ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 27-32

Equivalence of a Radial Basis Function NN and a Perceptron

Igor Grabec
Faculty of Mechanical Engineering, University of Ljubljana,
POB 394, 61000 Ljubljana, Slovenia

Abstract. This article deals with a transformation of conditional average estimator, which is utilized in the description of a radial basis function NN, to the multi-layer perceptron equation. The exact equivalence of both paradigms is demonstrated for a one-dimensional case with symmetric triangular activation functions. The transformation provides a simple interpretation of perceptron parameters in terms of statistical samples of input data.

1. Introduction

Multi-layer perceptrons (MLP) have played a central role in the research of neural networks.[1] Their study began with the nonlinear and adaptive response characteristics of neurons which have brought with them many difficulties related to the understanding of the collective properties of MLP. Consequently, it was discovered by a cumbersome analysis only recently that MLP is a universal approximator of relations between input signals.[1,2] But a supervised training of MLP by back-propagation of errors is time-consuming and does not provide a simple interpretation of MLP parameters. The inclusion of *a priori* information into MLP is also problematic. Many of these problems do not appear in simulations of radial basis function neural networks (RBFN). Their structure stems from the representation of empirical probability density functions of sensory signals in terms of prototype data which can simply be interpreted statistically.[3,4] An optimal description of relations is in this case described by a conditional average estimator (CA) which represents a general, non-linear regression. *A priori* information can also be included by initialization of prototypes. A learning rule which was derived from maximum entropy principle describes a self-organized adaptation of neural receptive fields.[5] The separation of input signals into independent and dependent variables need not be done before training, as at MLP, but it can be performed when applying a trained network. Because of these convenient properties of RBFN our aim was to explore if it is also more accurate at the modeling of mapping relations. Here we demonstrate their exact equivalence for a simple one-dimensional case by showing that the mapping relation of a RBFN can be converted into that of a MLP. This further indicates how the MLP parameters can be statistically interepreted.

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 27-32

2. Transition from RBFN to MLP

The task of both paradigms is a modeling of relations between measured data. We assume that D sensors provide signals $(s_1, s_2, \ldots, s_D)$ that comprise a vector $\mathbf{x}$. The modeling is here treated as an estimation of the joint probability density function (PDF) between components of $\mathbf{x}$. We assume that information about the probability distribution is obtained by a repetition of measurements which yield N independent samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. The PDF is then estimated by the expression

$$f_e(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} w(\mathbf{x} - \mathbf{x}_n, \sigma) \tag{1}$$

Here $w(\mathbf{x} - \mathbf{x}_n, \sigma)$ is a smooth approximation of the delta function, such as a radially symmetric Gaussian kernel $w(\mathbf{x}, \sigma) = \text{const } \exp[- \parallel \mathbf{x} \parallel^2 / 2\sigma^2]$.[6]

However, the complete PDF need not be stored, but it is sufficient to preserve a set of samples $\mathbf{x}_n$, while the width σ can be properly estimated as a typical distance between neighbor sample points. At continuous measurement the number of samples increases without limit and there arises a problem with the finite capacity of the memory in which the data are stored. Neural networks are comprised of finite numbers of memory cells and therefore we must assume that the PDF can be represented by a finite number K of prototype vectors $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_K\}$ as

$$f_r(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^{K} w(\mathbf{x} - \mathbf{q}_k, \sigma) \tag{2}$$

At the modeling of f_r the prototypes are first initialized by K samples: $\{\mathbf{q}_k = \mathbf{x}_k \text{ , for } k = 1 \ldots K\}$, which represent *a priori* given information. These prototypes can be adapted to additional samples $\mathbf{x}_N$ in such a way that a mean square difference between f_e and f_r is minimized. The corresponding rule was derived elsewhere and it describes a self-organized unsupervised learning of neurons each of which contains one prototype $\mathbf{q}_k$.[5]

