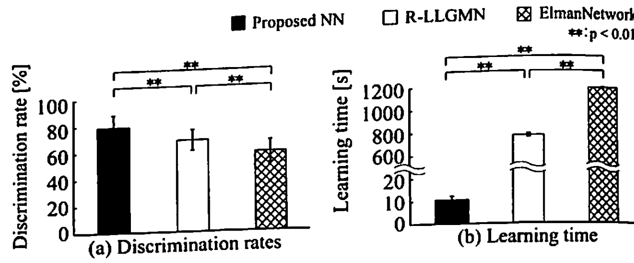


Fig. 5 Average discrimination rates and learning time for each method

$(K_c = 1,\ M_{c,k} = 1)$ と $(K_c = 3,\ M_{c,k} = 1)$，$(K_c = 1,\ M_{c,k} = 1)$ と $(K_c = 3,\ M_{c,k} = 2)$，$(K_c = 1,\ M_{c,k} = 1)$ と $(K_c = 3,\ M_{c,k} = 3)$，それぞれの群間において有意な差を確認した．また，学習時間に関してはすべての群間において有意差があった．

Fig. 4 に提案 NN を用いて圧縮した際の圧縮後の次元数 d' に対する平均識別率と平均学習時間を示す．ここで $K_c = 2$，$M_{c,k} = 2$ とした．このとき，$d' = 5, 4, \cdots, 1$ に対する識別率はそれぞれ $76.3 \pm 8.3\%$，$77.6 \pm 8.3\%$，$78.4 \pm 8.2\%$，$79.9 \pm 7.6\%$，$79.3 \pm 9.0\%$であり，すべての群間において有意な差はなかった．また，d' が小さいほど学習時間が短く，それぞれ 65.2 ± 10.9 s，42.9 ± 8.3 s，26.1 ± 7.5 s，13.1 ± 1.5 s，10.5 ± 1.6 s であり，すべての群間で有意差を確認した．

4.1.3 考察

Fig. 3 より，人工データに含まれるノイズの割合を増加させるにつれて識別問題が複雑になるため各手法の識別率が低下している．ただし，$a = 0.6$ 以上において提案 NN の識別率は HMM より有意に高い．これは，HMM がクラスごとにモデルパラメータを推定して識別するため，ノイズの影響を強く受けクラスごとの差をうまく学習できずよい汎化性が得られなかったのに対し，提案 NN では入力データの識別に必要な成分を抽出できたためであると考えられる．さらに，$a = 0.8$ 以上では提案 NN $(K_c = 2,\ M_{c,k} = 2)$ の識別率が $(K_c = 1,\ M_{c,k} = 1)$ の場合より有意に高く，これは K_c，$M_{c,k}$ を 2 以上にすることで提案モデルが有効にはたらき，複雑な時系列特性をモデル化できたためであると推測される．これらの結果より，提案 NN の有効性を他手法と比較し議論するためには $a = 0.8$ 以上が望ましく，以降の実験では $a = 0.8$ とした．

Table 1 に示す状態数およびコンポーネント数に対する識別率と学習時間では，(a) より $K_c = 1$，$M_{c,k} = 1$ のとき識別率が最小となり，$K_c = 2$，$M_{c,k} = 2$ などのときと比べて有意差が認められた．これは $K_c = 1$，$M_{c,k} = 1$ が，各クラスのデータ分布を 1 つの圧縮した正規分布のみで静的に推定するという特殊なケースに帰着するためだと考えられる．それ以外の組み合わせでは，K_c，$M_{c,k}$ を増やすと識別率はやや増加傾向を示すものの有意な差はみられなかった．また，(b) より K_c および $M_{c,k}$ が小さいほど学習時間が短い．これらの結果から，識別精度と学習時間のトレードオフを考慮し，以降の実験では $K_c = 2$, $M_{c,k} = 2$ としたが，厳密な設定方法については今後詳細に検討する必要があろう．

Fig. 4 に示す圧縮後の次元数 d' と識別率の関係から，いずれの d' に対してもある程度の精度で識別できていることがわかる．これは，提案 NN が入力データを各クラスの状態，コンポーネントごとに別々の空間へ圧縮でき，$d' = 1$ などの低次元に圧縮しても識別に必要な情報を抽出できるためだと考えられる．一方で学習時間においては，d' が小さいほど推定するネットワークのパラメータが少なくなるため学習時間が短い．

