On Virtually Binary Nature of Probabilistic Neural Networks

Jiří Grim and Pavel Pudil

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
P.O. Box 18, CZ-18208 Prague 8, Czech Republic

Abstract. A sequential design of multilayer probabilistic neural networks is considered in the framework of statistical decision-making. Parameters and interconnection structure are optimized layer-by-layer by estimating unknown probability distributions on input space in the form of finite distribution mixtures. The components of mixtures correspond to neurons which perform an information preserving transform between consecutive layers. Simultaneously the entropy of the transformed distribution is minimized. It is argued that in multidimensional spaces and particularly at higher levels of multilayer feedforward neural networks, the output variables of probabilistic neurons tend to be binary. It is shown that the information loss caused by the binary approximation of neurons can be suppressed by increasing the approximation accuracy.

1 Introduction

The possibility of a general maximum-likelihood (m.-l.) design of probabilistic neural networks based on distribution mixtures has received increasing attention in recent literature (cf. Specht [11], Streit and Luginbuhl [12], Bishop [1], Palm [9]). Considering the framework of statistical decision-making (cf. Grim [3] [5] [6], Vajda and Grim [13]) we assume that the components of mixtures may be shared by all class-conditional distributions to avoid possible structural limitations. Numerically the method is based on m.-l. estimation of finite mixtures by means of EM algorithm (cf. Dempster *et al.* [2], Grim [3], Schlesinger [10]).

The component distributions corresponding to neurons naturally define an additional "descriptive" decision problem and simultaneously each neuron can be viewed as a coordinate function of a vector transform mapping the input space into the space of output variables. It has been shown (cf. Grim [6], Vajda and Grim [13]) that, under very general conditions, the descriptive decision problem can be transformed without information loss by means of coordinate functions based on a posteriori probabilities of components. Simultaneously the transform

[0] Suported by the Grant No. A2075703 of the Academy of Sciences, by the Grant No. 402/97/1242 of the Czech Grant Agency and by the Grant of the Ministry of Education No. VS 96063 of the Czech Republic.

minimizes entropy of the output space. In this way we can design multilayer neural networks by estimating the mixture parameters layer-by-layer.

A typical feature of multilayer probabilistic neural networks obtained in this way is a biologically unnatural complete interconnection between neurons of consecutive layers. To avoid this undesirable property we proposed specially modified finite mixtures with factorizable components including structural parameters (cf. Grim [7]). Again the parameters of incompletely interconnected structures and the structure itself can be optimized by means of EM algorithm.

In the present paper we show that the information preserving minimum-entropy transform tends to produce features of low statistical complexity. As opposed to standard feature selection methods of pattern recognition our approach emphasizes simplicity of features rather than reducing their number. We make use of the well known empirical experience that, with increasing dimension, the input space becomes "sparse" and, simultaneously, the multivariate componets of finite mixtures become increasingly nonoverlapping. Consequently, the a posteriori probabilities of components can be looked upon almost as binary variables.

It is shown that binary approximation of the information preserving transform essentially simplifies the underlying probabilistic neural network and the related optimization problem. The computation of neuron responses amounts to competitive evaluation of simple linear expressions. We derive bounds of the information loss caused by a binary approximation and show that the information loss approaches zero with increasing approximation accuracy. The reliability of the proposed scheme can be improved by considering multiply parallel solutions for each layer.

2 Probabilistic Neural Networks

We consider a finite set of classes $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ with a priori probabilities $p(\omega)$ and the corresponding class-conditional multivariate probability distributions $P(\boldsymbol{x}|\omega)$ on a discrete space $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_N$. The unconditional joint probability distribution $P(\boldsymbol{x})$ is given by

$$P(\boldsymbol{x}) = \sum_{\omega \in \Omega} P(\boldsymbol{x}|\omega)p(\omega), \quad \boldsymbol{x} = (x_1, x_2, \ldots, x_N) \in \mathcal{X}. \tag{1}$$

All statistical information about the set of classes Ω, given some observation $\boldsymbol{x} \in \mathcal{X}$, is expressed by the a posteriori probabilities

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

which can be used to define the final decision function.

