Appl. Math. and Comp. Sci., 1994, vol.4, No.3, 413–422

ON PATTERN CLASSIFICATION AND SYSTEM IDENTIFICATION BY PROBABILISTIC NEURAL NETWORKS

LESZEK RUTKOWSKI*, TOMASZ GAŁKOWSKI*

In the paper probabilistic neural networks are discussed in detail. Neural network structures for non-parametric pattern classification and system identification are developed.

1. Introduction

If the process is characterized by the absence of *a priori* information, the most popular methodology for identification and pattern classification is based on non-parametric approach. Such techniques — derived from non-parametric estimates of probability density and regression functions — have been developed by many authors to classify and identify different types of systems (see e.g. Gałkowski and Rutkowski, 1985; 1986; Rutkowski, 1988; 1991; 1993; Rutkowski and Rafajłowicz, 1989). The purpose of this article is to propose neural network structures for implementation of non-parametric algorithms. We shall define a neural network structure as a collection of parallel processors connected together in the form of a directed graph, so organized that the network structure corresponds to the non-parametric problem being considered. We shall use the so-called probabilistic neural networks (Specht, 1990). It should be emphasized that our neural networks do not require learning phase and are asymptotically optimal.

2. Density Estimation

Let $\boldsymbol{X}_1, ..., \boldsymbol{X}_n$ be a sequence of independent, identically distributed random variables taking values in $\mathbb{R}^d$ and having a probability density function f. The Parzen-Rosenblatt estimate of f is given by the formula

$$\widehat{f}_n(\boldsymbol{x}) = \frac{1}{nh_n^d} \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{X}_i}{h_n}\right) \tag{1}$$

where K is an appriopriately choosen function fulfilling the conditions:

$$\sup_{\boldsymbol{y}} |K(\boldsymbol{y})| < \infty \tag{2}$$

$$\int_{\mathbb{R}^d} |K(\boldsymbol{y})| \mathrm{d}\boldsymbol{y} < \infty \tag{3}$$

* Institute of Electronics and Control Systems, Technical University of Częstochowa, 42–200 Częstochowa, Poland

$$\lim_{\|y\|\to\infty} \|\boldsymbol{y}\|^d |K(\boldsymbol{y})| = 0 \tag{4}$$

$$\int_{\mathbb{R}^d} K(\boldsymbol{y})\mathrm{d}\boldsymbol{y} = 1 \tag{5}$$

In eqn. (4) symbol $\|\cdot\|$ stands for the Euclidean vector norm. Sequence h_n is a function of n and satisfies the conditions

$$\lim_{n\to\infty} h_n = 0 \qquad \text{and} \qquad \lim_{n\to\infty} nh_n^d = \infty \tag{6}$$

We assume that the function K is of the form

$$K(\boldsymbol{x}) = (2\pi)^{-\frac{1}{2}d} e^{-\frac{1}{2}\|x\|^2} \tag{7}$$

where $\|\boldsymbol{x}\|^2 = \boldsymbol{x}^T\boldsymbol{x}$. Then we can rewrite estimator (1) as follows:

$$\widehat{f}_n(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}\, nh_n^d} \sum_{i=1}^{n} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{X}_i)^T(\boldsymbol{x}-\boldsymbol{X}_i)}{2h_n^2}\right) \tag{8}$$

Observe that

$$\begin{aligned}(\boldsymbol{x}-\boldsymbol{X}_i)^T(\boldsymbol{x}-\boldsymbol{X}_i) &= -2\left(x^{(1)}X_i^{(1)} + x^{(2)}X_i^{(2)} + \ldots + x^{(d)}X_i^{(d)}\right)\\ &+ \left(x^{(1)}\right)^2 + \left(x^{(2)}\right)^2 + \ldots + \left(x^{(d)}\right)^2\\ &+ \left(X_i^{(1)}\right)^2 + \left(X_i^{(2)}\right)^2 + \ldots + \left(X_i^{(d)}\right)^2\end{aligned} \tag{9}$$

