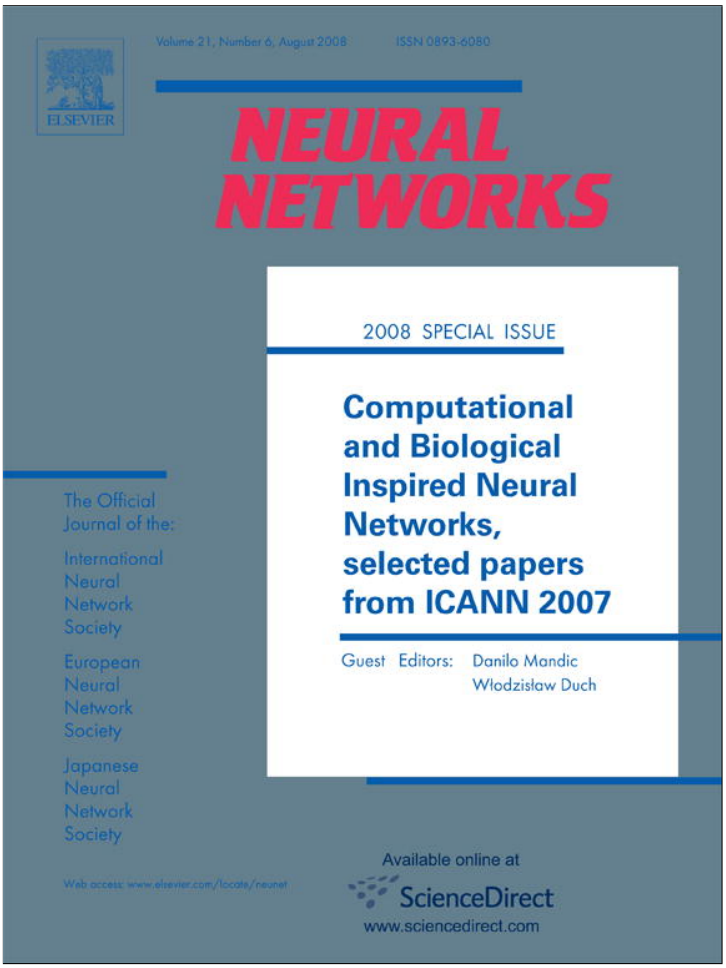

Neural Networks 21 (2008) 838–846

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2008 Special Issue

Iterative principles of recognition in probabilistic neural networks

Jiří Grim *, Jan Hora

Institute of Information Theory and Automation, Czech Academy of Sciences P.O. BOX 18, CZ-18208 Prague 8, Czech Republic
Faculty of Nuclear Science and Physical Engineering, Czech Technical University Trojanova 13, CZ-120 00 Prague 2, Czech Republic

ARTICLE INFO

Article history:
Received 15 January 2008
Accepted 14 March 2008

Keywords:
Probabilistic neural networks
Distribution mixtures
EM algorithm
Recognition of numerals
Recurrent reasoning

ABSTRACT

When considering the probabilistic approach to neural networks in the framework of statistical pattern recognition we assume approximation of class-conditional probability distributions by finite mixtures of product components. The mixture components can be interpreted as probabilistic neurons in neurophysiological terms and, in this respect, the fixed probabilistic description contradicts the well known short-term dynamic properties of biological neurons. By introducing iterative schemes of recognition we show that some parameters of probabilistic neural networks can be "released" for the sake of dynamic processes without disturbing the statistically correct decision making. In particular, we can iteratively adapt the mixture component weights or modify the input pattern in order to facilitate correct recognition. Both procedures are shown to converge monotonically as a special case of the well known EM algorithm for estimating mixtures.

1. Introduction

The concept of probabilistic neural networks (PNNs) relates to the early work of Specht (1988) and others (cf. Haykin (1993), Hertz, Krogh, and Palmer (1991), Palm (1994), Streit and Luginbuhl (1994) and Watanabe and Fukumizu (1995)). In this paper we consider the probabilistic approach to neural networks based on distribution mixtures with product components in the framework of statistical pattern recognition. We refer mainly to our papers on PNNs published in the last years (cf. Grim (1996a) –Grim, Somol, and Pudil (2005)). In order to design PNNs we approximate the unknown class-conditional distributions by finite mixtures of product components. In particular, given a training data set for each class, we compute the estimates of mixture parameters by means of the well known EM algorithm (Dempster, Laird, & Rubin, 1977; Grim, 1982; McLachlan & Peel, 2000; Schlesinger, 1968). Let us recall that, given the class-conditional probability distributions, the Bayes decision function minimizes the classification error.

The mixture-based PNNs do not provide a new biologically motivated technique of statistical pattern recognition. Nevertheless, the interpretation of a theoretically well justified statistical method provides an opportunity to understand complex functional principles of biological neural networks which are rarely observable in a pure form. The main idea of PNNs is to view the components of mixtures as formal neurons. In this way there is a possibility of explaining the properties of biological neurons in terms of the component parameters. Therefore, the primary motivation of our previous research was to demonstrate different neuromorphic features of PNNs. Simultaneously, the underlying statistical method has been modified in order to improve its compatibility with biological neural networks.

We have shown that the estimated mixture parameters can be used to define an information preserving transform with the aim of a sequential design of multilayer PNNs (Grim, 1996a, 1996b; Vajda & Grim, 1998). In the case of long training data sequences, the EM algorithm can be realized as a sequential procedure which corresponds to one "infinite" iteration of the EM algorithm, including periodic updating of the estimated parameters (Grim, 1999b; Grim, Just, & Pudil, 2003; Grim et al., 2005). In pattern recognition the classification accuracy can be improved by parallel combination of independently trained PNNs (Grim, Kittler, Pudil, & Somol, 2000; Grim, Pudil, & Somol, 2002). The probabilistic neuron can be interpreted in neurophysiological terms at the level of the functional properties of a biological neuron (Grim et al., 2003; Grim, Kittler, Pudil, & Somol, 2002). In this way we obtain an explicit formula for synaptic weights which can be seen as a theoretical counterpart of the well known Hebbian principle of learning (Hebb, 1949). The PNN can be trained sequentially while assuming the strictly modular properties of the probabilistic neurons (Grim, 1999a; Grim et al., 2003). Weighting of training data in PNNs is compatible with the technique of boosting which is widely used in pattern recognition (Grim et al., 2002). In this sense the importance of training data vectors may be evaluated selectively as it is assumed e.g. in connection with "emotional" learning. Also, there is a close relationship of PNNs with self-organizing maps (Grim, 2000).

* Corresponding author at: Institute of Information Theory and Automation, P.O. BOX 18, CZ-18208 Prague 8, Czech Republic. Tel.: +420 266052215; fax: +420 284683031.
E-mail address: grim@utia.cas.cz (J. Grim).

0893-6080/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2008.03.002

One of the most obvious limitations of the probabilistic approach to neural networks has been the biologically unnatural complete interconnection of neurons with all input variables. We have proposed a structural mixture model which avoids the biologically unnatural condition of complete interconnection of neurons. The resulting subspace approach to PNNs is compatible with the statistically correct decision making and, at the same time, optimization of the interconnection structure of PNNs can be included in the EM algorithm (Grim, 1999b; Grim et al., 2002; Grim, Pudil, & Somol, 2000).