At an application of adapted PDF the information must be extracted from prototypes which generally corresponds to some kind of statistical estimation. In a typical application there is given some partial information, for instance the first i components of the vector: $\mathbf{g} = (s_1, s_2, .., s_i, \emptyset)$ and the hidden data, which are to be estimated, are then represented by the vector $\mathbf{h} = (\emptyset, s_{i+1}, .., s_D)$.[4] Here $\emptyset$ denotes the missing part in a truncated vector. As an optimal estimator we apply the conditional average [3,4] which can be expressed by prototype vectors as :

$$\widehat{\mathbf{h}} = \sum_{k=1}^{K} B_k(\mathbf{g}) \mathbf{h}_k \quad \text{where} \quad B_k(\mathbf{g}) = \frac{w(\mathbf{g} - \mathbf{q}_k, \sigma)}{\sum_{j=1}^{K} w(\mathbf{g} - \mathbf{q}_j, \sigma)} \tag{3}$$

Here the given vector $\mathbf{g}$ plays the role of the condition. The basis functions $B_k(\mathbf{g})$ are strongly nonlinear and peaked at the truncated vectors $\mathbf{q}_k$. They

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 27-32

represent the measure of similarity between the given vector $\mathbf{g}$ and the prototypes. The CA represents a general non-linear non-parametric regression. It is important that selection into given and hidden data can be done after training of the network.

The CA corresponds to a mapping relation $\mathbf{g} \rightarrow \mathbf{h}$ which can be realized by a two layer RBFN.[3] The first layer consists of K neurons. The k-th neuron obtains the input signal $\mathbf{g}$ over synapses described by $\mathbf{g}_k$ and is excited as described by the radial basis function $B_k(\mathbf{g})$. The corresponding excitation signal is then transferred to the neurons of the second layer. The i-th neuron of this layer has synaptic weights $h_{k,i}$ and generates the output $\widehat{h}_i(\mathbf{g})$.

In order to obtain a relation with MLP it is instructive to analyze the performance of RBFN in a simple two-dimensional case. We consider the function $y(x)$ which is described by a set of sample pairs $\{x_1, y_1; x_2, y_2; \ldots; x_N, y_N\}$ with constant spacing $\triangle x = x_{j+1} - x_j$ for $j = 1 \ldots N-1$. We further introduce a triangular and a piecewise linear sigmoidal basis function :

$$B_i(x) = \{1 - \frac{|x - x_i|}{\triangle x} \ldots \text{for } x_{i-1} < x < x_{i+1} \ ; \ 0 \ldots \text{elsewhere}\} \tag{4}$$

$$S_i(x) = \{0 \text{ for } x < x_i \ ; \ (x - x_i)/\triangle x \text{ for } x_i \leq x \leq x_{i+1} \ ; \ 1 \text{ for } x_{i+1} < x\} \tag{5}$$

Using them we can represent the function $y(x)$ by straight line segments connecting the sample points. The CA can be transformed into MLP expression by utilizing the relations:

$$B_{i+1}(x) = S_i(x) - S_{i+1}(x) \tag{6}$$

$$S_i(x) = \begin{cases} 0 & \ldots \ x < x_{i-1} \\ \frac{B_{i+1}(x)}{B_i(x)+B_{i+1}(x)} & \ldots \ x_{i-1} \leq x \leq x_{i+2} \\ 1 & \ldots \ x > x_{i+2} \end{cases} \tag{7}$$

The result is :

$$\begin{aligned} \widehat{y}(x) &= \frac{y_1 B_1(x) + \ldots + y_N B_N(x)}{B_1(x) + \ldots + B_N(x)} = \frac{y_1 B_1(x)}{B_1(x) + \ldots + B_N(x)} + \ldots + \frac{y_N B_N(x)}{B_1(x) + \ldots + B_N(x)} \\ &= \frac{y_1 B_1(x)}{B_1(x) + B_2(x)} + y_2 B_2(x) + \ldots + y_{N-1} B_{N-1}(x) + \frac{y_N B_N(x)}{B_{N-1}(x) + B_N(x)} \end{aligned} \tag{8}$$