Now assuming normalization of the vectors x and Xi formula (8) simplifies to

$$\widehat{f}_n(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}\, nh_n^d} \sum_{i=1}^{n} \exp\left(-\frac{(1-\boldsymbol{x}^T\boldsymbol{X}_i)}{h_n^2}\right) \tag{10}$$

Figure 1 shows a neural realization of algorithm (10). The proposed net has d inputs and two layers. The first layer consists of n neurons and each neuron has d weights. The output layer has a single neuron with the linear activation function. We should emphasize that the proposed network does not require a training procedure (optimal choosing of connection weights). The succeeding coordinates of the observation vectors $\boldsymbol{X}_i$, $i = 1, \ldots, n$ play the role of the weights.

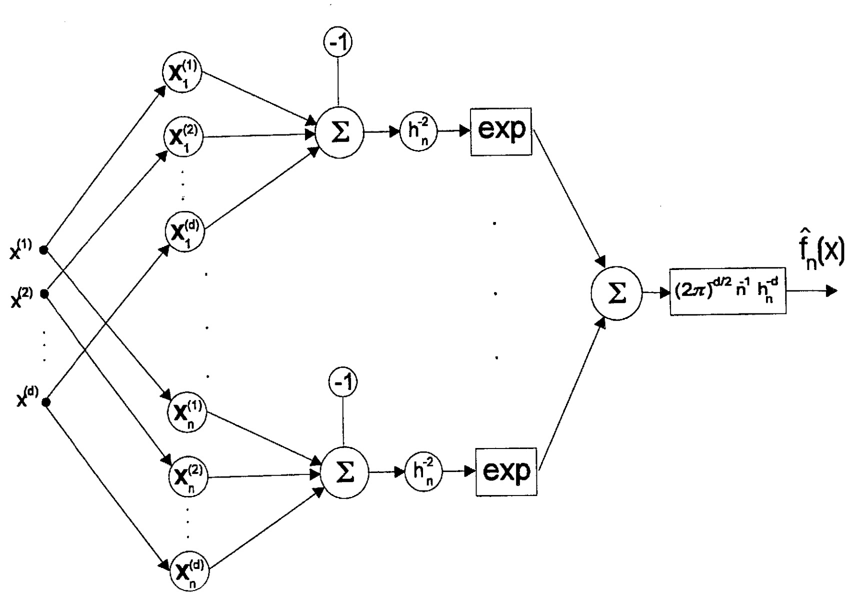

Fig. 1. Neural network density estimation.

The following theorem is a consequence of results presented by Parzen (1962) and Cacoullos (1966).

Theorem 1. *If the number of neurons in the first layer of the prabability neural network presented in Fig. 1 is chosen to satisfy conditions (6), then*

$$E\left[\widehat{f}_n(\boldsymbol{x}) - f_n(\boldsymbol{x})\right]^2 \xrightarrow{n} 0 \tag{11}$$

in the points at which f *is continuous.* ■

3. Pattern Classification

Let $(\boldsymbol{X}, \boldsymbol{Y})$, $(\boldsymbol{X}_1, \boldsymbol{Y}_1)$, $(\boldsymbol{X}_2, \boldsymbol{Y}_2), \ldots$, be a sequence of i.i.d. pairs of random variables; $\boldsymbol{Y}$ takes the values from the set; $S = \{1, \ldots, M\}$, whereas $\boldsymbol{X}$ takes the values in $\mathbb{R}^d$. The problem is to estimate $\boldsymbol{Y}$ from $\boldsymbol{X}$ and $\boldsymbol{V}_n$, where $\boldsymbol{V}_n = (\boldsymbol{X}_1, \boldsymbol{Y}_1), \ldots, (\boldsymbol{X}_n, \boldsymbol{Y}_n)$ is a learning sequence. Suppose that p_m and f_m, $m = 1, \ldots, M$, are the prior class probabilities and the class conditional densities, respectively. Define

$$T_{im} = \begin{cases} 1 & \text{if } \boldsymbol{Y}_i = m \\ 0 & \text{if } \boldsymbol{Y}_i \neq m \end{cases}$$

for $i = 1, 2, \ldots, n$ and $m = 1, 2, \ldots, M$. The Bayes discriminate function is given by

$$g_m(\boldsymbol{x}) = f(\boldsymbol{x}) E\left[T_{nm} | \boldsymbol{X}_n = \boldsymbol{x}\right] \tag{12}$$

where $f(\boldsymbol{x}) = \sum_{m=1}^{M} p_m f_m(\boldsymbol{x})$.