Further we assume that the conditional distributions $P(\boldsymbol{x}|\omega)$ can be approximated by finite mixtures of the form

$$P(\boldsymbol{x}|\omega) = \sum_{m \in \mathcal{M}} F(\boldsymbol{x}|m)f(m|\omega), \quad \omega \in \Omega, \quad \boldsymbol{x} \in \mathcal{X}, \quad (\sum_{m \in \mathcal{M}} f(m|\omega) = 1) \tag{3}$$

where $f(m|\omega) \geq 0$ are some conditional probabilistic weights. The components $F(\cdot|m) \in \mathcal{F}$ belong to a common pool of probability distributions on $\mathcal{X}$

$$\mathcal{F} = \{F(\cdot|m), m \in \mathcal{M}\}, \quad \mathcal{M} = \{1, 2, \ldots, M\}. \tag{4}$$

which may be shared by all classes $\omega \in \Omega$.

Making substitution (3) in Eq.(1) we can write

$$P(\boldsymbol{x}) = \sum_{m \in \mathcal{M}} F(\boldsymbol{x}|m) f(m), \quad \boldsymbol{x} \in \mathcal{X} \tag{5}$$

whereby

$$f(m) = \sum_{\omega \in \Omega} f(m|\omega) p(\omega), \quad f(m|\boldsymbol{x}) = \frac{F(\boldsymbol{x}|m) f(m)}{P(\boldsymbol{x})}. \tag{6}$$

Thus the concept of shared components [7], [1] naturally introduces an additional "descriptive" decision problem characterized by the component distributions $F(.|m) \in \mathcal{F}$ and the unconditional a priori weights $f(m), m \in \mathcal{M}$. Note that all parameters of the mixture (5) can be estimated from data in unsupervised way by means of EM algorithm (cf. Schlesinger [10], Depmster *et al.* [2], Grim [3]).

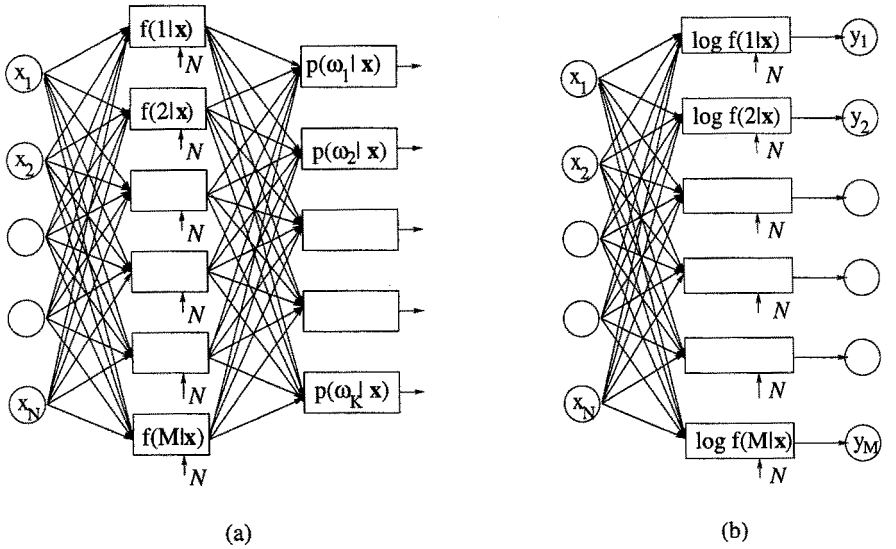

Fig. 1. a) Probabilistic neural network b) Hidden layer of neuron network for information preserving transform