In the denominator of the first and the last terms of this expression just those basis functions are kept which differ from zero in the region where the basis function in the numerator also differ from zero. The denominator in the terms of index 2 to $N-1$ is 1 because of the overlapping of neighboring basis function. We insert relations of Eq. (6,7) into Eq.(8) and obtain

$$\widehat{y}(x) = y_1 + \sum_{i=1}^{N-1} (y_{i+1} - y_i) \, S_i(x) \tag{9}$$

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 27-32

By introducing the parameters $\Delta y_i = y_{i+1} - y_i, c_i = 1/(x_{i+1} - x_i), \Theta_i = x_i/(x_{i+1} - x_i)$ and a unique, normalized sigmoidal basis function

$$S(x) = \{0 \ldots \text{ for } x < 0 \ ; \ x \ldots \text{ for } 0 \leq x \leq 1 \ ; \ 1 \ldots \text{ for } x > 1\} \qquad (10)$$

we can write Eq.(9) in the form of a two-layer perceptron mapping relation

$$\widehat{y}(x) = y_1 + \sum_{i=1}^{N-1} \Delta y_i \, S(c_i x - \Theta_i) \qquad (11)$$

The first layer corresponds to neurons with the synaptic weights c_i and threshold values Θ_i while the second layer contains a linear neuron with synaptic weights Δy_i and threshold y_1.

The above derivation demonstrates that for the two-dimensional distribution, the mapping $x \rightarrow y$ determined by conditional average is identical with the mapping relation of a multi-layer perceptron. However, a difference appears when the operations needed for the mapping are executed. The operators involved in both cases are described by different basis functions which are corresponding to different neurons in the implementation. If the prototypes are not evenly spaced, then the last equation can still be applied, although the transition regions will be of different spans. However, in this case the basis functions $B_i(x)$ are no longer symmetric. In applications it is more convenient to use a Gaussian basis function rather than a triangular one and this yields in the perceptron expression the function tanh(.). In this case the estimated function $\widehat{y}(x)$ generally does not run trough the sample points but rather approximates them by a function having more smooth derivative as the piecewise linear function. In this case the correspondence between RBFN and MLP is not exact but approximate. An additional interpretation is needed when the data are not related by a regular function $y(x)$ but randomly as described by a joint probability density function $f(x, y)$. In this case, various values of y can be observed at given x. Evaluation of CA in this case is not problematic while in the perceptron relation Eq.(11) the value y_i must be substituted by the conditional average of variable y at x_i.

The analysis of the correspondence between RBFN and MLP can be extended to multi-variate mappings. Let us first consider the situation with just two prototypes $\mathbf{q}_i$ and $\mathbf{q}_j$ and Gaussian basis functions. The CA is then described by the function

$$\widehat{\mathbf{h}}(\mathbf{g}) = \frac{\mathbf{h}_i \exp(- \parallel \mathbf{g} - \mathbf{g}_i \parallel^2 /2\sigma^2) + \mathbf{h}_j \exp(- \parallel \mathbf{g} - \mathbf{g}_j \parallel^2 /2\sigma^2)}{\exp(- \parallel \mathbf{g} - \mathbf{g}_i \parallel^2 /2\sigma^2) + \exp(- \parallel \mathbf{g} - \mathbf{g}_j \parallel^2 /2\sigma^2)} \qquad (12)$$