Another serious limitation of PNNs arizes from the conflicting properties of the estimated fixed mixture parameters and of the well known short-term dynamic processes in biological neural networks. The mixture parameters computed by means of the EM algorithm reflect the global statistical properties of training data and uniquely determine the performance of PNNs. Unfortunately, the "static" role of the mixture parameters is sharply contrasted by the short-term dynamic properties of biological neurons. Motivated by this contradiction, we propose in this paper iterative schemes of recognition. In particular, we propose the iterative use of Bayes formula in order to adapt *a priori* component weights to a specific input. Simultaneously, with the same theoretical background, we consider iterative modification of input patterns in order to facilitate the recognition. In this way some mixture parameters may participate in the short-term dynamic processes without disturbing the statistically correct decision making

In the following we first describe the structural mixture model (Section 2) as applied to recognition of handwritten numerals (Section 3) and summarize the basic properties of PNNs (Section 4). In Section 5 we describe the proposed principles of iterative recognition and prove the convergence properties. Finally the results are summarized in the Conclusion.

2. Structural mixture model

In the following sections we confine ourselves to the problem of statistical recognition of binary data vectors

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_N) \in \mathcal{X}, \quad \mathcal{X} = \{0, 1\}^N \tag{1}$$

to be classified according to a finite set of classes $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$. Assuming a probabilistic description of classes we can reduce the final decision making to the Bayes formula

$$p(\omega|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\omega)p(\omega)}{P(\boldsymbol{x})}, \tag{2}$$
$$P(\boldsymbol{x}) = \sum_{\omega\in\Omega} P(\boldsymbol{x}|\omega)p(\omega), \quad \boldsymbol{x} \in \mathcal{X}$$

where $P(\boldsymbol{x}|\omega)$ represents the class-conditional probability distributions and $p(\omega)$, $\omega \in \Omega$ denotes the related *a priori* probabilities of classes. We recall that, in the case of exact probabilistic description, the Bayes decision function

$$d(\boldsymbol{x}) = \arg\max_{\omega\in\Omega}\{p(\omega|\boldsymbol{x})\}, \quad \boldsymbol{x} \in \mathcal{X} \tag{3}$$

minimizes the probability of classification error.

Placing us in the framework of PNN, we approximate the conditional distributions $P(\boldsymbol{x}|\omega)$ by finite mixtures of product components

$$P(\boldsymbol{x}|\omega) = \sum_{m\in\mathcal{M}_\omega} F(\boldsymbol{x}|m)f(m)$$
$$= \sum_{m\in\mathcal{M}_\omega} f(m) \prod_{n\in\mathcal{N}} f_n(x_n|m), \quad \sum_{m\in\mathcal{M}_\omega} f(m) = 1. \tag{4}$$

Here $f(m) \geq 0$ are probabilistic weights, $F(\boldsymbol{x}|m)$ denote the component specific product distributions, $\mathcal{M}_\omega$ are the component index sets of different classes and $\mathcal{N}$ is the index set of variables. In order to simplify notation we assume consecutive indexing of components. Hence, for each component index $m \in \mathcal{M}_\omega$ the related class $\omega \in \Omega$ is uniquely determined and therefore the parameter ω can be partly omitted in the above notation.

In the case of binary data vectors we assume the components $F(\boldsymbol{x}|m)$ to be multivariate Bernoulli distributions, i.e. we have

$$F(\boldsymbol{x}|m) = \prod_{n\in\mathcal{N}} f_n(x_n|m), \quad m \in \mathcal{M}_\omega, \mathcal{N} = \{1, \ldots, N\} \tag{5}$$

$$f_n(x_n|m) = \theta_{mn}^{x_n}(1-\theta_{mn})^{1-x_n}, \quad 0 \leq \theta_{mn} \leq 1, \ n \in \mathcal{N}. \tag{6}$$

We recall that any discrete probability distribution can be expressed in the form (4) when the number of components is sufficiently large (Grim et al., 2002).

The basic idea of PNNs is to view the component distributions in Eq. (4) as formal neurons. If we define the output of the m-th neuron in terms of the mixture component $F(\boldsymbol{x}|m)f(m)$ then the posterior probabilities $p(\omega|\boldsymbol{x})$ are proportional to partial sums of neural outputs:

$$p(\omega|\boldsymbol{x}) = \frac{p(\omega)}{P(\boldsymbol{x})} \sum_{m\in\mathcal{M}_\omega} F(\boldsymbol{x}|m)f(m). \tag{7}$$

The well known disadvantage of the probabilistic approach to neural networks is the biologically unnatural interconnection of neurons with all input variables. The complete interconnection property follows from the basic paradigms of probability theory. For the sake of Bayes formula, all class-conditional probability distributions must be defined in the same space and therefore each neuron must be connected with the same (complete) set of input variables.

In order to avoid the undesirable complete interconnection condition, we make use of the structural mixture model (Grim, 1999b; Grim et al., 2000, 2002) originally proposed in multivariate statistical pattern recognition (Grim, 1986). In particular, we set

$$F(\boldsymbol{x}|m) = F(\boldsymbol{x}|0)G(\boldsymbol{x}|m, \phi_m), \quad m \in \mathcal{M}_\omega \tag{8}$$

where $F(\boldsymbol{x}|0)$ is a "background" probability distribution, usually defined as a fixed product of global marginals

$$F(\boldsymbol{x}|0) = \prod_{n\in\mathcal{N}} f_n(x_n|0) = \prod_{n\in\mathcal{N}} \theta_{0n}^{x_n}(1-\theta_{0n})^{1-x_n},$$
$$(\theta_{0n} = \mathcal{P}\{x_n = 1\}) \tag{9}$$

and the component functions $G(\boldsymbol{x}|m, \phi_m)$ include additional binary structural parameters $\phi_{mn} \in \{0, 1\}$

$$G(\boldsymbol{x}|m, \phi_m) = \prod_{n\in\mathcal{N}} \left[\frac{f_n(x_n|m)}{f_n(x_n|0)}\right]^{\phi_{mn}}$$
$$= \prod_{n\in\mathcal{N}} \left[\left(\frac{\theta_{mn}}{\theta_{0n}}\right)^{x_n} \left(\frac{1-\theta_{mn}}{1-\theta_{0n}}\right)^{1-x_n}\right]^{\phi_{mn}}. \tag{10}$$

An important feature of PNNs is the possibility of optimizing the mixture parameters $f(m)$, θ_{mn} and the structural parameters ϕ_{mn}, simultaneously, by means of the EM algorithm (cf. e.g. Grim et al. (2002)). Given a training set of independent observations from the class $\omega \in \Omega$

$$\mathcal{S}_\omega = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K_\omega)}\}, \quad \boldsymbol{x}^{(k)} \in \mathcal{X},$$

we obtain the maximum-likelihood estimates of the mixture (4), (8) by maximizing the log-likelihood function

$$L = \frac{1}{|\mathcal{S}_\omega|} \sum_{x\in\mathcal{S}_\omega} \log\left[\sum_{m\in\mathcal{M}_\omega} F(\boldsymbol{x}|0)G(\boldsymbol{x}|m, \phi_m)f(m)\right]. \tag{11}$$

Table 1
Classification accuracy

Experiment no.:	1	2	3	4	5	6	7	8	9
Number of components	100	393	520	540	663	756	852	1015	1327
Number of parameters	87 525	342 593	377 822	408 461	478 473	515 715	642 623	539 453	848 557
Bayes decision rule	6.51	4.10	3.71	3.82	3.56	3.46	3.38	3.45	3.15
Manipulated weights	9.22	7.50	7.30	7.43	4.93	5.55	5.67	5.24	5.24
Iterated weights	6.52	4.11	3.73	3.85	3.59	3.48	3.40	3.46	3.17
Adapted input vector	6.52	4.10	3.71	3.83	3.57	3.47	3.39	3.45	3.16
Extended test data									
Bayes decision rule	5.39	3.27	2.85	2.95	2.74	2.61	2.55	2.68	2.34
Manipulated weights	5.42	3.46	3.00	3.07	2.83	2.74	2.71	2.77	2.46
Iterated weights	5.41	3.35	2.92	3.03	2.76	2.69	2.53	2.69	2.35
Adapted input vector	5.46	3.29	2.91	2.98	2.76	2.65	2.51	2.69	2.33

Recognition of numerals from the NIST SD19 database. Classification error in % obtained by different methods and for differently complex mixtures.