Using this notation we can express the finally desired a posteriori probabilities $p(\omega|\boldsymbol{x})$ as linear combinations of $f(m|\boldsymbol{x})$ (cf. Grim [7], Bishop [1]):

$$p(\omega|\boldsymbol{x}) = \sum_{m \in \mathcal{M}} p(\omega|m) f(m|\boldsymbol{x}), \quad p(\omega|m) = \frac{f(m|\omega) p(\omega)}{f(m)}, \quad \omega \in \Omega, \quad \boldsymbol{x} \in \mathcal{X}. \tag{7}$$

The last Eqs. can be interpreted as a three-layer neural network (cf. Fig. 1(a)): the first layer of input variables x_n, the second "hidden" layer of components computing the normed outputs $f(m|\boldsymbol{x})$ (the norming is only symbolically indicated by an arrow and the letter N), and the third layer of linear units computing the aposteriori probabilities $p(\omega|\boldsymbol{x})$ as a weighted sum of $f(m|\boldsymbol{x})$.

Let us recall that, as the functions $F(\boldsymbol{x}|m) \in \mathcal{F}$ may be shared by all class-conditional distributions $P(.|\omega)$, the units of the second layer are not "specialized" and therefore the outputs of the second layer are not confined to particular classes (cf. Grim [5]). In this sense the structure of the resulting neural network is not restricted, in accordance with the basic structural properties of ascending neural pathways.

From a functional point of view each neuron of a given layer realizes a coordinate function of a vector transform $\mathbf{T}$ mapping the input space $\mathcal{X}$ into the space of output variables $\mathcal{Y}$. We denote

$$\mathbf{T} : \mathcal{X} \to \mathcal{Y}, \quad \mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_M \subset R^M,$$

$$\boldsymbol{y} = \mathbf{T}(\boldsymbol{x}) = (T_1(\boldsymbol{x}), T_2(\boldsymbol{x}), \ldots, T_M(\boldsymbol{x})) \in \mathcal{Y}. \tag{8}$$

It has been shown (Grim [6], Vajda and Grim [13]) that the transform defined by Eqs.

$$y_m = T_m(\boldsymbol{x}) = \log f(m|\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X}, \quad m \in \mathcal{M} \tag{9}$$

belongs to a class of information preserving transforms minimizing the entropy of the output space $\mathcal{Y}$ (cf. Fig. 1(b)).

If we introduce the following notation for the transformed distributions and a posteriori probabilities

$$Q(\boldsymbol{y}) = P(\mathbf{T}^{-1}(\boldsymbol{y})), \quad Q(\boldsymbol{y}|m) = F(\mathbf{T}^{-1}(\boldsymbol{y})|m), \quad \boldsymbol{y} \in \mathcal{Y} \tag{10}$$

$$\mathbf{T}^{-1}(\boldsymbol{y}) = \{\boldsymbol{x} \in \mathcal{X} : \mathbf{T}(\boldsymbol{x}) = \boldsymbol{y}\}, \quad q(m|\boldsymbol{y}) = \frac{Q(\boldsymbol{y}|m)f(m)}{Q(\boldsymbol{y})}, \quad m \in \mathcal{M} \tag{11}$$

and for the related uconditional and conditional Shannon entropies

$$H(\mathcal{M}) = \sum_{m \in \mathcal{M}} -f(m) \log f(m), \quad H(\mathcal{Y}) = \sum_{y \in \mathcal{Y}} -Q(\boldsymbol{y}) \log Q(\boldsymbol{y}), \tag{12}$$

$$H(\mathcal{M}|\mathcal{X}) = \sum_{x \in \mathcal{X}} P(\boldsymbol{x}) H_{\boldsymbol{x}}(\mathcal{M}), \quad H_{\boldsymbol{x}}(\mathcal{M}) = \sum_{m \in \mathcal{M}} -f(m|\boldsymbol{x}) \log f(m|\boldsymbol{x}), \tag{13}$$

$$H(\mathcal{M}|\mathcal{Y}) = \sum_{y \in \mathcal{Y}} Q(\boldsymbol{y}) H_{\boldsymbol{y}}(\mathcal{M}), \quad H_{\boldsymbol{y}}(\mathcal{M}) = \sum_{m \in \mathcal{M}} -q(m|\boldsymbol{y}) \log q(m|\boldsymbol{y}) \tag{14}$$