We consider a procedure of classifying every $\boldsymbol{x}$ to a class m, $m \in S$, which maximizes $\widehat{g}_{nm}(\boldsymbol{x})$, where $\widehat{g}_{nm}(\boldsymbol{x})$ is the following estimate of the Bayes discriminate function

$$\widehat{g}_{nm}(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}\, n h_n^d} \sum_{i=1}^{n} T_{im} \exp\left(-\frac{1 - \boldsymbol{x}^T \boldsymbol{X}_i)}{h_n^2}\right) \tag{13}$$

Figure 2 shows the neural network classifying pattern $\boldsymbol{x} \in \mathrm{I\!R}^d$ to the class m, $m \in S = \{1, 2\}$, for which the expression

$$\widetilde{g}_{nm}(\boldsymbol{x}) = \sum_{i=1}^{n} T_{im} \exp\left(-\frac{1 - \boldsymbol{x}^T \boldsymbol{X}_i)}{h_n^2}\right)$$

takes the maximum value.

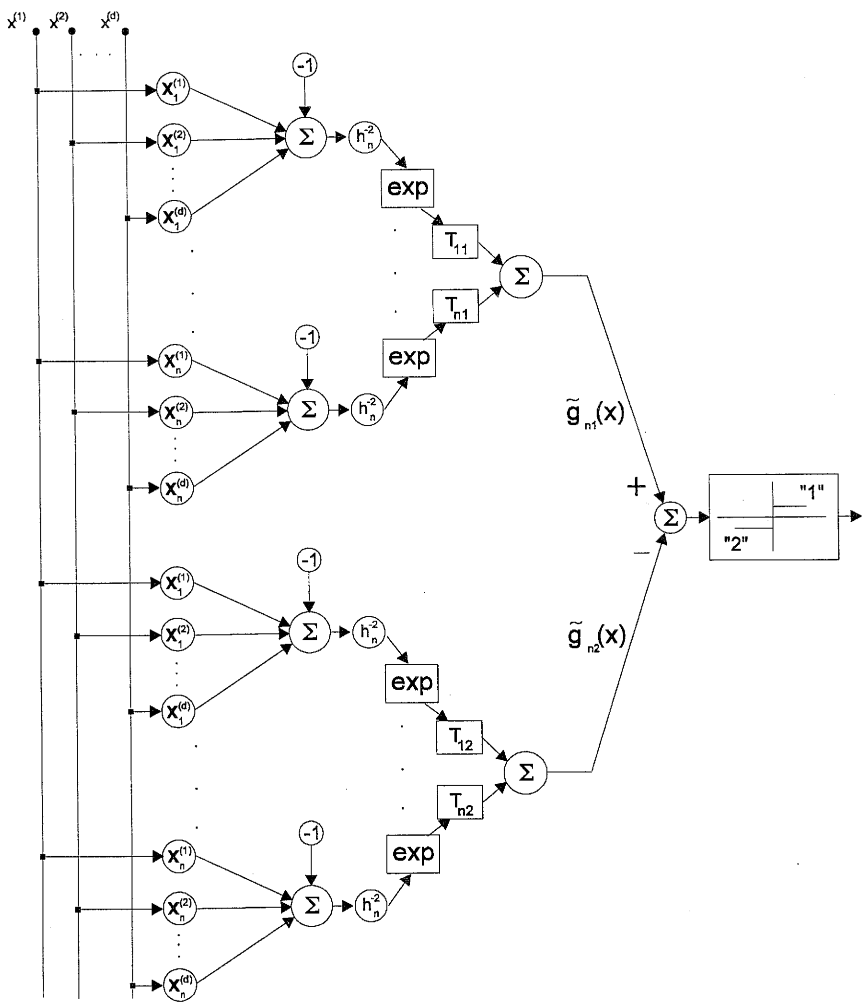

Fig. 2. Neural network pattern classification.