For this purpose we can derive the following EM iteration equations (Grim, 1986; Grim et al., 2002): ($m \in \mathcal{M}_\omega$, $n \in \mathcal{N}$, $\boldsymbol{x} \in \mathcal{S}_\omega$)

$$f(m|\boldsymbol{x}) = \frac{G(\boldsymbol{x}|m, \phi_m) f(m)}{\sum_{j \in \mathcal{M}_\omega} G(\boldsymbol{x}|j, \phi_j) f(j)}, \tag{12}$$

$$f'(m) = \frac{1}{|\mathcal{S}_\omega|} \sum_{x \in \mathcal{S}_\omega} f(m|\boldsymbol{x}), \tag{13}$$

$$\theta'_{mn} = \frac{1}{|\mathcal{S}_\omega| f'(m)} \sum_{x \in \mathcal{S}_\omega} x_n f(m|\boldsymbol{x}), \tag{14}$$

$$\gamma'_{mn} = f'(m) \left[\theta'_{mn} \log \frac{\theta'_{mn}}{\theta_{0n}} + (1 - \theta'_{mn}) \log \frac{(1 - \theta'_{mn})}{(1 - \theta_{0n})} \right],$$

$$\phi'_{mn} = \begin{cases} 1, & \gamma'_{mn} \in \Gamma', \\ 0, & \gamma'_{mn} \notin \Gamma', \end{cases}, \tag{15}$$

$$\Gamma' \subset \{\gamma'_{mn}\}_{m \in \mathcal{M}_\omega \; n \in \mathcal{N}}, \quad |\Gamma'| = r. \tag{16}$$

Here $f'(m)$, θ'_{mn}, and ϕ'_{mn} are the new iteration values of the mixture parameters and Γ' is the set of a given number of highest quantities γ'_{mn}. As can be seen, the structural optimization naturally arizing from the EM algorithm is controlled by the criterion γ'_{mn} which is proportional to Kullback–Leibler information divergence. In this sense the structural EM algorithm prefers the most informative variables. We also note that the structural mixture model can be applied to continuous data e.g. by considering Gaussian mixtures of product components (Grim, 1986; Grim, Haindl, Somol, & Pudil, 2006).

The iterative Eqs. (12)–(15) generate a nondecreasing sequence $\{L^{(t)}\}_0^\infty$ converging to a possibly local maximum of the log-likelihood function (11). However, in our experience, the meaning of local maxima is less relevant in case of large data sets and mixtures of several tens of components.

3. Recognition of handwritten numerals

Throughout the paper we illustrate the properties of PNNs by considering a practical problem of recognition of handwritten numerals. In order to estimate the class-conditional distributions we have used the well known NIST Special Database 19 (SD19) containing about 400 000 handwritten digits (Grother, 1995). Unlike our previous experiments (Grim & Hora, 2007), we have normalized the digit patterns to a 32×32 binary raster instead of 16×16, in order to achieve more precise pattern representation. In this way each digit pattern is described by an N-dimensional binary vector ($N = 1024 = 32 \times 32$), where the binary variables $x_n \in \{0, 1\}$ in (1) correspond to the raster fields in a given fixed order. Let us recall that the structural mixture model is invariant with respect to arbitrary permutation of variables in the vector $\boldsymbol{x}$ because the products in components are commutative.

The SD19 numerals have been widely used for benchmarking of classification algorithms. Unfortunately, there is no generally accepted partition of NIST numerals into training or testing subsets. In order to guarantee the same statistical properties of both data sets we have used the odd samples of each class for training and the even samples for testing.

The classification problem has been solved in the original 1024-dimensional binary space without employing any feature extraction or dimensionality reduction method. However, in order to increase the variability of hand-written numerals, the data sets have been extended by including additional rotated variants of the original patterns. In particular, each digit pattern has been rotated by -20, -10 and 10 degrees and the corresponding variants of the original pattern were included in the training data set. The idea of generating additional shifted or slightly rotated variants of the original patterns (Grim et al., 2002) is motivated by the well known microscopic movements of the human eye observing a fixed object (so called "saccadic" movements). Obviously the principle is applicable both to the training- and test data sets.

Using the extended training data, we have estimated the class-conditional mixtures of different complexity in nine different experiments by means of the EM algorithm of Section 2. In the experiments the initial number of mixture components was chosen identically in all classes with the component parameters initialized randomly. However, the structural optimization may cause some components to "lose" all specific parameters. The resulting "nonspecific" components can be replaced by a single one and therefore the final number of components may decrease. The total numbers of mixture components and of the specific parameters in each experiment are given in the second and third row of Table 1 respectively. Fig. 2 displays examples of the estimated component parameters θ_{mn} in raster arrangement. We may expect the component "means" to correspond to the typical variants of digit patterns from the respective classes.

The white raster fields in Fig. 2 represent the "unused" variables with the respective structural parameters $\phi_{mn} = 0$. Unlike Eq. (15) the structural parameters ϕ_{mn} have been chosen by using the computationally more efficient thresholding:

$$\phi'_{mn} = \begin{cases} 1, & \gamma'_{mn} \geq 0.1\gamma'_0, \\ 0, & \gamma'_{mn} < 0.1\gamma'_0, \end{cases},$$

$$\gamma'_0 = \frac{1}{|\mathcal{M}_\omega| N} \sum_{m \in \mathcal{M}_\omega} \sum_{n \in \mathcal{N}} \gamma'_{mn}. \tag{17}$$

Here γ'_0 is the mean value of the individual informativity of variables γ'_{mn}. The chosen coefficient 0.1 is relatively low because all variables are rather informative and the training data set is large enough to get reliable estimates of all the parameters. Hence, in each component the threshold value $0.1\gamma'_0$ actually suppresses only those variables which are in fact superfluous, as can be seen in Fig. 2. For each experiment the resulting total

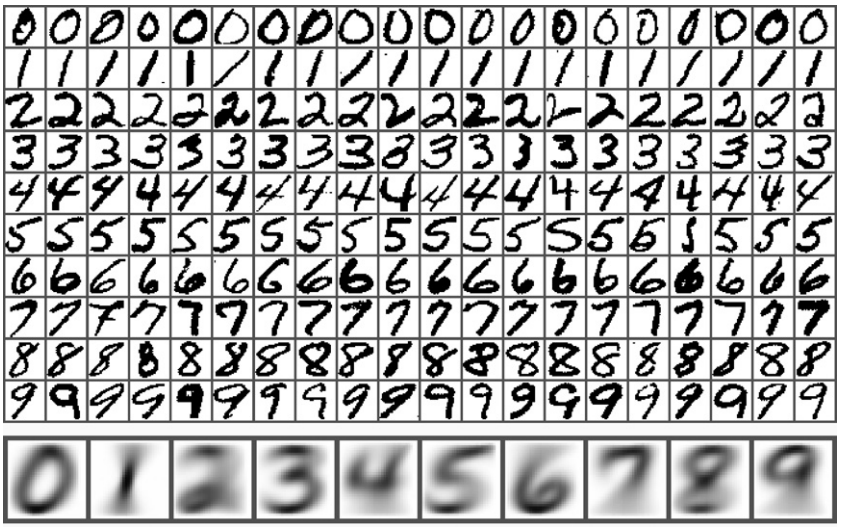

Fig. 1. Examples of the NIST numerals. Hand-written numerals from the NIST-database normalized to 32x32 binary raster and the respective class-means ("mean images").