then we can write

$$I(\mathcal{X}, \mathcal{M}) = H(\mathcal{M}) - H(\mathcal{M}|\mathcal{X}) = H(\mathcal{M}) - H(\mathcal{M}|\mathcal{Y}) = I(\mathcal{Y}, \mathcal{M}), \tag{15}$$

i.e. the transform $\mathbf{T}$ preserves the statistical Shannon information about the descriptive decision problem $\{\mathcal{X}, F(.|m)f(m), m \in \mathcal{M}\}$. Simultaneously, the information preserving transform (8), (9) minimizes the entropy $H(\mathcal{Y})$ of the output

distribution $Q(\boldsymbol{y})$. It has been shown [6] that analogous assertions hold for the original decision problem $\{\mathcal{X}, P(.|\omega)p(\omega), \omega \in \Omega\}$, too.

Obviously, the procedure can be used to design multilayer neural networks by transforming the vector of input variables $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ to some output space $\mathcal{Y}$ repeatedly. Let us recall that at the highest layer of the probabilistic neural network we need labelled data for supervised estimation of the conditional weights $f(m|\omega)$ (cf. Eq. (7), Fig. 1(a)).

3 Maximum-Likelihood Structuring

A typical feature of the above multilayer feed-forward probabilistic neural networks is a biologically unnatural complete interconnection between units (neurons) of consecutive layers. In order to avoid this undesirable property we proposed specially modified components of the approximating mixtures [7]. The approach is based on an idea originally applied to multivariate pattern recognition [4].

Making substitution

$$F(\boldsymbol{x}|m) = F(\boldsymbol{x}|0)G(\boldsymbol{x}|m, \phi_m), \quad F(\boldsymbol{x}|0) = \prod_{n \in \mathcal{N}} f_n(x_n|0) \tag{16}$$

we obtain finite mixture of the form

$$P(\boldsymbol{x}) = \sum_{m \in \mathcal{M}} F(\boldsymbol{x}|0)G(\boldsymbol{x}|m, \phi_m)f(m) \tag{17}$$

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

$$G(\boldsymbol{x}|m, \phi_m) = \prod_{n \in \mathcal{N}} \left[\frac{f_n(x_n|m)}{f_n(x_n|0)}\right]^{\phi_{mn}}, \quad \phi_m = (\phi_{m1}, \ldots, \phi_{mN}) \in \{0, 1\}^N \tag{18}$$

and $F(\boldsymbol{x}|0)$ is a nonzero "background" probability distribution usually defined as a product of marginals, i.e. $f_n(x_n|0) = P_n(x_n)$.

It can be seen that any component-specific distribution $f_n(x_n|m)$ can be substituted by the respective univariate background distribution $f_n(x_n|0)$ by setting $\phi_{mn} = 0$, i.e.

$$F(\boldsymbol{x}|m) = \prod_{n \in \mathcal{N}} f_n(x_n|m)^{\phi_{mn}} f_n(x_n|0)^{1-\phi_{mn}}, \quad \boldsymbol{x} \in \mathcal{X}. \tag{19}$$

As the background distribution $F(\boldsymbol{x}|0)$ can be cancelled in the Bayes formula (6)

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

the computation of the a posteriori probabilities $f(m|\boldsymbol{x})$ may be confined only to some "relevant" variables without making any approximations or heuristic steps.