Let $\widehat{\boldsymbol{Y}}_n$ be a decision obtained according to the classification procedure defined above. Let $D_n = P(\widehat{\boldsymbol{Y}}_n \neq \boldsymbol{Y}|\boldsymbol{V}_n)$. The procedure is said to be weakly (strongly) Bayes risk consistent if $D_n \xrightarrow{n} D_0$ in probability (with probability one), where D_0 is the Bayes probability error. The result below follows from the theorem presented by Greblicki *et al.* (1984).

Theorem 2. *If the number of neurons of the probabilistic neural network presented in Fig. 2 satisfies conditions (6), then* $D_n \xrightarrow{n} D_0$ *in probability.* ■

4. Neural Realization of the Recurrent Non-parametric Estimation Algorithms

The following recurrent estimate of the probability density function was first proposed by Wolverton and Wagner (1969)

$$\widehat{f}_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_i^d}K\left(\frac{\boldsymbol{x}-\boldsymbol{X}_i)}{h_i}\right) \tag{14}$$

Expression (14) can be rewritten as

$$\widehat{f}_n(\boldsymbol{x}) = \frac{n-1}{n}f_{n-1}(\boldsymbol{x}) + \frac{1}{nh_n^d}K\left(\frac{\boldsymbol{x}-\boldsymbol{X}_n)}{h_n}\right) \tag{15}$$

For the kernel K of form (7), using the same arguments as those in Section 2, one gets

$$\widehat{f}_n(\boldsymbol{x}) = \frac{n-1}{n}f_{n-1}(\boldsymbol{x}) + \frac{1}{(2\pi)^{d/2}\,nh_n^d}\exp\left(-\frac{1-\boldsymbol{x}^T\boldsymbol{X}_n)}{h_n^2}\right) \tag{16}$$

Figure 3 shows the neural network performing algorthm (16) in a fixed point $\boldsymbol{x} \in \mathbb{R}^d$. The net consists of one neuron in the first layer having d inputs — coordinates of the vector $\boldsymbol{X}_n$, $n = 1, 2, \ldots$. Let us notice that the role of weights is played by the coordinates of the vector $\boldsymbol{x}$. The second layer also consists of only one neuron with the feedback typical for recurrent neural networks. When the density is to be estimated at several points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$, then the proposed structure should be copied L times. We shall obtain the neural network of L neurons processing the input observations in the parallel way.

Observe that the recurrent version of algorithm (13) takes the form

$$\widehat{g}_{n,m}(\boldsymbol{x}) = \frac{n-1}{n}\widehat{g}_{n-1,m}(\boldsymbol{x}) + \frac{1}{(2\pi)^{d/2}\,nh_n^d}T_{n,m}\exp\left(-\frac{1-\boldsymbol{x}^T\boldsymbol{X}_n)}{h_n^2}\right) \tag{17}$$

The corresponding recurrent neural network for pattern classification is shown in Fig. 4.

