ANN WHICH COVERS MLP AND RBF

Josef Boštík, Jaromír Kukal

Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
Department of Software Engineering

Abstract

Two basic types of artificial neural networks Multi Layer Perceptron (MLP) and Radial Basis Function network (RBF) are frequently discussed. The MLP is built of one type of neuron decomposable into linear and sigmoid part. The second type (RBF) consists of radial and linear neurons. The new Multi layer Radial Basis Function (M-RBF) consists of two types of neurons: linear and extended sigmoid ones. Four layer M-RBF network should approximate any RBF network while five layer M-RBF network should replace any MLP network with three layers. The new M-RBF network than generalize abilities of both basic types of ANN. The network and its learning are demonstrated on numerical example. The results are compared with RBF and MLP.

Keywords: *ANN, MLP, RBF, sigmoid function, decomposition.*

1 Introduction

1.1 Basic characteristics of MLP

General *Multi Layer Perceptron* (MLP) [1] [2] network consists of single input layer, at least one hidden layer and single output layer. The signal processing in every hidden and output neuron is described by formula

$$y = \mathrm{f}\left(\sum_{k=0}^{n} w_k x_k\right) \tag{1}$$

where $n \in \mathbf{N}$ is number of neuron inputs, $x_k, w_k \in \mathbf{R}$ are k^{th} input value and its weight, $y \in [a;b]$ is neuron output, $x_0 = 1$ and $\mathrm{f} : \mathbf{R} \to [a;b]$ is a non-decreasing continuous *sigmoid function*. Traditional example of sigmoid function is *logistic function*

$$\mathrm{f}_1(s) = \frac{1 + \tanh 2s}{2} \tag{2}$$

Other examples of sigmoid functions are

$$\mathrm{f}_2(s) = \frac{1}{2} + \frac{1}{\pi} \arctan \pi s \tag{3}$$

$$\mathrm{f}_3(s) = \frac{1}{2} + \frac{s}{1 + 2|s|} \tag{4}$$

$$\mathrm{f}_4(s) = \min\left(1, \max\left(0, \frac{1}{2} + s\right)\right) \tag{5}$$

$$\mathrm{f}_5(s)=\frac{1}{2}+\frac{1}{2}\operatorname{erf}\sqrt{\pi}s \tag{6}$$

where $\operatorname{erf} x=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,\mathrm{d}t$

1.2 Basic characteristics of RBF
General *Radial Basis Function* (RBF) [2] [3] network consists of three layers: input, hidden and output ones. The signal processing in every output neuron is described by linear formula

$$y=\sum_{k=0}^{n} w_k x_k \tag{7}$$

where $n\in\mathbf{N}$ is number of neuron inputs, $x_k, w_k \in\mathbf{R}$ are k^{th} input value and its weight, $y\in\mathbf{R}$ is neuron output and $x_0=1$. The signal processing in every hidden neuron is described by formula

$$y=\exp\left(-\frac{1}{2\sigma^2}\sum_{k=1}^{n}(x_k-w_k)^2\right) \tag{8}$$

where $\sigma>0$ is space factor, $y\in(0;1]$ is neuron output and the other quantities have the same meaning.

2 Problem Formulation

2.1 Universal non–linear element for MLP and ANN
Having our favorite sigmoid function we can construct base function for RBF hidden layer as a product of $\mathrm{f}(s)$ and $\mathrm{f}(-s)$ but alas the radial property is not guaranteed. Base function can be used for the approximation of RBF neuron using formula

$$\mathrm{G}(s_1,\ldots,s_n)=\prod_{k=1}^{n}\mathrm{f}(s_k)\mathrm{f}(-s_k) \tag{9}$$

Let

$$\mathrm{f}(s)=\frac{1+\operatorname{sign}(s)\sqrt{1-\exp\left(-4s^2\right)}}{2} \tag{10}$$

be a new kind of sigmoid function. In this case

$$g(s)=\mathrm{f}(s)\mathrm{f}(-s)=\frac{1-\operatorname{sign}^2(s)\left(1-\mathrm{e}^{-4s^2}\right)}{4}=\frac{1}{4}e^{-4s^2} \tag{11}$$

is a Gaussian kernel function. After the substitution (11) into (9) we obtain a funcion which is proportional to traditional radial basis function.