4.2 識別能力の比較

識別対象は 4.1 節で用いた高次元時系列データとした．比較対象は R-LLGMN [6] および ElmanNetwork [18] とした．ここで，R-LLGMN は GMM を有する HMM を対数線形化してネットワーク構造に展開したリカレント型の確率 NN であり，時系列データを高い精度で識別できる．提案 NN および R-LLGMN の状態数，コンポーネント数はそれぞれ $K_c = 2$，$M_{c,k} = 2$ とし，提案 NN の圧縮後の次元数は $d' = 1$ とした．また，ElmanNetwork は入力層，隠れ層，出力層の 3 層構造とし，入力層のユニット数は 30，隠れ層のユニット数は 10 とした．入出力関数には対数シグモイド関数を利用した．学習則には誤差逆伝播法を用い，誤差閾値を 0.01，学習率を 0.01 とした．

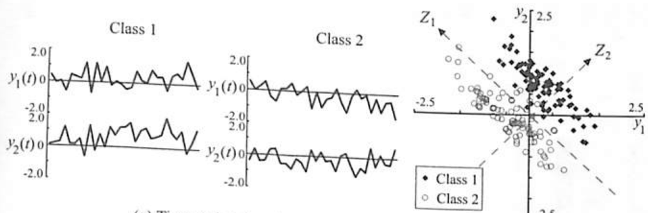
Fig. 5 に提案 NN，R-LLGMN，ElmanNetwork を用いて算出した平均識別率および平均学習時間を示す．(a) より提案 NN，R-LLGMN，ElmanNetwork の順に識別率が高く，それぞれ $79.3 \pm 9.0\%$，$68.6 \pm 7.7\%$，$59.1 \pm 8.6\%$であった．また (b) より平均学習時間は，提案 NN が 10.5 ± 1.6 s，R-LLGMN が 769.4 ± 16.3 s，ElmanNetwork が 1179.5 ± 6.4 s であった．

各手法を比較すると，ElmanNetwork は高次元データに対しノンパラメトリックな推定を行なったため，識別問題が複雑になり十分な識別能力が得られず，学習にも多くの時間がかかった．R-LLGMN は既知のモデル構造を内包するためネットワークの複雑さは ElmanNetwork より改善されるものの，入力データの全次元を用いて識別を行なうため学習には平均 770 s ほど必要としている．これに対し提案 NN は単一のネットワークで圧縮と識別を同時に実現するため，識別が可能な低次元空間への写像が学習によって獲得でき，識別精度の改善と学習時間の短縮が実現できたと考えられる．

4.3 次元圧縮能力の比較

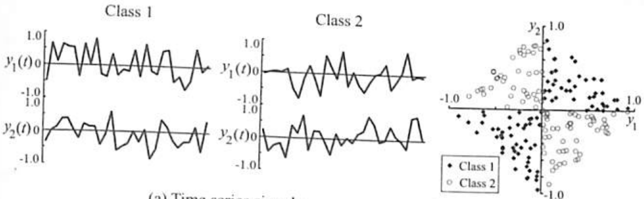
4.3.1 方法

比較対象は，PCA と R-LLGMN を組み合わせた識別法 (以



(a) Time series signals　(b) Scattergram of the population

Fig. 6　Example set of the principal component problem



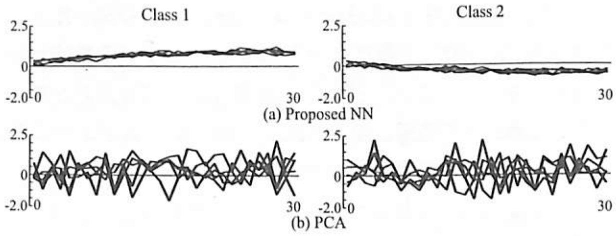
(a) Time series signals　(b) Scattergram of the population