Let us note that using logarithmic coordinate function we can write (cf. (20))

$$y_m = \mathrm{T}_m(\boldsymbol{x}) = \log f(m|\boldsymbol{x}) =$$

$$= \log[G(\boldsymbol{x}|m, \phi_m) f(m)] - \log[\sum_{j \in \mathcal{M}} G(\boldsymbol{x}|j, \phi_j) f(j)], \quad m \in \mathcal{M} \tag{21}$$

and further, making substitution (18), we obtain

$$y_m = \log f(m) + \sum_{n \in \mathcal{N}} \phi_{mn} \log \frac{f_n(x_n|m)}{f_n(x_n|0)} - \log[\sum_{j \in \mathcal{M}} G(\boldsymbol{x}|j, \phi_j) f(j)]. \tag{22}$$

Thus, in view of the formula (22), the input connections of a single neuron can be confined to any subset of variables (input neurons) by means of the binary parameters ϕ_{mn} - in a statistically correct way.

The "incomplete interconnection" model (22) can be optimized by using maximum-likelihood criterion. The m.-l. estimates of all involved parameters, inclusive of the structural parameters ϕ_{mn}, can be computed by means of the iterative EM algorithm (cf. Grim [7]).

4 Binary Approximation

It is a well known practical experience that high-dimensional spaces are rather "sparse" and for this reason multivariate component distributions $F(\boldsymbol{x}|m)$ will tend to have only small "overlap" with the increasing dimensionality. For the same reason the a posteriori probabilities $f(m|\boldsymbol{x})$ (20) will tend to take only the extreme values 0 and 1.

On the other hand, the "soft-binary" properties of the conditional probabilities $f(m|\boldsymbol{x})$ have only empirical validity. Obviously, in any high-dimensional space the componnets $F(\boldsymbol{x}|m)$ may be defined sufficiently "flat" to increase their overlap and, consequently, to supress the tendency to binary values of $f(m|\boldsymbol{x})$.

Nevertheless, in multilayer probabilistic neural networks the statistical decision problem is defined on input space of the first layer and the dimension of the transformed spaces at higher levels can be chosen as high as necessary. Thus the acuracy of the assumed binary approximation can be influenced by a proper choice of parameters.

Motivated by the above arguments we shall assume in the following that the dimension of the input space $\mathcal{X}$ (related to the output variables of the immediately preceeding layer) is very high and the a posteriori probabilities $f(m|\boldsymbol{x})$ computed for the corresponding distribution mixture behave like binary variables. In other words we assume only small overlap of the components of the mixture $P(\boldsymbol{x})$ and approximate the coordinate functions (9) by two extreme output values

$$y_m = \mathrm{T}_m(\boldsymbol{x}) = \begin{cases} \eta_1, f(m|\boldsymbol{x}) \to 1_- \\ \eta_0, f(m|\boldsymbol{x}) \to 0_+ \end{cases}, \quad \boldsymbol{x} \in \mathcal{X}, \quad m \in \mathcal{M}. \tag{23}$$

Let us recall that the "synaptic weights" in the formula (22)

$$y_m \approx \sum_{n \in \mathcal{N}} \phi_{mn} \log \frac{f_n(x_n|m)}{f_n(x_n|0)}, \quad m \in \mathcal{M} \tag{24}$$

depend on the probabilities $f_n(x_n|m)$ and not directly on the variables x_n. Therefore, the input values $x_n \in \mathcal{X}_n$ themselves cannot directly influence the output value y_m. For this reason the true values η_0, η_1 approximating the real output function $T_m(\boldsymbol{x})$ are irrelevant and, for the sake of simplicity, we may define e.g.

$$y_m = T_m(\boldsymbol{x}) = \begin{cases} 1, & m = \mu(\boldsymbol{x}) \\ 0, & m \neq \mu(\boldsymbol{x}) \end{cases}, \quad \boldsymbol{x} \in \mathcal{X}, \quad m \in \mathcal{M}, \tag{25}$$

where $\mu(\boldsymbol{x})$ identifies the highest aposteriori probability $f(m|\boldsymbol{x})$. To avoid possible ties, we define

$$\mu(\boldsymbol{x}) = \min\{\arg \max_{m \in \mathcal{M}} \{f(m|\boldsymbol{x})\}\}, \quad \boldsymbol{x} \in \mathcal{X} \tag{26}$$

or more simply (cf. (20))

$$\mu(\boldsymbol{x}) = \min\{\arg \max_{m \in \mathcal{M}} \{\log[G(\boldsymbol{x}|m) f(m)]\}\}. \tag{27}$$