The existence of new sigmoid function

$$\mathrm{f}_6(s)=\frac{1+\operatorname{sign}(s)\sqrt{1-\exp\left(-4s^2\right)}}{2} \tag{12}$$

is in contradiction with traditional logistic function $\mathrm{f}_1(s)=\frac{1+\tanh 2s}{2}$. The absolute difference between them can be denoted as $\mathrm{D}(s)=\left|\mathrm{f}_6(s)-\mathrm{f}_1(s)\right|$ and reach absolute maximum $\mathrm{D}=0.0207995$ for $s=0.684824$, meanwhile $\mathrm{D}(0)=\mathrm{D}'(0)=\mathrm{D}''(0)=\lim_{s\to+\infty}\mathrm{D}(s)=\lim_{s\to-\infty}\mathrm{D}(s)=0$. So the difference between new and traditional perceptron characteristics is rather symbolic then dramatic and the way is open for the design of new artificial neural network.

2.2 New structure of ANN which covers MLP and RBF

The definition of new sigmoid function $\mathrm{f}_6(s)$ is a motivation to build up a new kind of artificial neural network with two types of processing neurons.

Definition 1

Let $n\in\mathbf{N}$, $x_k, w_k \in\mathbf{R}$ for $k=0,...,n$, $x_0=1$. Then the function

$$y=\varphi(\mathbf{x},\mathbf{w})=\sum_{k=0}^{n} w_k x_k \tag{13}$$

is called *linear neuron.*

Definition 2

Let $n\in\mathbf{N}$, $s_k\in\mathbf{R}$ for $k=1,...,n$. Then the function

$$\mathrm{G}(\mathbf{s})=\frac{1}{2^n}\prod_{k=1}^{n}\left(1+\operatorname{sign}(s_k)\left(1-\exp\left(-4s_k^2\right)\right)^{1/2}\right) \tag{14}$$

is called *multiplicative perceptron*.

Now we can define M-RBF ANN as hierarchical artificial neural network with processing layers of two kinds.

Definition 3

Let $L\in\mathbf{N}$ be number of layers. Let $N_k\in\mathbf{N}$ be number of neurons in k^{th} layer of hierarchical ANN for $k=1,...,L$. Let the 1^{st} layer consist of input neurons. Let L^{th} layer be output layer. Let $2j$ layer consists of linear neurons for $j=1,...,\lfloor L/2\rfloor$. Let $2j+1$ layer consists of multiplicative perceptrons for $j=1,...,\lceil L/2\rceil-1$. Then the network is called *Multi layer Radial Basis Function ANN* and denoted as M–RBF–N_1–N_2–…–N_L or M–RBF L.

Now, it is easy to recognize that any linear ANN can be realized as M–RBF 2, any RBF network can be realized as M–RBF 4, any two layer perceptron network can be approximated as M–RBF 3, any three layer perceptron network with linear output (MLL) can be approximated as M–RBF 4 and any three layer perceptron network (MLP) can be approximated as M–RBF 5. The M–RBF is a hierarchical ANN with incomplete connectivity of neighbor layers. Any multiplicative perceptron need not operate on complete set of neurons of previous linear layer. Formally, the novel network is a function $\boldsymbol{y}=\mathrm{ANN}\left(\boldsymbol{x},\boldsymbol{W}^{(1)},...,\boldsymbol{W}^{(L-1)}\right)$.

Supposing the training set of patterns $\left(\mathbf{x}_k,\mathbf{y}_k^*\right)$ for $k=1,...,M$, we can use least square method for learning of M-RBF. Then the sum of squares

$$\mathrm{SSQ}\left(\boldsymbol{W}^{(1)},\ldots,\boldsymbol{W}^{(L-1)}\right)=\sum_{k=1}^{M}\left\|\boldsymbol{y}_k^* - \mathrm{ANN}\left(\boldsymbol{x}_k,\boldsymbol{W}^{(1)},\ldots,\boldsymbol{W}^{(L-1)}\right)\right\|^2 \tag{15}$$

is subject of minimization, where $\|\ldots\|$ is euclidean norm.