We introduce the notation : $\mathbf{g}_i = \overline{\mathbf{g}} - \Delta\mathbf{g}$, $\mathbf{g}_j = \overline{\mathbf{g}} + \Delta\mathbf{g}$, $\mathbf{h}_i = \overline{\mathbf{h}} - \Delta\mathbf{h}$, $\mathbf{h}_j = \overline{\mathbf{h}} + \Delta\mathbf{h}$ in which overline denotes the average value and $2\Delta\mathbf{g}$ is the spacing of the prototypes. If we express the norm by a scalar product and cancel the term $\exp[-(\parallel \mathbf{g} - \overline{\mathbf{g}} \parallel^2 + \parallel \Delta\mathbf{g} \parallel^2)/2\sigma^2)$ in the numerator and denominator, we obtain the expression:

$$\widehat{\mathbf{h}}(\mathbf{g}) = \overline{\mathbf{h}} + \Delta\mathbf{h} \tanh\left[\Delta\mathbf{g} \cdot (\mathbf{g} - \overline{\mathbf{g}})/\sigma^2\right] \qquad (13)$$

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 27-32

in which $\cdot$ denotes the scalar product. In order to obtain the relation between RBFN and MLP, we introduce a weight vector $\mathbf{c} = \triangle\mathbf{g}/\sigma^2$ and a threshold value $\Theta = \overline{\mathbf{g}} \cdot \triangle\mathbf{g}/\sigma^2$ into Eq. (13) and obtain:

$$\widehat{\mathbf{h}}(\mathbf{g}) = \overline{\mathbf{h}} + \triangle\mathbf{h} \tanh\left[\mathbf{c} \cdot \mathbf{g} - \Theta\right] \tag{14}$$

This expression again describes a two layer perceptron: the first layer is comprised of one neuron having the synaptic weights described by the vector $\mathbf{c}$ and the threshold value Θ. The second layer is comprised of linear neurons having synaptic weights $\triangle h_i$ and threshold values $\overline{h_i}$. A first order approximation of the mapping expression is

$$\widehat{\mathbf{h}}(\mathbf{g}) = \overline{\mathbf{h}} + \triangle\mathbf{h}\triangle\mathbf{g} \cdot (\mathbf{g} - \overline{\mathbf{g}})/\sigma^2 \tag{15}$$

This equation represents a linear regression of $\mathbf{h}$ on $\mathbf{g}$ which runs trough both prototype points if we put $\sigma^2 = \| \triangle\mathbf{g} \|^2$. Its slope is determined by the covariance matrix $\mathbf{\Sigma} = \triangle\mathbf{h}\triangle\mathbf{g}^{\mathrm{T}}$. However, the nonlinear regression specified in Eq.(13) follows a linear regression only in the vicinity of point determined by $\overline{\mathbf{g}}$ and $\overline{\mathbf{h}}$ while it exhibits saturation when $\mathbf{g}$ runs from $\overline{\mathbf{g}}$ over given prototypes to infinity. The saturation is a consequence of function tanh(.) which is basic in the modeling of multi-layered perceptron.

The reasoning presented above for a multi-variate case requires additional explanation when transferred to a situation consisting of many prototypes. Let us assume that N prototypes with indexes $1 \ldots N$ can be found in the hypersphere of radius approximately σ around the given datum $\mathbf{g}$ and let these prototypes be spaced for approximately equal distances. The CA can now be expressed with leading terms and remainders as

$$\widehat{\mathbf{h}}(\mathbf{g}) = \frac{\sum_{i=1}^{N} \mathbf{h}_i \exp(- \| \mathbf{g} - \mathbf{g}_i \|^2 / 2\sigma^2)}{\sum_{i=1}^{N} \exp(- \| \mathbf{g} - \mathbf{g}_i \|^2 / 2\sigma^2) + O_w} + O_h \tag{16}$$