Consequently, we may ignore the norming term occurring in the formula (21) or (22):

$$\mu(\boldsymbol{x}) = \min\{\arg \max_{m \in \mathcal{M}} \{\log f(m) + \sum_{n \in \mathcal{N}} \phi_{mn} \log \frac{f_n(x_n|m)}{f_n(x_n|0)}\}\} \tag{28}$$

Note that the last formula corresponds well with the competitive principle (winner-takes-all rule) frequently applied to neural networks.

Binary approximation of the coordinate functions $T_m(\boldsymbol{x})$ implies a binary output space and, for this reason, we may assume the input space $\mathcal{X}$ to be binary too at higher levels, i.e. $\mathcal{X} = \{0,1\}^N$.

The binary input space simplifies the underlying probabilistic model essentially. Denoting

$$\theta_{nm} = f_n(1|m), \quad n \in \mathcal{N}, \quad m \in \mathcal{M} \tag{29}$$

we can write

$$f_n(x_n|m) = \theta^{x_n}(1-\theta)^{1-x_n}, \quad x_n \in \mathcal{X}_n = \{0,1\} \tag{30}$$

and further

$$\log \frac{f_n(x_n|m)}{f_n(x_n|0)} = \log\left(\frac{1-\theta_{nm}}{1-\theta_{n0}}\right) + x_n \log\left(\frac{\theta_{nm}(1-\theta_{nm})}{\theta_{n0}(1-\theta_{n0})}\right). \tag{31}$$

Making substitution (31) in (28) and denoting

$$\alpha_{nm} = \log\left(\frac{1-\theta_{nm}}{1-\theta_{n0}}\right), \quad \beta_{nm} = \log\left(\frac{\theta_{nm}(1-\theta_{nm})}{\theta_{n0}(1-\theta_{n0})}\right) \tag{32}$$

we can write

$$\mu(\boldsymbol{x}) = \min\{\arg\max_{m\in\mathcal{M}}\{\log f(m) + \sum_{n\in\mathcal{N}} \phi_{mn}\alpha_{nm} + \sum_{n\in\mathcal{N}} \phi_{mn}x_n\beta_{nm}\}\}. \quad (33)$$

Thus, as it can be seen, the functioning of neurons can be reduced to competitive evaluation of simple linear expressions.

Let us recall also that the a posteriori probabilities $f(m|\boldsymbol{x})$ have to be repeatedly evaluated in the iterative EM algorithm [5]. In view of the binary assumption (25) the iterative equations of EM algorithm can be simplified, too.

5 Bounds of Information Loss

In order to clarify possible consequences of the inaccuracy of the considered binary approximation we derive bounds for information loss arising when the binary coordinate function (25) is used instead of the original information preserving transform (21).

First we define the partition of the input space $\mathcal{X}$ induced by the function $\mu(\boldsymbol{x})$, (cf. (27), (33)):

$$\mathcal{S} = \{S_1, S_2, \ldots, S_M\}, \quad S_m = \{\boldsymbol{x} \in \mathcal{X} : \mu(\boldsymbol{x}) = m\}. \quad (34)$$

In view of the definition (25) we can write

$$y_m = T_m(\boldsymbol{x}) = \phi(\boldsymbol{x}|S_m) = \begin{cases} 1, \boldsymbol{x} \in S_m \\ 0, \boldsymbol{x} \notin S_m \end{cases}, \quad \boldsymbol{x} \in \mathcal{X}, \quad m \in \mathcal{M} \quad (35)$$

where $\phi(\cdot|S_m)$ is the characteristic function of the set $S_m \subset \mathcal{X}$. Introducing notation

$$\boldsymbol{y}^{(m)} = (\delta_{1m}, \delta_{2m}, \ldots, \delta_{Mm}) \in \mathcal{Y}; \quad \mathcal{Y}_{\mathcal{M}} = \{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(M)}\} \subset \mathcal{Y} \quad (36)$$

we can see that the set $\mathcal{Y}_{\mathcal{M}}$ contains all nonzero points of the transformed distribution Q, i.e.