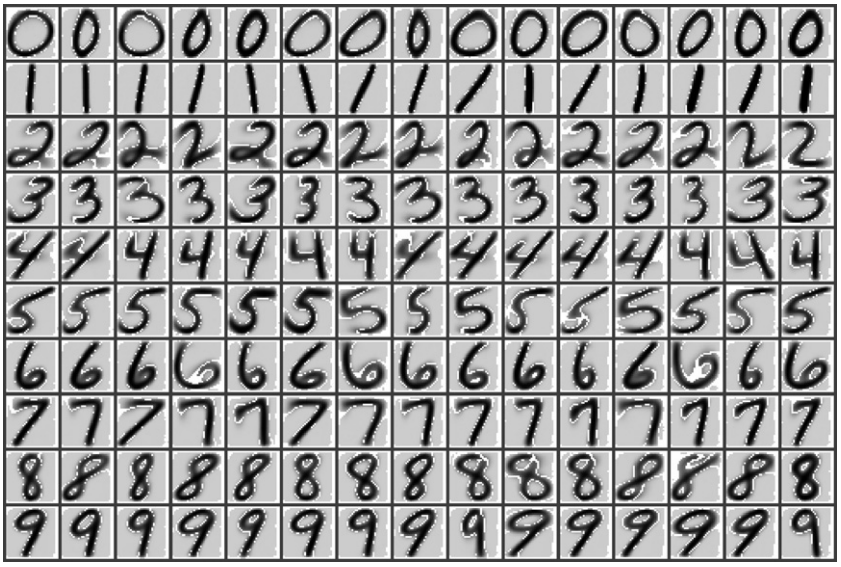

Fig. 2. Component parameters. Examples of the estimated component parameters θ_{mn} in raster arrangement. The component "means" in each row correspond to the typical variants of digit patterns from the respective classes. The white raster fields denote the "unused" variables specified by $\phi_{mn} = 0$.

number of component specific variables ($\sum_m \sum_n \phi_{mn}$) is given in the third row of Table 1. The EM algorithm has been stopped by the relative increment threshold $\Delta L = 10^{-5}$. In this way, the estimation procedure usually resulted in several tens ($10 \div 40$) of EM iterations.

We have verified the achieved recognition accuracy by using an independent test set and by considering four different types of decision making. We remark that in all comparable experiments the results are distinctly better than in the case of 16×16 raster resolution (Grim & Hora, 2007). The fourth row of Table 1 corresponds to the standard Bayes decision rule (3) based on the estimated class-conditional mixtures. In this case the mean recognition error decreases with the total number of mixture parameters from 6.51% (87 525 parameters, 100 components) to 2.34% (848 557 parameters, 1327 components). The next three rows of Table 1 relate to the iterative procedures of Section 5.

All classification rules have been applied to the extended test data set in an analogous way (cf. row 9 – 12). The four variants of each test pattern (original pattern and three rotations) have been first classified independently. The final Bayes decision rule (3) has been applied to *a posteriori* weights obtained by summing the respective four *a posteriori* probability distributions $p(\omega|\boldsymbol{x})$. In the case of the extended data set the mean recognition error of the standard Bayes rule is essentially lower, in particular, in the experiments 1 to 9 it is decreasing from 5.39% to 2.34% respectively. Fig. 3 contains examples of incorrectly classified patterns from the last experiment.

4. Probabilistic neural networks

The main advantage of the structural mixture model is the possibility of cancelling the background probability density $F(\boldsymbol{x}|0)$ in the Bayes formula, since then the decision making may be confined only to "relevant" variables. In particular, introducing notation

$$w_m = p(\omega) f(m), \quad m \in \mathcal{M}_\omega,\ \omega \in \Omega \tag{18}$$

and making substitution (8) in (4), we can express the unconditional probability distribution $P(\boldsymbol{x})$ in the form

$$P(\boldsymbol{x}) = \sum_{\omega\in\Omega} p(\omega) \sum_{m\in\mathcal{M}_\omega} F(\boldsymbol{x}|m) f(m) = \sum_{m\in\mathcal{M}} F(\boldsymbol{x}|0) G(\boldsymbol{x}|m, \phi_m) w_m. \quad \mathcal{M} = \bigcup_{\omega\in\Omega} \mathcal{M}_\omega. \tag{19}$$

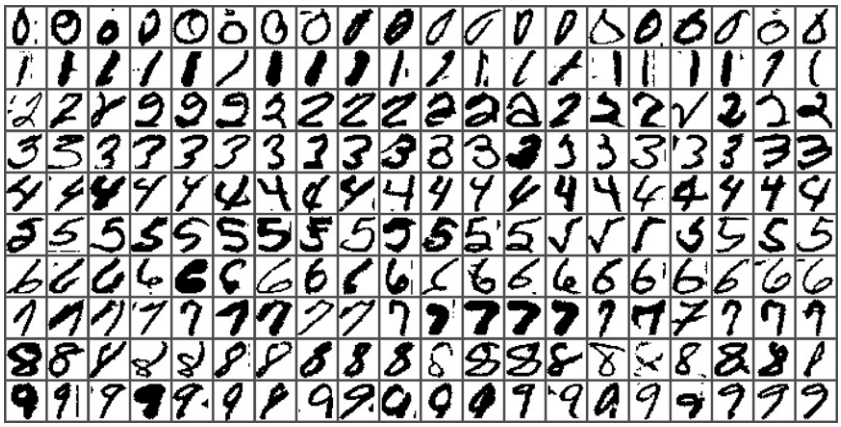

Fig. 3. Examples of misclassified numerals from the last numerical experiment (cf. last column of Table 1).

Further, considering the conditional component weights

$$q(m|\boldsymbol{x}) = \frac{F(\boldsymbol{x}|m)w_m}{P(\boldsymbol{x})} = \frac{G(\boldsymbol{x}|m,\phi_m)w_m}{\sum_{j\in\mathcal{M}} G(\boldsymbol{x}|j,\phi_j)w_j}, \tag{20}$$

we can write

$$\begin{aligned} p(\omega|\boldsymbol{x}) &= \frac{P(\boldsymbol{x}|\omega)p(\omega)}{P(\boldsymbol{x})} \\ &= \frac{\sum_{m\in\mathcal{M}_\omega} G(\boldsymbol{x}|m,\phi_m)w_m}{\sum_{j\in\mathcal{M}} G(\boldsymbol{x}|j,\phi_j)w_j} = \sum_{m\in\mathcal{M}_\omega} q(m|\boldsymbol{x}). \end{aligned} \tag{21}$$

Thus the posterior probability $p(\omega|\boldsymbol{x})$ becomes proportional to a weighted sum of the component functions $G(\boldsymbol{x}|m,\phi_m)$, each of which can be defined in a different subspace. Consequently the input connections of a neuron can be confined to an arbitrary subset of input neurons or, in other words, the "receptive fields" of neurons can be arbitrarily specified.