Fig. 7　Example set of the non-linear problem

下，PCA with R-LLGMN）および RD-RLLGMN [14]とした．PCA with R-LLGMN および RD-RLLGMN はそれぞれ PCA，LDA と R-LLGMN を組み合わせており，高次元時系列データを圧縮して識別可能であるが，PCA は入力データの主成分が識別困難な場合に，LDA は非線形分離問題を取り扱う場合にそれぞれ識別精度が大きく低下する可能性がある．実験ではまず，PCA および LDA で識別困難になる状況を意図的に作り出した 2 種類の 2 次元時系列データを用いて，識別精度と圧縮後の時系列データを算出し，提案 NN の次元圧縮と PCA および LDA の違いを明確にする．そして，これらの違いが高次元時系列データの識別においてどのように影響するかを 4.1 節のデータで確認する．**Fig. 6** に PCA との比較に用いたデータ（以下，主成分問題）を示す．(a) は各次元の時系列データの一例，(b) はデータの発生に用いた母集団の分布を示す散布図である．(b) の補助軸 Z_1，Z_2 はそれぞれデータ全体の最も分散が大きい方向（第 1 主成分）および，それと直交する方向（第 2 主成分）を示している．また，**Fig. 7** に LDA との比較に用いたデータ（以下，非線形問題）を示す．Fig. 6 と同様，(a) は各次元の時系列データの一例，(b) はデータの発生に用いた母集団の分布である．(b) の分布は代表的な非線形分離問題である排他的論理和を参考にして，Class 1 のデータを第 1・第 3 象限に，Class 2 のデータを第 2・第 4 象限にそれぞれ分布させており，明らかな非線形分離問題として作成した．このとき，5 サンプルを学習データ，100 サンプルを識別データとするデータセットを 10 セット作成し，それぞれのデータセットにおいて初期重みを 10 回変更して学習・識別を行なった際の平均識別率を算出した．

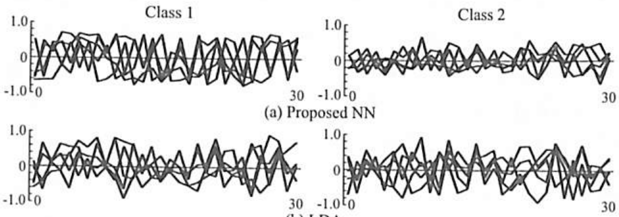
4.3.2　結果と考察

Fig. 8 は主成分問題（Fig. 6）を提案 NN および PCA を用いて圧縮した波形であり，5 例重ねて示している．なお，提案 NN の圧縮波形は学習後のネットワークから Vitabi アルゴリズム [19]を用いて計算した最尤の状態推移系列から算出した．



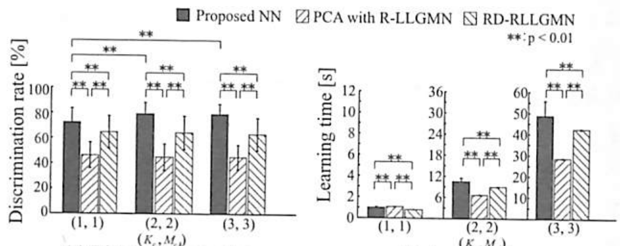
(a) Proposed NN　(b) PCA

Fig. 8　Reduced dimensional signals for the principal component problem



(a) Proposed NN　(b) LDA

Fig. 9　Reduced-dimensional signals for the non-linear problem



(a) Discrimination rates　(b) Learning time

Fig. 10　Average discrimination rates and learning time for each method

このとき，各手法の識別率は提案 NN が 100%，PCA with R-LLGMN が 49.6 ± 3.1%であった．

Fig. 9 に，非線形問題（Fig. 7）を提案 NN および LDA を用いて圧縮した波形（5 例）を示す．この問題に対する各手法の識別率は，提案 NN が 95.0 ± 6.6%，RD-RLLGMN が 66.1 ± 9.6%であり，有意な差が確認された．

Fig. 10 に 4.1 節で使用した高次元時系列データを，提案 NN, PCA with R-LLGMN, RD-RLLGMN を用いて学習・識別した際の平均識別率および学習時間を示す．図は，各手法で K_c，$M_{c,k}$ の組み合わせを $(K_c, M_{c,k}) = (1,1),(2,2),(3,3)$ と変更したときの値を示している．このとき，提案 NN および PCA の圧縮後の次元数は $d' = 1$ とした．また，RD-RLLGMN の圧縮後の次元数はクラス数から $C - 1 = 2$ となる [14]．(a) より各手法の識別率は提案 NN が 72.3 ± 11.2%, 79.3±9.0%, 79.1±8.1%, PCA with R-LLGMN が 46.3 ± 10.3%，45.0 ± 10.4%，45.1 ± 9.6%，RD-RLLGMN が 64.7 ± 12.9%，64.4 ± 13.3%，63.5 ± 12.8%であった．ただし，値は順に $(K_c, M_{c,k}) = (1,1),(2,2),(3,3)$ のときのものである．同様に，学習時間は (b) より提案 NN が 1.0±0.1 s, 10.5±1.6 s, 48.7±7.0 s, PCA with R-LLGMN が 1.1 ± 0.0 s，6.7 ± 0.0 s，28.5 ± 0.1 s，RD-RLLGMN が 0.8 ± 0.0 s，9.0 ± 0.1 s，42.3 ± 0.2 s であった．識別率に関