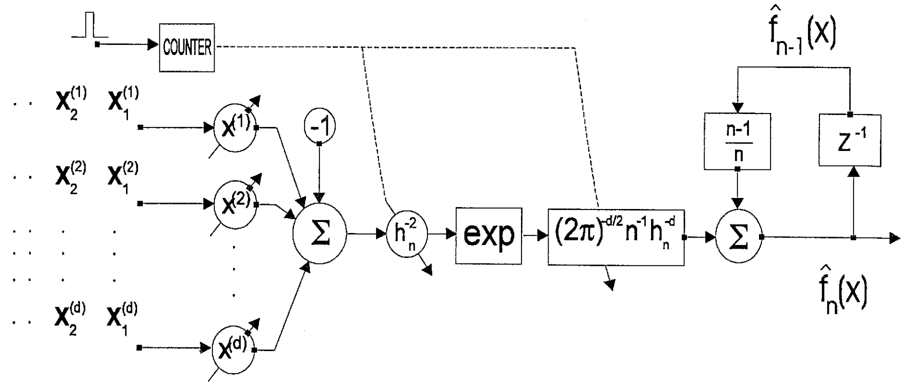

Fig. 3. Neural network density estimation — recurrent algorithm.

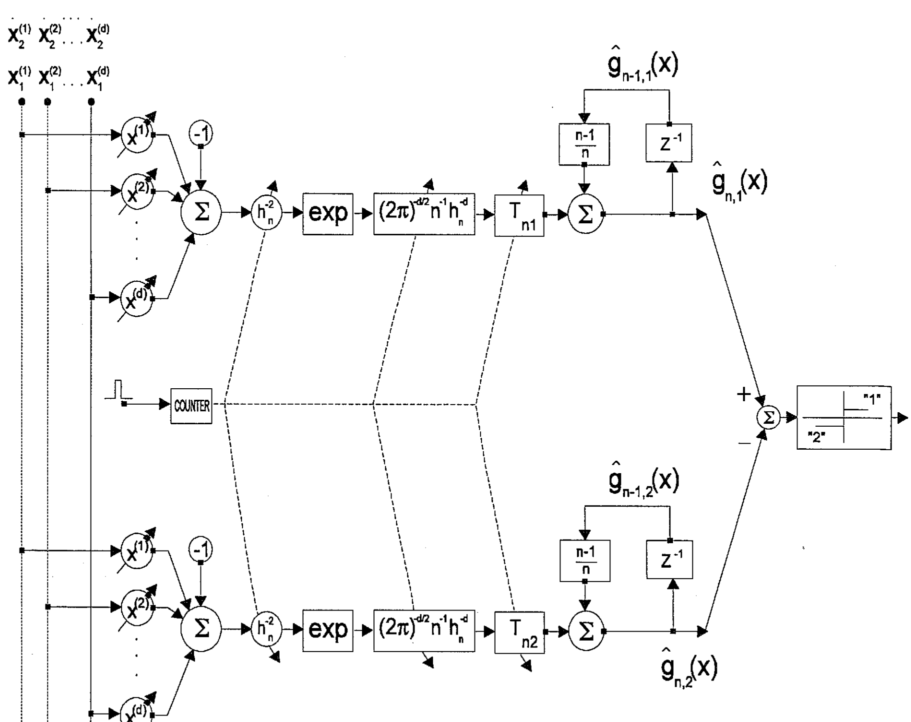

Fig. 4. Neural network pattern classification — reccurent algorithm.

5. Identification by Regression Function Estimation

5.1. Stochastic Input Signal

Let $(\boldsymbol{X}, \boldsymbol{Y})$ be a pair of random variables. $\boldsymbol{X}$ takes values in $\mathbb{R}^d$, whereas $\boldsymbol{Y}$ takes values in $\mathbb{R}$. Let f be the marginal Lebesgue density of $\boldsymbol{X}$. Based on the sample

$(\boldsymbol{X}_1, \boldsymbol{Y}_1), ..., (\boldsymbol{X}_n, \boldsymbol{Y}_n)$ of independent observations of $(\boldsymbol{X}, \boldsymbol{Y})$, we wish to estimate the regression $\mathrm{I\!R}$ of $\boldsymbol{Y}$ on $\boldsymbol{X}$, i.e. $\mathrm{I\!R}(x) = E[\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}]$. The probabilistic neural network that provides estimates of function $\mathrm{I\!R}$ and converges to the underlying (linear or non-linear) regression surface is presented in Fig. 4. In system identification the independent variable $\boldsymbol{X}$ is the system input and the dependent variable $\boldsymbol{Y}$ is the system output. Assume that $k(\boldsymbol{x}, y)$ represents the joint probability density function of a vector random variable $\boldsymbol{X}$ and a scalar random variable $\boldsymbol{Y}$. The conditional mean of $\boldsymbol{Y}$ given $\boldsymbol{x}$ (particular measured value of random variable $\boldsymbol{X}$) is given by