Here O_h and O_w represent two remainders which are small in comparison with the two leading terms. We again introduce the average value, but now with respect to N prototypes: $\mathbf{g}_i = \overline{\mathbf{g}} + \triangle\mathbf{g}_i$, $\mathbf{h}_i = \overline{\mathbf{h}} + \triangle\mathbf{h}_i$ for $i = 1 \ldots N$. We obtain the approximate expression:

$$\widehat{\mathbf{h}}(\mathbf{g}) \cong \overline{\mathbf{h}} + \frac{\sum_{i=1}^{N} \triangle\mathbf{h}_i \exp[\triangle\mathbf{g}_i \cdot (\mathbf{g} - \overline{\mathbf{g}})/\sigma^2]}{\sum_{i=1}^{N} \exp[\triangle\mathbf{g}_i \cdot (\mathbf{g} - \overline{\mathbf{g}})/\sigma^2]} \tag{17}$$

For $\mathbf{g}$ in the vicinity of the average value, a linear approximation of exponential function is applicable which yields

$$\widehat{\mathbf{h}}(\mathbf{g}) \cong \overline{\mathbf{h}} + \frac{1}{N} \sum_{i=1}^{N} \triangle\mathbf{h}_i\triangle\mathbf{g}_i \cdot (\mathbf{g} - \overline{\mathbf{g}})/\sigma^2 \tag{18}$$

This expression represents a linear regression of $\mathbf{h}$ on $\mathbf{g}$ specified by N points. If we express the matrix by two principal vectors $\triangle\mathbf{h}_p$ and $\triangle\mathbf{g}_p$:

$$\mathbf{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} \triangle\mathbf{h}_i\triangle\mathbf{g}_i^{\mathrm{T}} = \triangle\mathbf{h}_p\triangle\mathbf{g}_p^{\mathrm{T}} \tag{19}$$

we obtain a simplified expression of the linear regression

$$\widehat{\mathbf{h}}(\mathbf{g}) \cong \overline{\mathbf{h}} + \triangle\mathbf{h}_p \triangle\mathbf{g}_p \cdot (\mathbf{g} - \overline{\mathbf{g}})/\sigma^2 \tag{20}$$

which is an approximation of a MLP mapping relation

$$\widehat{\mathbf{h}}(\mathbf{g}) \cong \overline{\mathbf{h}} + \triangle\mathbf{h}_p \tanh[\triangle\mathbf{g}_p \cdot (\mathbf{g} - \overline{\mathbf{g}})/\sigma^2] \tag{21}$$

The parameters of a single neuron in the perceptron expression thus correspond to the principal vectors of the covariance matrix determining a local regression around the center of several neighboring prototypes. The above expression shows that the transition from RBFN to MLP can be quite generally performed. However, in the multi-variate case, the decomposition of CA into a perceptron mapping is not so simple as in the one-dimensional case because the interpretation of perceptron parameters goes over local regression determined by various prototypes surrounding the given datum $\mathbf{g}$.

3. Conclusions

The conditional average representing a regular function $y(x)$ can be exactly decomposed into multilayer perceptron relation. When there are a small number of noise corrupted sample data points representing the function the question of proper smoothing arises. In the case of CA this is done by using symmetric radial basis functions and increasing their width. However, when the prototypes are obtained by self-organization, they represent a statistical regularity and the CA generally does not exhibit statistical fluctuations. In this case, the proper RBF width is determined by the distance between closest neighbors. The corresponding parameters of the perceptron for the one-dimensional mapping can then be simply interpreted in terms of prototypes. Because of the complexity of the back-propagation learning, we have not yet analytically demonstrated that it yields the same MLP parameters as the decomposition of the conditional average.

Acknowledgement: The author thanks Prof. W. Sachse for valuable suggestions in the preparation of this article.

4. References

1. R. Hecht-Nielsen, Neurocomputing, Addison-Wesley, Reading, MA, 1990
2. G. Cybenko, Math. Cont., Sig. & Syst., **2**, 303-314 (1989)
3. C. M. Bishop, Rev. Sci. Instr., **65**, 1830-1832 (1994)
4. I. Grabec, W. Sachse, J. Appl. Phys., **69** (9), 6233-6244 (1991)
5. I. Grabec, Biological Cybernetics,**64**, 403-409 (1990)
6. R. O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, John Wiley, NY, 1973