The structural mixture model represents a statistically correct subspace approach to Bayesian decision making. It is directly applicable to the input space without employing any feature selection- or dimensionality reduction method (Grim, 1986, 1999b; Grim et al., 2000, 2002). In the literature subspace approaches usually refer to the "subspace projection method" originally proposed by Watanabe (1967) and Watanabe and Pakvasa (1973) and later modified by many other authors (Hertz et al., 1991; Oja, 1983, 1989; Oja & Kohonen, 1988; Prakash & Murty, 1997; Workshop: Subspace, 2007). The subspace projection approaches are computationally advantageous but they do not provide a statistically correct decision scheme because they are not properly normalizable.

In multilayer neural networks each neuron of a hidden layer plays the role of a coordinate function of a vector transform $\boldsymbol{T}$ mapping the input space $\mathcal{X}$ into the space of output variables $\mathcal{Y}$. We denote

$$\begin{aligned} &\boldsymbol{T}: \mathcal{X} \to \mathcal{Y}, \quad \mathcal{Y} \subset R^M, \\ &\boldsymbol{y} = \boldsymbol{T}(\boldsymbol{x}) = (T_1(\boldsymbol{x}), T_2(\boldsymbol{x}), \ldots, T_M(\boldsymbol{x})) \in \mathcal{Y}. \end{aligned} \tag{22}$$

It has been shown (cf. Grim (1996b); Vajda and Grim (1998)) that the transform defined in terms of the posterior probabilities $q(m|\boldsymbol{x})$:

$$y_m = T_m(\boldsymbol{x}) = \log q(m|\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X},\ m \in \mathcal{M} \tag{23}$$

preserves the statistical decision information and minimizes the entropy of the output space $\mathcal{Y}$. Loosely speaking, the transformation (23) "unifies" only the points $\boldsymbol{x} \in \mathcal{X}$ of identical *a posteriori* probabilities $q(m|\boldsymbol{x})$ and therefore the implicit partition of the input space $\mathcal{X}$ induced by the inverse mapping $\boldsymbol{T}^{-1}$ does not cause any information loss. Simultaneously, the induced partition of the input space can be shown to be the "simplest" one in the sense of the minimum entropy property.

From the neurophysiological point of view, conditional probability $q(m|\boldsymbol{x})$ can be naturally interpreted as a measure of excitation or probability of firing of the m-th neuron given the input pattern $\boldsymbol{x} \in \mathcal{X}$. In view of Eqs. (10) and (20) we can write

$$\begin{aligned} y_m &= T_m(\boldsymbol{x}) = \log q(m|\boldsymbol{x}) \\ &= \log w_m + \sum_{n\in\mathcal{N}} \phi_{mn} \log \frac{f_n(x_n|m)}{f_n(x_n|0)} \\ &\quad - \log\left[\sum_{j\in\mathcal{M}} G(\boldsymbol{x}|j,\phi_j)w_j\right]. \end{aligned} \tag{24}$$

The logarithm in Eq. (24) expands the excitation quantity $q(m|\boldsymbol{x})$ into different additive contributions. Consequently, we may assume the first term on the right-hand side of Eq. (24) to be responsible for the spontaneous activity of the m-th neuron. The second term in Eq. (24) summarizes contributions of the input variables x_n (input neurons). The choice of input variables is specified by means of the binary structural parameters $\phi_{mn} = 1$. In this sense, the term

$$g_{mn}(x_n) = \log f_n(x_n|m) - \log f_n(x_n|0) \tag{25}$$

represents the synaptic weight of the n-th neuron at the input of the m-th neuron, as a function of the input value x_n.

The synaptic weight (25) is defined generally as a function of the input variable x_n and therefore separates the weight g_{mn} from the particular input values x_n. The effectiveness of the synaptic transmission, as expressed by the weight function (25), combines the statistical properties of the input variable x_n with the activity of the "postsynaptic" neuron "m". In other words, the synaptic weight (25) is high when the input signal x_n frequently takes part in excitation of the m-th neuron and it is low when the input signal x_n rarely contributes to the excitation of the m-th neuron. This formulation resembles the classical Hebb's postulate of learning (cf. Hebb (1949), p. 62):

"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic changes take place in one or both cells such that A's efficiency as one of the cells firing B, is increased."

Recall that the last term in (24) corresponds to the norming coefficient responsible for competitive properties of neurons. It can be interpreted as a cumulative effect of special neural structures performing lateral inhibition. This term is identical for all components of the underlying mixture and, for this reason,

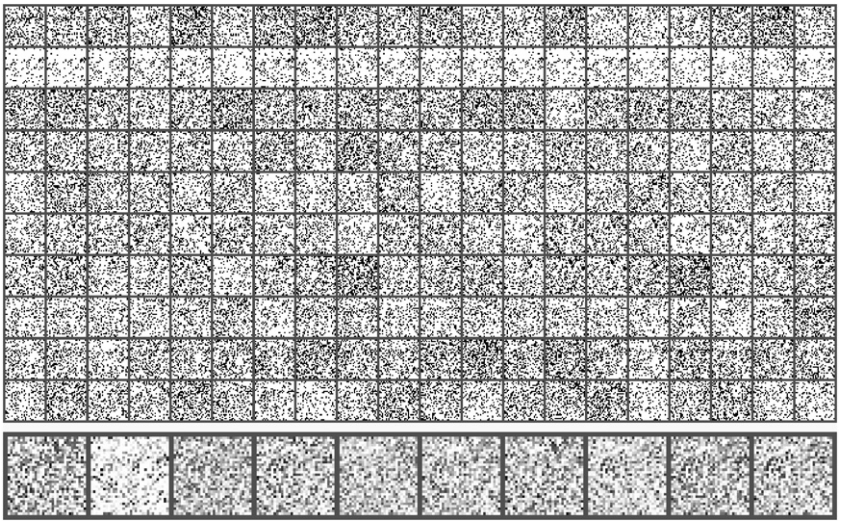

Fig. 4. Example of permutation of variables. The recognition accuracy based on the conditional distributions (21) is invariant with respect to the order of raster fields in the vector $\boldsymbol{x}$. Fig. 4 shows numerals and class-means from Fig. 1 for a fixed random permutation of raster fields. The permutated patterns can be recognized with the same accuracy.

the Bayesian decision making would not be influenced by its inaccuracy.

However, at the level of hidden-layer neurons, there is a problem of correspondence of lateral inhibition connections. In the case of an information preserving transform the lateral inhibition should exactly norm the output signals of neurons which constitute the transform. Unfortunately, such an exact correspondence is hardly possible in biological neural systems.

We recall in this connection the invariance of the information preserving transform with respect to weighting of data (Grim et al., 2002). If we assume that correspondence of neurons in the lateral inhibition structure is violated, then the resulting improper norming can be included in a weighting function:

$$\tilde{q}(m|\boldsymbol{x}) = \frac{G(\boldsymbol{x}|m,\phi_m)w_m}{\sum\limits_{j\in\tilde{\mathcal{M}}} G(\boldsymbol{x}|j,\phi_j)w_j} = \frac{\sum\limits_{j\in\mathcal{M}} G(\boldsymbol{x}|j,\phi_j)w_j}{\sum\limits_{j\in\tilde{\mathcal{M}}} G(\boldsymbol{x}|j,\phi_j)w_j} q(m|\boldsymbol{x}) = \tilde{\lambda}(\boldsymbol{x})q(m|\boldsymbol{x}).$$

Here $\tilde{\mathcal{M}}$ denotes the improper set of neurons. In view of the asymptotic invariance of EM-learning with respect to weighting, the final recognition would not be influenced by the latent weighting function $\tilde{\lambda}(\boldsymbol{x})$.