して，同一 $(K_c, M_{c,k})$ における各手法間，および同一手法における各 $(K_c, M_{c,k})$ 間において Holm 法に基づく多重比較検定を行なったところ，図に示す群間において有意差を確認した．なお，学習時間に関しては同一 $(K_c, M_{c,k})$ における各手法間において検定を行なった．

Fig. 8 (a) より，主成分問題（Fig. 6）を提案 NN で圧縮した波形は，ほぼすべての時刻で Class 1 は正の値を，Class 2 は負の値を示しており明確な違いが確認できる．これは，提案 NN が各クラスの特徴の違いを分離できる Z_2 軸（Fig. 6 (b) 参照）方向への圧縮を獲得できたためと考えられる．一方，(b) の PCA による圧縮波形では，Class 1 と Class 2 の特徴の違いを確認することは困難である．これは PCA が，各クラスの特徴の違いを抽出できない第 1 主成分 Z_1 軸方向への圧縮を行なったためである．また，これらの次元圧縮の結果は識別精度へも大きく影響し，提案 NN の識別率は PCA with R-LLGMN の識別率を大幅に上回った．以上より，提案 NN の圧縮は入力データの主成分が識別困難な場合に有効であることが明らかになった．

Fig. 9 (a) に示す提案 NN を用いた非線形問題（Fig. 7）の圧縮波形は，Fig. 8 (a) のような明確な分離は確認できないものの，Class 1 の振幅が Class 2 の振幅より大きいことがみてとれる．一方で (b) の LDA による圧縮波形では Class 1 と Class 2 の振幅の違いは確認できない．ここで，各手法の圧縮波形 100 例について標準偏差を計算したところ，提案 NN は Class 1 が 0.49 ± 0.03，Class 2 が 0.29 ± 0.03 で有意差があったのに対し，LDA は Class 1 が 0.40 ± 0.04，Class 2 が 0.40 ± 0.04 で有意な差は確認できなかった．提案 NN の識別率が RD-RLLGMN より高かったのは，提案 NN が振幅情報としてクラスごとの違いを保存できたのに対し，RD-RLLGMN は LDA での圧縮により識別に必要な情報が失われてしまったと考えられる．これらより，LDA では特徴抽出が困難になる非線形分離問題に対しても提案 NN の次元圧縮は有効であることが示された．

Fig. 10 (a) の高次元データに対する識別率の比較において，各 $(K_c,\ M_{c,k})$ で提案 NN は他手法より有意に高い識別率を示している．また，識別部において重要なパラメータである K_c，$M_{c,k}$ を増加させた場合，PCA with R-LLGMN と RD-RLLGMN は識別率の有意な変化がみられなかったのに対し，提案 NN のみ (1, 1) から (2, 2) や (3, 3) へ増やしたとき識別率の有意な増加がみられた．これは，PCA with R-LLGMN や RD-RLLGMN は次元圧縮部の PCA や LDA において識別に有用な情報が大きく失われるが，提案 NN の次元圧縮では必要な情報がある程度保存されているためであると考えられる．以上の結果より，PCA や LDA で圧縮すると識別困難になる高次元時系列データに対して，提案 NN が有用であることがわかった．