$$\mathrm{I\!R}(\boldsymbol{x}) = E\big[\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}\big] = \frac{1}{f(\boldsymbol{x})} \int_{-\infty}^{+\infty} y k(\boldsymbol{x}, y)\, \mathrm{d}y \tag{18}$$

If the probability density functions f and k are unknown, then they should be estimated from a sample of observations of $\boldsymbol{X}$ and $\boldsymbol{Y}$. The joint probability density estimator derived from the Gaussian kernel takes the form:

$$\widehat{k}_n(\boldsymbol{x}, y) = \frac{1}{(2\pi)^{(d+1)/2}\, n h_n^{d+1}} \sum_{i=1}^{n} \exp\left(-\frac{(1 - \boldsymbol{x}^T \boldsymbol{X}_i)}{h_n^2}\right) \exp\left(-\frac{(y - \boldsymbol{Y}_i)^2}{2h_n^2}\right) \tag{19}$$

Replacing $k(\boldsymbol{x}, y)$ by $\widehat{k}_n(\boldsymbol{x}, y)$ and $f(\boldsymbol{x})$ by $\widehat{f}_n(\boldsymbol{x})$ in (18) and performing the indicated integration we get the following estimate of the regression function:

$$\widehat{\mathrm{I\!R}}_n(\boldsymbol{x}) = \frac{\displaystyle\sum_{i=1}^{n} \boldsymbol{Y}_i \exp\left(\frac{1 - \boldsymbol{x}^T \boldsymbol{X}_i}{h_n^2}\right)}{\displaystyle\sum_{i=1}^{n} \exp\left(\frac{1 - \boldsymbol{x}^T \boldsymbol{X}_i}{h_n^2}\right)} \tag{20}$$

Figure 5 shows a neural network realization of algorithm (20).

One may easily derive the neural network structure for recurrent regression function estimation. The corresponding net is shown in Fig. 6.

Theorem 3. *If $E|\boldsymbol{Y}| < \infty$ and the number of neurons of the probabilistic neural network presented in Fig. 5 satisfies conditions (6), then $\widehat{\mathrm{I\!R}}_n \xrightarrow{n} \mathrm{I\!R}$ in probability for almost all $\boldsymbol{x} \in \mathrm{I\!R}^d$.* ■

This result follows the theorem given by Greblicki *et al.* (1984).

5.2. Deterministic Input Signal

Many engineering problems are concerned with systems described by the following equation:

$$y_i = \mathrm{I\!R}(x_i) + Z_i, \qquad i = 1, ..., n$$

relating the input x_i and output y_i, and the measurement noise Z_i.

Consider the d-dimensional unit cube $Q = [0, 1]^d$. Let $n^{1/d}$ be an integer, and $i_j = 1, ..., n^{1/d}$, $j = 1, ..., d$. Partition the interval $[0, 1]$ on each axis into $n^{1/d}$ subsets $\Delta x_{j,i_j}$. Define the following Cartesian product

$$\Delta x_{1,i_1} \times \Delta x_{2,i_2} \times ... \times \Delta x_{d,i_d} = Q_{d,i}$$

Let $Q_{d,i} \cap Q_{d,j} = \emptyset$, for $i \neq j$ and $\cup Q_{d,i} = Q_d$. The inputs x_i are selected so that $x_i \in Q_{d,i}$.