The relationships between new ANN, MLL, MLP, RBF and OLAM will help to find initial structure and weights of M–RBF and then continue to the nearest local optimum of SSQ^*.

2.3 Testing and comparison

The new testing environment in Matlab was designed. It allows adding new ANN models and compare them with another ones. The weights of ANN are optimized by the application of fmincon function in Matlab which employs the LSQ method to global minimum searching via repeated local optimization from random initial points. Following values are recorded:

- *ni* number of input neurons
- *nh* number of neurons in hidden layer
- *no* number of output neurons
- *nw* number of weights
- *df* degrees of freedom
- *ssq* sum of squares of ANN residues
- *sy* model error as

$$sy = \sqrt{\frac{ssq}{df}} \tag{16}$$

The improvement of model error was tested in the case of M–RBF network related to RBF one. The M–RBF learning was based on optimum RBF weights.The weights of optimum RBF network were converted to equivalent weights of of M–RBF. The optimum parameters of M–RBF were searched in the neighborhood of this initial estimate in constrained space with 10% tolerance as local LSQ optimum, of course.

3 Results

The tests were performed on ANN time series prediction task. Data set of annual number of sunspots (sunspots.dat) is freely available in the Matlab environment. Only MLP with characteristics f_1, RBF with characteristics g_1 and M–RBF with three input neurons and single output neuron were tested and compared. Various number of hidden neurons up to 5 was used for the study of their properties.

Results of MLP, RBF and M–RBF learning were collected in the Tab. 1. The optimum structures with the smallest model error (within type of ANN) were MLP 3–4–1, RBF 3–3–1 and M–RBF 3–4–1. But M–RBF network had the better performance than MLP or RBF one.

4 Conclusions

Three types of ANN (MLP, RBF, M–RBF) were learned to be the best predictors of sunspot number from last three year history. The results show that M–RBF was the best model in the case of model error minimization. The M-RBF network has more weights then MLP or RBF with the same number nonlinear neurons, which reduces the degrees of freedom. However, this effect is included in the model error calculations and thus we recommend the M-RBF network as very efficient tool for data modeling. The learning of M-RBF network have to be predeceased with RBF learning, which is a good generator of initial M-RBF weight estimate.

Acknowledgement: The paper was created under the support of grant OHK4-165/11 CTU in Prague.

References:

[1] Rumelhart D.E., Hinton G.E.,Williams R.J., Learning Representations by Back-Propagating Error. *Nature*, 323, 1986, pp. 533-536.
[2] Haykin S., *Neural Networks*, Macmillan, New York, 1994.
[3] Poggio T, Networks for Approximation and Learning, *Proceedings of the IEEE*, 78, 1990, pp. 1481-1497.

Josef Boštík, Jaromír Kukal
Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
Department of Software Engineering
Břehová 7, Praha 1, 11519
Czech republic
josefbostik@gmail.com, jaromir.kukal@fjfi.cvut.cz

Tab.1: Results of MLP, RBF and M-RBF learning

ANN model	*ni*	*nh*	*no*	*nw*	*df*	*sy*	*ssq*
MLP	3	1	1	6	278	0.144562	5.809707
MLP	3	2	1	11	273	0.128509	4.508461
MLP	3	3	1	16	268	0.126624	4.297032
MLP	3	4	1	21	263	**0.125553**	4.145817
MLP	3	5	1	26	258	0.126550	4.131851
RBF	3	1	1	6	278	0.132752	4.899236
RBF	3	2	1	11	273	0.124625	4.240088
RBF	3	3	1	16	268	**0.122964**	4.051967
RBF	3	4	1	21	263	0.123880	4.036092
RBF	3	5	1	26	258	0.124729	4.011853
M–RBF	3	1	1	8	276	0.117532	3.812577
M–RBF	3	2	1	15	269	0.117688	3.725774
M–RBF	3	3	1	22	262	0.116791	3.573745
M–RBF	3	4	1	29	255	**0.115388**	3.395183
M–RBF	3	5	1	36	248	0.115343	3.299403