5. 脳波信号識別実験

提案 NN の実データへの有効性を検証するために，UCI

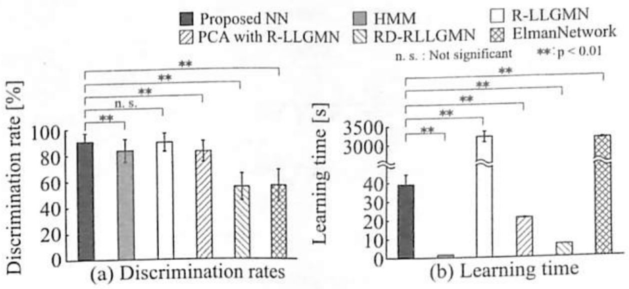


Fig. 11 Average discrimination rates and learning time for each method

Machine Learning Repository（注 1）よりダウンロードした 64 チャネルの脳波データを用いて識別実験を行なった ($d = 64$)．被験者は健常者 1 名とし，2 種類の刺激（$C = 2$；(a) 1 つの視覚刺激，(b) 2 つの異なる視覚刺激）を与えたときの EEG 信号を識別した．実験では，各クラス 1 秒間のデータを 1 サンプルとして 10 サンプル分のデータのうち 3 サンプルを学習データ，残りの 7 サンプルを識別データとした．また，学習データと識別データの組み合わせを無作為に 10 通り変更し，それぞれの組み合わせにおいて初期重みを 10 回変更して学習・識別を行なった際の平均識別率を算出した．提案 NN の各パラメータは $K_c = 2$，$M_{c,k} = 2$，$d' = 1$ とした．また，そのほかの条件は前小節と同じとした．

Fig. 11 に各手法を用いて算出した平均識別率と平均学習時間，および提案 NN と他手法の間での多重比較検定結果を示す．(a) より平均識別率は提案 NN，R-LLGMN，HMM，PCA with R-LLGMN, ElmanNetwork，RD-RLLGMN の順に高く，それぞれ $90.6 \pm 6.3\%$，$89.6 \pm 7.0\%$，$83.4 \pm 8.9\%$，$82.4 \pm 8.0\%$，$55.9 \pm 12.1\%$，$55.1 \pm 10.2\%$であった．また (b) より平均学習時間は，提案 NN が 39.1 ± 5.4 s，HMM が 1.4 ± 0.0 s，R-LLGMN が 3215.0 ± 140.5 s，PCA with R-LLGMN が 21.1 ± 0.2 s，RD-RLLGMN が 6.7 ± 0.0 s，ElmanNetwork が 3150.5 ± 15.1 s であった．このことから，提案 NN は R-LLGMN や ElmanNetwork のような従来のニューラルネットと比較して大幅に学習時間を削減でき，EEG 信号のような高次元時系列データを圧縮してある程度の精度で識別できることが明らかになった．

6. まとめと今後の課題

本論文では，入力データの時系列特性を考慮して圧縮・識別できる新しい時系列パターン識別モデルとして時系列判別成分分析（TSDCA）を提案した．提案モデルは，正規直交変換行列を複数用いて入力データを圧縮し，次元圧縮後の特徴ベクトルが GMM 型の CD-HMM に従うと仮定することで，入力時系列データの各クラスに対する事後確率を算出することができる．さらに，TSDCA をリカレント NN の構造に展開することにより，モデルパラメータを NN の重み係数

（注 1）http://archive.ics.uci.edu/ml/databases/eeg/eeg.data.html

として学習的に獲得でき，モデルに基づく事後確率の算出が可能であることを示した．

実験では，提案 NN の識別能力を確認するため人工的に生成した高次元時系列データを入力として識別実験を行ない，ネットワークパラメータが識別結果に与える影響を議論した．また，提案ネットワークの次元圧縮能力によって短時間の学習が可能となり，PCA や LDA で識別困難となる問題に対して有用であることを示した．さらに，実データを用いた実験により，脳波信号識別への応用可能性を確認した．

今後は，直交化計算を考慮した学習方法について検討するとともに，状態数やコンポーネント数などの適切な決定方法について検討し，BCI など生体信号を用いたインタフェースへの応用を試みる予定である．