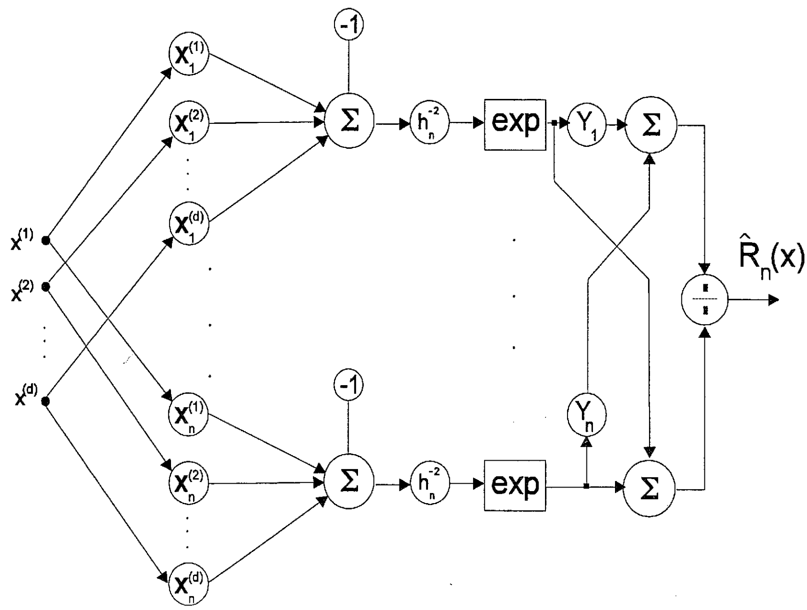

Fig. 5. Neural regression function estimation — stochastic input signal.

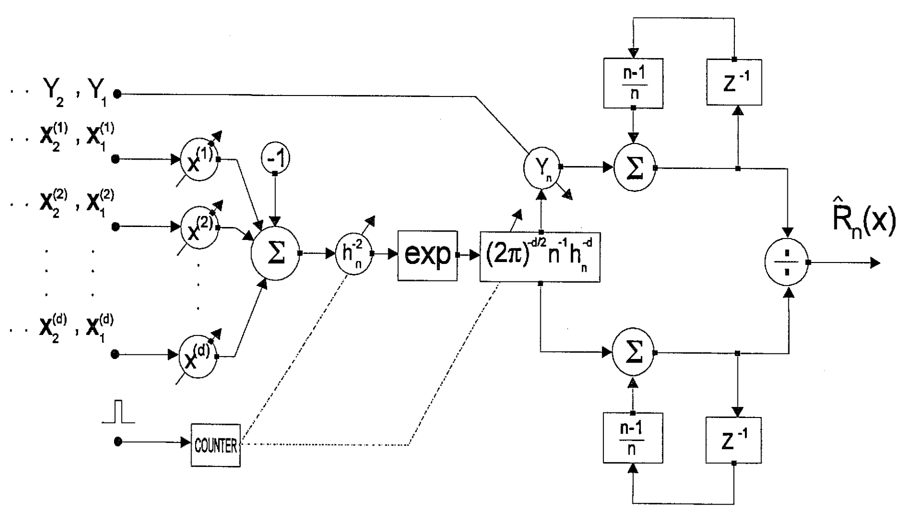

Fig. 6. Recurrent neural regression estimation.

We propose the following algorithm

$$\widehat{\mathbb{R}}_n(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}\, n h_n^d} \sum_{i=1}^{n} \boldsymbol{Y}_i \exp\left(-\frac{(1-\boldsymbol{x}^T\boldsymbol{X}_i)}{h_n^2}\right) \tag{21}$$

The appropriate net is shown in Fig. 7.