Finally, let us recall the invariance of the structural mixture model with respect to permutation of variables. In order to illustrate this property Fig. 4 shows the raster patterns of Fig. 1 reordered according to a fixed randomly chosen permutation. Obviously, by using permutated data of the type shown in Fig. 4, we would achieve the same recognition accuracy as in Table 1. In other words, the structural mixture model does not use any topological properties of the raster.

In view of this fact it can be understood that some misclassified numerals in Fig. 3 appear quite "readable". They may be incorrectly recognized e.g. because their position is unusual. On the other hand there is a good chance of achieving high recognition accuracy in the case of data which has no "natural" topological properties (e.g. binary results of medical tests). The topological invariance of PNNs is a rather essential neuromorphic feature since in biological neural networks (ascending pathways) the information about the topological arrangement of input layer neurons is not available at higher levels.

5. Iterative principles of recognition

Let us note that, in biological terms, the estimation of mixture parameters in Eq. (24) can be seen as a long-term process, the results of which reflect the global statistical properties of training data. We recall in this connection that, in the case of exactly estimated class-conditional distributions $P(\boldsymbol{x}|\omega)$, the Bayes rule (3) provides a minimum classification error, which can be reduced only by means of some additional external knowledge. Obviously, any change of mixture parameters may strongly affect the outputs of hidden layer neurons (24) and the Bayes probabilities (21) with the resulting unavoidable loss of decision-making optimality. In view of this fact the "static" nature of PNNs strongly contradicts the well known short-term dynamic processes in biological neural networks. The fixed parameters cannot play any role in the short-term synaptic plasticity or in the complex transient states of neural assemblies.

In this section we show that some parameters of PNNs can be "released" for the sake of short-term dynamic processes without adversely affecting the statistically correct decision making. In particular, by means of the recurrent use of Bayes formula, we can adapt the component weights to a specific data vector on input. On the other hand, by using similar theoretical background, the input pattern can be iteratively adapted to a more probable form in order to facilitate correct recognition.

5.1. Recurrent bayesian reasoning

Let us recall first that the unconditional distribution mixture $P(\boldsymbol{x})$ formally introduces an additional low-level "descriptive" decision problem (Grim, 1996a). In this sense the mixture components $F(\boldsymbol{x}|m)$ may correspond to some "elementary" properties or situations. Given a vector $\boldsymbol{x} \in \mathcal{X}$, the implicit presence of the elementary properties can be characterized by the conditional probabilities $q(m|\boldsymbol{x})$ which are related to the *a posteriori* probabilities of classes by (21).

In Eq. (20) the component weights w_m represent *a priori* knowledge of the descriptive decision problem. Given a particular input vector $\boldsymbol{x} \in \mathcal{X}$, the *a priori* weights w_m can be replaced by the more specific conditional weights $q(m|\boldsymbol{x})$. The idea can be summarized by the following recurrent Bayes formula

$$w_m^{(t+1)} = q^{(t)}(m|\boldsymbol{x}) = \frac{F(\boldsymbol{x}|m)w_m^{(t)}}{P^{(t)}(\boldsymbol{x})}, \tag{26}$$

$$P^{(t)}(\boldsymbol{x}) = \sum_{m\in\mathcal{M}} F(\boldsymbol{x}|m)w_m^{(t)}, \tag{27}$$

$$w_m^{(0)} = w_m, \quad m \in \mathcal{M},\ t = 0, 1, \ldots.$$

The recurrent computation of the conditional weights $q^{(t)}(m|\boldsymbol{x})$ resembles the natural process of cognition as iteratively improving understanding of input information. Also it is related to the original convergence proof of the EM algorithm proposed by Schlesinger (1968). We recall that Eq. (26) is a special case of the iterative inference mechanism originally proposed in probabilistic expert systems (Grim & Vejvalková, 1999). In a simple form restricted to two components, iterative weighting has been considered in pattern recognition, too (Baram, 1999).

It can be easily verified that the iterative procedure defined by (26) converges. In particular, we introduce a simple log-likelihood function corresponding to a single data vector $\boldsymbol{x} \in \mathcal{X}$:

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{x}) = \log P(\boldsymbol{x}) = \log\left[\sum_{m\in\mathcal{M}} F(\boldsymbol{x}|m)w_m\right]. \tag{28}$$

Considering the well known property of Kullback–Leibler information divergence

$$I(q^{(t)}(\cdot|\boldsymbol{x}) \parallel q^{(t+1)}(\cdot|\boldsymbol{x})) = \sum_{m\in\mathcal{M}} q^{(t)}(m|\boldsymbol{x}) \log \frac{q^{(t)}(m|\boldsymbol{x})}{q^{(t+1)}(m|\boldsymbol{x})} \geq 0 \tag{29}$$

we can write (cf. (26)) the following inequality

$$\sum_{m\in\mathcal{M}} q^{(t)}(m|\boldsymbol{x}) \log \frac{P^{(t+1)}(\boldsymbol{x})}{P^{(t)}(\boldsymbol{x})}$$

$$\geq \sum_{m\in\mathcal{M}} q^{(t)}(m|\boldsymbol{x}) \log\left[\frac{F(\boldsymbol{x}|m)w_m^{(t+1)}}{F(\boldsymbol{x}|m)w_m^{(t)}}\right] \tag{30}$$

which can be equivalently rewritten in the form

$$\log \frac{P^{(t+1)}(\boldsymbol{x})}{P^{(t)}(\boldsymbol{x})} \geq \sum_{m\in\mathcal{M}} q^{(t)}(m|\boldsymbol{x}) \log \frac{w_m^{(t+1)}}{w_m^{(t)}}. \tag{31}$$

Again, considering the substitution $w_m^{(t+1)} = q^{(t)}(m|\boldsymbol{x})$, we can see that the right-hand side of (31) is the non-negative Kullback–Leibler divergence and therefore the sequence of values $\{\mathcal{L}(\boldsymbol{w}^{(t)}, \boldsymbol{x})\}_{t=0}^{\infty}$ generated by the iterative Eq. (26) is nondecreasing.

It follows that the formula (26) defines the EM iteration equations to maximize (28) with respect to the component weights w_m. Moreover, the nondecreasing sequence $\{\mathcal{L}(\boldsymbol{w}^{(t)}, \boldsymbol{x})\}_{t=0}^{\infty}$ converges to a unique limit $\mathcal{L}(\boldsymbol{w}^*, \boldsymbol{x})$ since the criterion (28) is easily verified to be strictly concave as a function of $\boldsymbol{w}$ (for details cf. Grim and Vejvalková (1999)). Consequently, the limits of the component weights w_m^* are independent of the initial values $w_m^{(0)} = w_m$. Simultaneously, we remark that the log-likelihood function $\mathcal{L}(\boldsymbol{w}, \boldsymbol{x})$ achieves an obvious maximum by setting the weight of the maximum component function $F(\boldsymbol{x}|m_0)$ to one, i.e. for $w_{m_0} = 1$. For this reason the limit values w_m^* can be specified without computation as follows

$$m_0 = \arg\max_{m\in\mathcal{M}}\{F(\boldsymbol{x}|m)\}, \tag{32}$$

$$w_m^* = \begin{cases} 1, & m = m_0, \\ 0, & m \neq m_0, \end{cases}, \quad m \in \mathcal{M}. \tag{33}$$

We have verified the computational aspects of the recurrent Bayes formula (26) in the numerical experiments of Section 3. In order to illustrate the limited relevance of the initial values of component weights in (26) we have repeated all experiments with strongly manipulated weights. In particular, approximately one half of the weights (randomly chosen) has been almost suppressed by setting $w_m = 10^{-8}$ with the remaining weights being equal to $w_m = 10$ without any norming. The fifth row of the Table 1 shows the influence of the component weight manipulation. It is not surprising that the corresponding classification accuracy is much worse than that of the standard Bayes rule in all experiments. On the other hand, the sixth row shows how the "spoiled" weights can be repaired with the aid of iterative weighting (26). The achieved accuracy illustrates that the classification based on iterated weights is actually independent of the initial weights — in accordance with the theoretical conclusion.