参 考 文 献

1) D.E. Rumelhart, G.E. Hintont and R.J. Williams: Learning representations by back-propagating errors, Nature, **323**-6088, 533/536 (1986)
2) M.F. Kelly, P.A. Parker and R.N. Scott: Neural network classification of myoelectric signal for prosthesis control, J. Electromyograph. Kinesiol., **1**-4, 229/236 (1991)
3) E. Haselsteiner and G. Pfurtscheller: Using time-dependent neural networks for EEG Classification, IEEE Trans. Rehabilitat. Eng., **8**-4, 457/463 (2000)
4) M. Agarwal, H. Agrawal, N. Jain and M. Kumar: Face Recognition Using Principle Component Analysis, Eigenface and Neural Network, Int. Conf. Signal Acquisit. Process., 2010, 310/314 (2010)
5) T. Tsuji, O. Fukuda, H. Ichinobe and M. Kaneko: A Log-Linearized Gaussian Mixture Network and Its Application to EEG Pattern Classification, IEEE Trans. Syst. Man Cybern. Part C, **29**-1, 60/72 (1999)
6) T. Tsuji, N. Bu, M. Kaneko and O. Fukuda: A Recurrent Log-linearized Gaussian Mixture Network, IEEE Trans. Neural Netw., **14**-2, 304/316 (2003)
7) N. Bu, O. Fukuda and T. Tsuji: EMG-Based Motion Discrimination Using a Novel Recurrent Neural Network, J. Intell. Informat. Syst., **21**-2, 113/126 (2003)
8) B.O. Peters, G. Pfurtscheller and H. Flyvbjerg: Mining multi-channelEEG for its information content: An ANN-based method for a brain-computer interface, Neural Netw., **11**-7, 1429/1433 (1998)
9) S. Zhong and J. Ghosh: HMMs and Coupled HMMs for multi-channel EEG classification, Proc. 2002 Int. Joint Conf. Neural Netw., 1154/1159 (2002)
10) N.F. Güler and S. Kocer: Classification of EMG Signals Using PCA and FFT, J. Med. Syst., **29**-3, 241/250 (2005)
11) S. Lawrence, C.L. Giles, A.C. Tsoi and A.D. Back: Face Recognition: A Convolutional Neural Network Approach, IEEE Trans. Neural Netw., **8**-1, 98/113 (1997)
12) N. Bu and T. Tsuji: Multivariate Pattern Classification based on Local Discriminant Component Analysis, IEEE Int. Conf. Robot. Biomim., 924/929 (2004)
13) T. Tsuji, N. Bu, J. Arita and M. Ohga: A speech synthesizer using facial EMG signals, Int. J. Computat. Intell. Appl., **7**-1, 1/15 (2008)
14) N. Bu, K. Shima and T. Tsuji: EEG Discrimination using Wavelet Packet Transform and a Reduced-dimensional Recurrent Neural Network, Proc. 10th IEEE Int. Conf. Informat. Technol. Appl. Biomed., Paper-ID 261 (2010)
15) L.E. Baum, T. Petrie, G. Soules and N. Weiss: A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains, Ann. Math. Stat., **41**-1, 164/171 (1970)
16) P.J. Werbos: Backpropagation through time: What it does and how to do it, Proc. IEEE, **78**-10, 1550/1560 (1990)
17) M. Zak: Terminal Attractors for Addressable Memory in Neural Networks, Phys. Lett. A, **133**, 18/22 (1988)
18) J.L. Elman: Finding structure in time, Cognit. Sci., **14**-2, 179/211 (1990)
19) A. Viterbi: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Trans. Informat. Theory, **13**-2, 260/269 (1967)

[著 者 紹 介]

早 志 英 朗

2012 年広島大学工学部第二類卒業．現在同大学大学院工学研究科博士課程前期在学中．生体信号解析，パターン識別，ニューラルネットに関する研究に従事．IEEE の会員．

島　　圭 介（正会員）

2007 年広島大学大学院工学研究科博士課程前期修了．2009 年同博士課程後期修了．横浜国立大学大学院工学研究院助教を経て，2013 年同准教授．博士（工学）．生体信号解析，ニューラルネット，ヒューマンインタフェースなどの研究に従事．IEEE などの会員．

芝 軒 太 郎

2008 年徳島大学工学部知能情報工学科卒業．2010 年広島大学大学院工学研究科博士課程前期修了．2012 年同博士課程後期修了．2013 年より広島大学大学院工学研究院特任助教．博士（工学）．生体信号解析，次元削減，ヒューマンインタフェースなどの研究に従事．IEEE などの会員．

栗 田 雄 一（正会員）

2004 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（工学）．奈良先端科学技術大学院大学情報科学研究科助教を経て，2011 年より広島大学大学院工学研究院准教授．同年より科学技術振興機構さきがけ研究員を兼任．IEEE，日本ロボット学会などの会員．

辻　　敏 夫（正会員）

1985 年広島大学大学院工学研究科博士課程前期修了．同年広島大学工学部助手．94 年同助教授を経て，2002 年同大学大学院工学研究科教授，現在に至る．工学博士．人間とロボットの運動制御，ニューラルネット，マン・マシンシステムなどの研究に従事．IEEE，日本人間工学会，バイオメカニズム学会，日本機械学会などの会員．