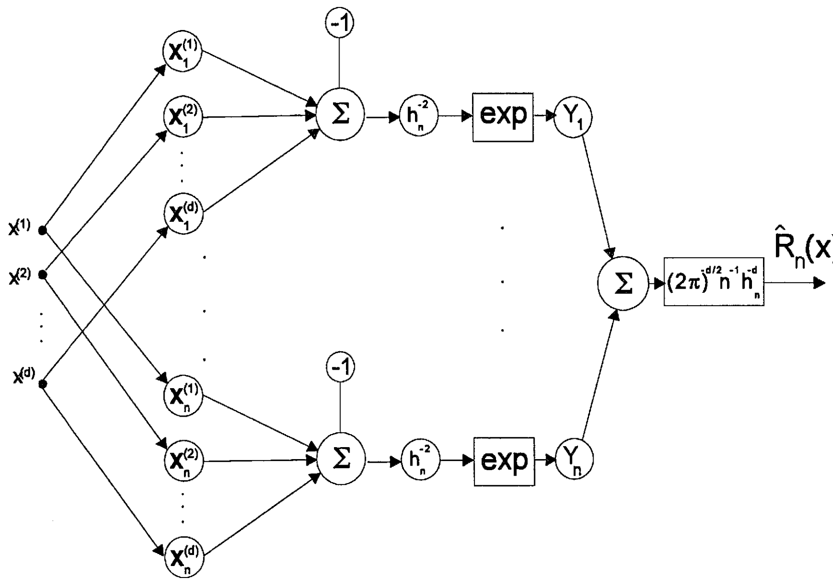

Fig. 7. Neural regression function estimation — deterministic input signal.

Theorem 4. *If the number of neurons of the probabilistic neural network presented in Fig. 7 satisfies conditions (6) and* $\sup_{1 \le i \le n} \sup_{x,y \in Q_i} ||x - y|| = O(n^{-1/d})$, *then* $\widehat{\mathbb{R}}_n \xrightarrow{n} \mathbb{R}$ *in probability at all continuity points* x *of* $\mathbb{R}$. ■

The above theorem follows from theorems presented by Gałkowski and Rutkowski (1985) and Georgiev (1990).

References

Cacoullos T. (1966): *Estimation of multivariate density.* — Annals of Institute of Statistical Mathematics, v.18, pp.179–189.

Gałkowski T. and Rutkowski L. (1985): *Non-parametric recovery of multivariate functions with applications to system identification.* — Proc. IEEE, v.73, pp.942–943.

Gałkowski T. and Rutkowski L. (1986): *Non-parametric fitting of multivariate functions.* — IEEE Trans. Automat. Contr. v.AC–31, No.8, pp.785–787.

Gałkowski T. and Rutkowski L. (1994): *Realizacja algorytmów rozpoznawania i nieparametrycznej estymacji za pomocą sieci neuronowych.* — Proc. 1st Nat. Conf. *Neural Networks and Their Applications*, Kule – Częstochowa, (Poland), pp.237–242, (in Polish).

Georgiev A.A. (1990): *Non-parametric multiple functions fitting.* — Statistics and Probability Letters, No.3.

Greblicki W., Krzyżak A. and Pawlak M. (1984): *Distribution- free pointwise consistency of kernel regression estimate.* — The Annals of Statistics, v.12, No.4, pp.1570–1575.

Parzen E. (1962): *On estimation of a probability density function and mode.* — Annals of Math. Statistics, v.33, pp.1065–1076.

Rosenblatt M. (1956): *Remarks on some estimates of a density function.* — Annals of Math. Statistics, v.27, pp.823–837.

Rutkowski L. (1988): *Sequential pattern recognition procedures derived from multiple Fourier series.* — Pattern Recognition Letters, No.8, pp.213–216.

Rutkowski L. and Rafajłowicz E. (1989): *On global rate of convergence of some non-parametric identification procedures.* — IEEE Trans. Automat. Contr. v.AC–34, No.10, pp.1089–1091.

Rutkowski L. (1991): *Identification of MISO non-linear regressions in the presence of a wide class of disturbances.* — IEEE Trans. Information Theory, v.IT–37, pp.214–216.

Rutkowski L. (1993): *Multiple Fourier series procedures of non-linear regressions from noisy data.* — IEEE Trans. Signal Processing, v.41, No.10, pp.3062–3065.

Specht D.F. (1990): *Probabilistic neural networks.* — Neural Networks, v.3, pp.109–118.

Wolverton C.T. and Wagner T.J. (1969): *Recursive estimates of probability densities.* — IEEE Trans. SSC, v.SSC–5, No.3