$$\boldsymbol{y} \in (\mathcal{Y} - \mathcal{Y}_{\mathcal{M}}) \Rightarrow Q(\boldsymbol{y}) = 0 \quad (37)$$

and further (cf. (10))

$$\mathbf{T}^{-1}(\boldsymbol{y}^{(m)}) = S_m, \quad Q(\boldsymbol{y}^{(m)}) = P(S_m); \quad Q(\boldsymbol{y}^{(m)}|j) = F(S_m|j). \quad (38)$$

By using the last substitutions we can rewrite Eq. (11) as follows

$$q(j|\boldsymbol{y}^{(m)}) = \frac{F(S_m|j)f(j)}{P(S_m)} = \sum_{x\in S_m} f(j|\boldsymbol{x})\frac{P(\boldsymbol{x})}{P(S_m)} = f(j|S_m), \quad j, m \in \mathcal{M}. \quad (39)$$

To prove the following Theorem we also use repeatedly the inequality

$$\mathcal{H}(p_1, p_2, \ldots, p_M) = \sum_{j=1}^{M} -p_j \log p_j \leq \mathcal{H}(p_m, 1-p_m) + (1-p_m)\log(M-1) \quad (40)$$

which is true for any $m \in \mathcal{M}$. Here $\mathcal{H}(p_1, p_2, \ldots, p_M)$ denotes the Shannon entropy of the listed probabilities p_j .

Theorem 1.
Let the information preserving transform (21) of the descriptive decision problem $\{\mathcal{X}, F(\cdot|m)f(m), m \in \mathcal{M}\}$ be approximated by the binary coordinate functions (25). Further let δ and ϵ be small positive numbers. If we define

$$\bar{S}_m = \{\boldsymbol{x} \in S_m : f(m|\boldsymbol{x}) \geq 1 - \delta\}, \quad \tilde{S}_m = S_m - \bar{S}_m, \quad \tilde{\mathcal{X}} = \cup_{m \in \mathcal{M}} \tilde{S}_m \tag{41}$$

and assume that it holds

$$\frac{P(\tilde{S}_m)}{P(S_m)} < \epsilon, \quad m \in \mathcal{M} \tag{42}$$

then the information loss caused by the binary approximation (25) is bounded by the inequality

$$I(\mathcal{X}, \mathcal{M}) - I(\mathcal{Y}, \mathcal{M}) < \mathcal{H}(\epsilon + \delta; 1 - \epsilon - \delta) + (\epsilon + \delta)\log(M - 1). \tag{43}$$

Proof.
Note first that it holds (cf. (39))

$$f(j|S_m) = \frac{P(\bar{S}_m)}{P(S_m)} f(j|\bar{S}_m) + \frac{P(\tilde{S}_m)}{P(S_m)} f(j|\tilde{S}_m) \quad j, m \in \mathcal{M}. \tag{44}$$

Applying the following property of the sets $\bar{S}_m$

$$\boldsymbol{x} \in \bar{S}_m \;\Rightarrow\; f(m|\boldsymbol{x}) \geq 1 - \delta \;\Rightarrow\; f(m|\bar{S}_m) \geq 1 - \delta. \tag{45}$$

to Eq. (44) and considering the inequality (42) we obtain

$$f(m|S_m) \geq (1 - \frac{P(\tilde{S}_m)}{P(S_m)})(1 - \delta) \geq 1 - \frac{P(\tilde{S}_m)}{P(S_m)} - \delta > 1 - \epsilon - \delta. \tag{46}$$