For the extended test data set we have obtained analogous results as it can be seen in the last three rows of Table 1. Nevertheless, it is surprising that the consequences of manipulation of weights are less apparent than in the case of the non-extended test data set. It appears that the additional rotated variants of digit patterns succeed to correct the "spoiled" recognition accuracy almost entirely. Further, by using the iterated weights, we obtain recognition errors which are nearly equal to that of the standard Bayes rule.

The mechanism of recurrent use of the Bayes formula could intuitively seem to be suitable to explain the theoretical background of short-term synaptic plasticity. On the other hand the adaptively changing component weights w_m correspond more to the spontaneous activity of neurons (cf. Section 4), and the synaptic weights (25) seem to be responsible for the highly specific long-term statistical information to guarantee the final correct recognition. From this point of view any short-term adaptive changes of the synaptic weights appear to be rather unlikely unless they would be interrelated with the spontaneous activity of the respective neuron.

5.2. Adaptively modified input

Let us further note that the log-likelihood criterion (28) can also be maximized as a function of $\boldsymbol{x}$ by means of the EM algorithm in a similar way. We denote

$$q(m|\boldsymbol{x}^{(t)}) = \frac{F(\boldsymbol{x}^{(t)}|m)w_m}{P(\boldsymbol{x}^{(t)})}, \quad m \in \mathcal{M},\ \boldsymbol{x}^{(0)} = \boldsymbol{x},\ t = 0, 1, \ldots \tag{34}$$

and, in analogy with (30), (31) we can write

$$\log \frac{P(\boldsymbol{x}^{(t+1)})}{P(\boldsymbol{x}^{(t)})} \geq \sum_{m\in\mathcal{M}} q(m|\boldsymbol{x}^{(t)}) \log \frac{F(\boldsymbol{x}^{(t+1)}|m)}{F(\boldsymbol{x}^{(t)}|m)}. \tag{35}$$

Further, considering substitution (8), we obtain the following inequality

$$\log \frac{P(\boldsymbol{x}^{(t+1)})}{P(\boldsymbol{x}^{(t)})}$$

$$\geq \sum_{m\in\mathcal{M}} q(m|\boldsymbol{x}^{(t)}) \log \frac{F(\boldsymbol{x}^{(t+1)}|0)G(\boldsymbol{x}^{(t+1)}|m, \phi_m)}{F(\boldsymbol{x}^{(t)}|0)G(\boldsymbol{x}^{(t)}|m, \phi_m)} \tag{36}$$

which can be rewritten in the form

$$\log \frac{P(\boldsymbol{x}^{(t+1)})}{P(\boldsymbol{x}^{(t)})} \geq \sum_{n\in\mathcal{N}} (x_n^{(t+1)} - x_n^{(t)})Q_n^{(t)} \tag{37}$$

by using notation

$$Q_n^{(t)} = \log \frac{\theta_{0n}}{1-\theta_{0n}} + \sum_{m\in\mathcal{M}} \phi_{mn} q(m|\boldsymbol{x}^{(t)}) \log \frac{\theta_{mn}(1-\theta_{0n})}{\theta_{0n}(1-\theta_{mn})}. \tag{38}$$

Now, in order to guarantee the right-hand part of (37) to be non-negative, we define the new modified input data vector $\boldsymbol{x}^{(t+1)}$ by Eqs.:

$$x_n^{(t+1)} = \begin{cases} 1, & Q_n^{(t)} \geq 0, \\ 0, & Q_n^{(t)} < 0, \end{cases} \quad n \in \mathcal{N}, t = 0, 1, \ldots. \tag{39}$$

Consequently, starting with an initial value $\boldsymbol{x}^{(0)}$, the sequence of data vectors $\{\boldsymbol{x}^{(t)}\}_{t=0}^{\infty}$ maximizes the criterion $\mathcal{L}(\boldsymbol{w}, \boldsymbol{x})$ as a function of $\boldsymbol{x}$. In particular, the sequence $\{\mathcal{L}(\boldsymbol{w}, \boldsymbol{x}^{(t)})\}_{t=0}^{\infty}$ is nondecreasing

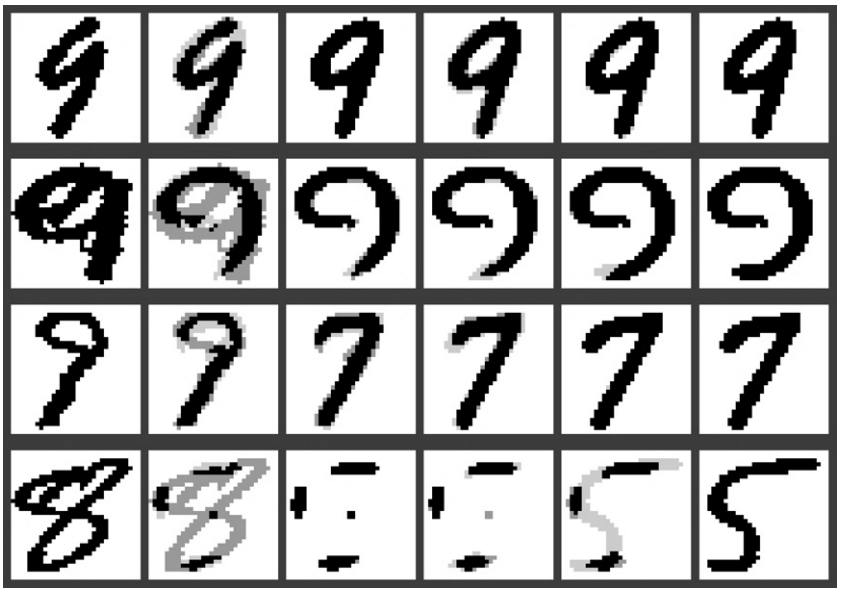

Fig. 5. Examples of iteratively modified input patterns. In most cases the modified input pattern $\boldsymbol{x}$ converges in several steps to a local extreme of $P(\boldsymbol{x})$, which corresponds to a more probable form of the numeral. However, some unusual patterns may be destroyed or modified incorrectly.

and converges to a local maximum of $P(\boldsymbol{x})$. In our experiments, the convergence of the iteration Eqs. (34), (38) and (39) has usually been achieved in a few steps — as illustrated by Fig. 5.

In other words, given an initial input vector $\boldsymbol{x}$, we obtain a sequence of data vectors $\boldsymbol{x}^{(t)}$ with increasing probability

$$P(\boldsymbol{x}^{(t)}) \leq P(\boldsymbol{x}^{(t+1)}), \quad t = 0, 1, \ldots \tag{40}$$

and therefore the modified vectors should be more easily classified. In most cases the modified input pattern adapts to a more typical form but some unusual patterns may be damaged or even destroyed by the adaptation procedure (cf. Fig. 5). As the adaptation procedure does not provide any new external knowledge, the classification accuracy of the modified input pattern is only comparable with the standard Bayes rule (cf. the seventh row of Table 1). In the case of the extended test data set this conclusion is still more obvious (cf. the last row of Table 1).