Further, in view of the inequality (40), we can write

$$H_{S_m}(\mathcal{M}) < \mathcal{H}(\epsilon + \delta; 1 - \epsilon - \delta) + (\epsilon + \delta)\log(M - 1) \tag{47}$$

and finally

$$H(\mathcal{M}|\mathcal{Y}) = \sum_{m \in \mathcal{M}} P(S_m) H_{S_m}(\mathcal{M}) < \mathcal{H}(\epsilon + \delta; 1 - \epsilon - \delta) + (\epsilon + \delta)\log(M - 1). \tag{48}$$

The last inequality completes the proof since

$$I(\mathcal{X}, \mathcal{M}) - I(\mathcal{Y}, \mathcal{M}) = H(\mathcal{M}|\mathcal{Y}) - H(\mathcal{M}|\mathcal{X}) < H(\mathcal{M}|\mathcal{Y}) <$$
$$< \mathcal{H}(\epsilon + \delta; 1 - \epsilon - \delta) + (\epsilon + \delta)\log(M - 1). \tag{49}$$

It can be seen that, as it could be expected, for $\epsilon \to 0$ and $\delta \to 0$ the information loss approaches zero.

6 Concluding Remarks

Let us recall, that starting with a rather complicated probabilistic approach, we obtain finally a simple binary model of neural network. As it follows from the Eq. (33), the functioning of neurons can be reduced to competitive evaluation of simple linear expressions. This fact alone suggests the possibility to apply at each layer multiple solutions which are usually produced by standard use of EM algorithm. Repeated application of EM algorithm is desirable to avoid possible locally optimal solutions. However, on the other hand, as the considered type of mixtures (17) is not identifiable, we may expect many different solutions of comparable quality. This circumstance could be particularly useful from the point of view of combining multiple solutions. Some experiments concerning these aspects of the present paper will be the subject of a forthcoming paper.

References

1. Bishop C.M.: Neural Networks for Pattern Recognition. Clarendon Press: Oxford, (1995)
2. Dempster A.P., Laird N.M., Rubin D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc. **B 39** (1977) 1 – 38
3. Grim J.: On numerical evaluation of maximum likelihood estimates for finite mixtures of distributions. Kybernetika **18** (1982) 173-190
4. Grim J.: Multivariate statistical pattern recognition with nonreduced dimensionality. Kybernetika **22** (1986) 142-157
5. Grim J.: Maximum Likelihood Design of Layered Neural Networks. Proceedings of the 13th International Conference on Pattern Recognition **IV**, Los Alamitos: IEEE Computer Society Press (1996) 85 - 89
6. Grim J.: Design of multilayer neural networks by information preserving transforms. Proceedings of the Third European Congress on System Science, Eds. E. Pessa, M.P. Penna, A. Montesanto, Roma: Edizzioni Kappa (1996) 977-982
7. Grim J.: Maximum-Likelihood Structuring of Probabilistic Neural Networks. Research Report UTIA, AS CR, No. 1894, (1997)
8. Haykin S.: Neural Networks: a comprehensive foundation. Morgan Kaufman: San Mateo CA, (1993)
9. Palm H.Ch.: A new method for generating statistical classifiers assuming linear mixtures of Gaussian densities. Proceedings of the 12th IAPR International Conference on Pattern Recognition, Jerusalem **II**, Los Alamitos: IEEE Computer Society Press (1994) 483 – 486
10. Schlesinger M.I.: Relation between learning and self-learning in pattern recognition (in Russian). Kibernetika (Kiev), No. 2 (1968) 81 – 88
11. Specht D.F.: Probabilistic neural networks for classification, mapping or associative memory. Proceeding of the IEEE Int. Conference on Neural Networks, July 1988, **I** (1988) 525 – 532
12. Streit L.R., Luginbuhl T.E.: Maximum likelihood training of probabilistic neural networks. IEEE Trans. on Neural Networks **5** (1994) 764-783
13. Vajda I., Grim J.: About optimality of probabilistic basis function neural networks. Research Report UTIA, AS CR, No. 1887, (1996)