The considered adaptive modification of input patterns can be viewed as a theoretical model of the well known fact that the perception of visual stimuli by the human eye tends to be influenced by previously seen images.

6. Conclusion

Considering PNNs in the framework of statistical pattern recognition we obtain formal neurons strictly defined by means of parameters of the estimated class-conditional mixtures. A serious disadvantage of PNNs in this respect is the fixed probabilistic description of the underlying decision problem, which is not compatible with the well known short-term dynamic processes in biological neural networks. We have shown that, by considering the iterative procedures of recognition, some mixture parameters may take part in short term dynamic processes without disturbing the statistically correct decision making. In particular, the mixture component weights can be iteratively adapted to a specific input pattern or the input pattern can be iteratively modified in order to facilitate correct recognition.

Acknowledgements

This research was supported by the Czech Science Foundation project No. 102/07/1594 and partially by the projects 2C06019 ZIMOLEZ and 1M0572 DAR of the Czech Ministry of Education.

References

Baram, Y. (1999). Bayesian classification by iterated weighting. *Neurocomputing*, *25*, 73–79.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, *39*, 1–38.

Grim, J. (1982). On numerical evaluation of maximum — likelihood estimates for finite mixtures of distributions. *Kybernetika*, *18*(2), 173–190.

Grim, J. (1986). Multivariate statistical pattern recognition with non-reduced dimensionality. *Kybernetika*, *22*(3), 142–157.

Grim, J. (1996a). Maximum-likelihood design of layered neural networks. In *International conference on pattern recognition. Proceedings* (pp. 85–89). Los Alamitos: IEEE Computer Society Press.

Grim, J. (1996b). Design of multilayer neural networks by information preserving transforms. In E. Pessa, M. P. Penna, & A. Montesanto (Eds.), *Third European congress on systems science* (pp. 977–982). Roma: Edizioni Kappa.

Grim, J. (1999a). A sequential modification of EM algorithm. In W. Gaul, & H. Locarek-Junge (Eds.), *Studies in classification, data analysis and knowledge organization, classification in the information age* (pp. 163–170). Berlin: Springer.

Grim, J. (1999b). Information approach to structural optimization of probabilistic neural networks. In L. Ferrer, & A. Caselles (Eds.), *Fourth European congress on systems science* (pp. 527–539). Valencia: SESGE.

Grim, J. (2000). Self-organizing maps and probabilistic neural networks. *Neural Network World*, *10*, 407–415.

Grim, J., Haindl, M., Somol, P., & Pudil, P. (2006). A subspace approach to texture modelling by using Gaussian mixtures. In B. Haralick, & T. K. Ho (Eds.), *Proceedings of the 18th international conference on pattern recognition. ICPR 2006* (pp. 235–238). Los Alamitos: IEEE Computer Society.

Grim, J., & Hora, J. (2007). Recurrent bayesian reasoning in probabilistic neural networks. In Marques de Sá, et al., (Eds.), *Lecture notes in computer science*: *vol. 4669*. *Artificial neural networks — ICANN 2007* (pp. 129–138). Berlin: Springer.

Grim, J., Just, P., & Pudil, P. (2003). Strictly modular probabilistic neural networks for pattern recognition. *Neural Network World*, *13*, 599–615.

Grim, J., Kittler, J., Pudil, P., & Somol, P. (2000). Combining multiple classifiers in probabilistic neural networks. In Kittler J., & F. Roli (Eds.), *Lecture notes in computer science*: *vol. 1857*. *Multiple classifier systems* (pp. 157–166). Berlin: Springer.

Grim, J., Kittler, J., Pudil, P., & Somol, P. (2002). Multiple classifier fusion in probabilistic neural networks. *Pattern Analysis & Applications*, *5*, 221–233.

Grim, J., Pudil, P., & Somol, P. (2000). Recognition of handwritten numerals by structural probabilistic neural networks. In H. Bothe, & R. Rojas (Eds.), *Proceedings of the second ICSC symposium on neural computation* (pp. 528–534). Wetaskiwin: ICSC.

Grim, J., Pudil, P., & Somol, P. (2002). Boosting in probabilistic neural networks. In R. Kasturi, D. Laurendeau, & C. Suen (Eds.), *Proc. 16th international conference on pattern recognition* (pp. 136–139). Los Alamitos: IEEE Comp. Soc.

Grim, J., Somol, P., & Pudil, P. (2005). Probabilistic neural network playing and learning Tic-Tac-Toe. *Pattern Recognition Letters*, *26*(12), 1866–1873 [Special issue].

Grim, J., & Vejvalková, J. (1999). An iterative inference mechanism for the probabilistic expert system PES. *International Journal of General Systems*, *27*, 373–396.

Grother, P. J. (1995). NIST special database 19: Handprinted forms and characters database, Technical Report and CD ROM.
Haykin, S. (1993). *Neural networks: A comprehensive foundation*. San Mateo CA: Morgan Kaufman.
Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory*. New York: Wiley.
Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation*. New York, Menlo Park CA, Amsterdam: Addison-Wesley.
McLachlan, G. J., & Peel, D. (2000). *Finite mixture models*. New York, Toronto: John Wiley and Sons.
Oja, E. (1983). *Subspace methods of pattern recognition*. Letchworth, U.K.: Research Studies Press.
Oja, E. (1989). Neural networks, principal components and subspaces. *International Journal of Neural Systems*, *1*, 61–68.
Oja, E., & Kohonen, T. (1988). The subspace learning algorithm as a formalism for pattern recognition and neural networks. In *Proceeding 1988 IEEE International Conference on Neural Networks* (pp. 277–284).
Palm, H. Ch. (1994). A new method for generating statistical classifiers assuming linear mixtures of Gaussian densities. In *Proc. of the 12th IAPR international conference on pattern recognition*, Jerusalem, 1994, II. (pp. 483–486). Los Alamitos: IEEE Computer Soc. Press.
Prakash, M., & Murty, M. N. (1997). Growing subspace pattern recognition methods and their neural-network models. *IEEE Transactions on Neural Networks*, *8*, 161–168.
Schlesinger, M. I. (1968). Relation between learning and self-learning in pattern recognition. *Kibernetika, (Kiev)*, *6*, 81–88 [in Russian].
Specht, D. F. (1988). Probabilistic neural networks for classification, mapping or associative memory. In: *Proc. of the IEEE international conference on neural networks*, I, (pp. 525–532).
Streit, L. R., & Luginbuhl, T. E. (1994). Maximum-likelihood training of probabilistic neural networks. *IEEE Transactions on Neural Networks*, *5*, 764–783.
Vajda, I., & Grim, J. (1998). About the maximum information and maximum likelihood principles in neural networks. *Kybernetika*, *34*, 485–494.
Watanabe, S. (1967). Karhunen-Loeve expansion and factor analysis. In *Trans. of the fourth Prague conf. on information theory* (pp. 635–660). Prague: Academia.
Watanabe, S., & Fukumizu, K. (1995). Probabilistic design of layered neural networks based on their unified framework. *IEEE Transactions on Neural Networks*, *6*(3), 691–702.
Watanabe, S., & Pakvasa, N. (1973). Subspace method in pattern recognition. In *Proc. int. joint conf. on pattern recognition* (pp. 25–32).
Workshop: Subspace 2007. Workshop on ACCV2007. Tokyo, Japan, Nov. 19, 2007 http://www.viplab.is.tsukuba.ac.jp/ ss2007/downloadsite.html.