Perceptron Network for RBF Lovers

Josef Boštík

Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering

Introduction

- There are two basic types of artificial neural networks:
  - Multi Layer Perceptron (MLP)
  - Radial Basis Function network (RBF).
- New PRBF (Perceptron Radial Basis Function) generalize MLP and RBF network abilities

MLP preliminaries

- Single input layer, at least one hidden layer and single output layer

- Consists of one type of neuron, which can be decomposed into linear and sigmoid part.

MLP preliminaries

- The signal processing formula

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

where $n \in \mathbf{N}$ is number of neuron inputs

- f is called *sigmoid function*

Sigmoid function properties

- $f : \mathbf{R} \to [a;b]$ is a non-decreasing continuous function satisfying: $f(s) + f(-s) = a + b$
- is concave on $\mathbf{R}_0^+$
- $\lim_{s \to +\infty} f(s) = b$
- $f''(0)$ exists
- $\lim_{s \to +0} f''(s)$ exists

Sigmoid function properties

- The other properties of sigmoid function can be easily derived:
- $f(0) = (a+b)/2$
- f is convex on $\mathbf{R}_0^-$
- $\lim_{s \to -\infty} f(s) = a$
- $\lim_{s \to 0-} f''(s) = -\lim_{s \to 0+} f''(s)$

Sigmoid functions

- Traditional example of sigmoid function is *logistic function*

$$f_L(s) = \left(1 + \exp(-s)\right)^{-1}$$

$$f_1(s) = \frac{1 + \tanh 2s}{2}$$

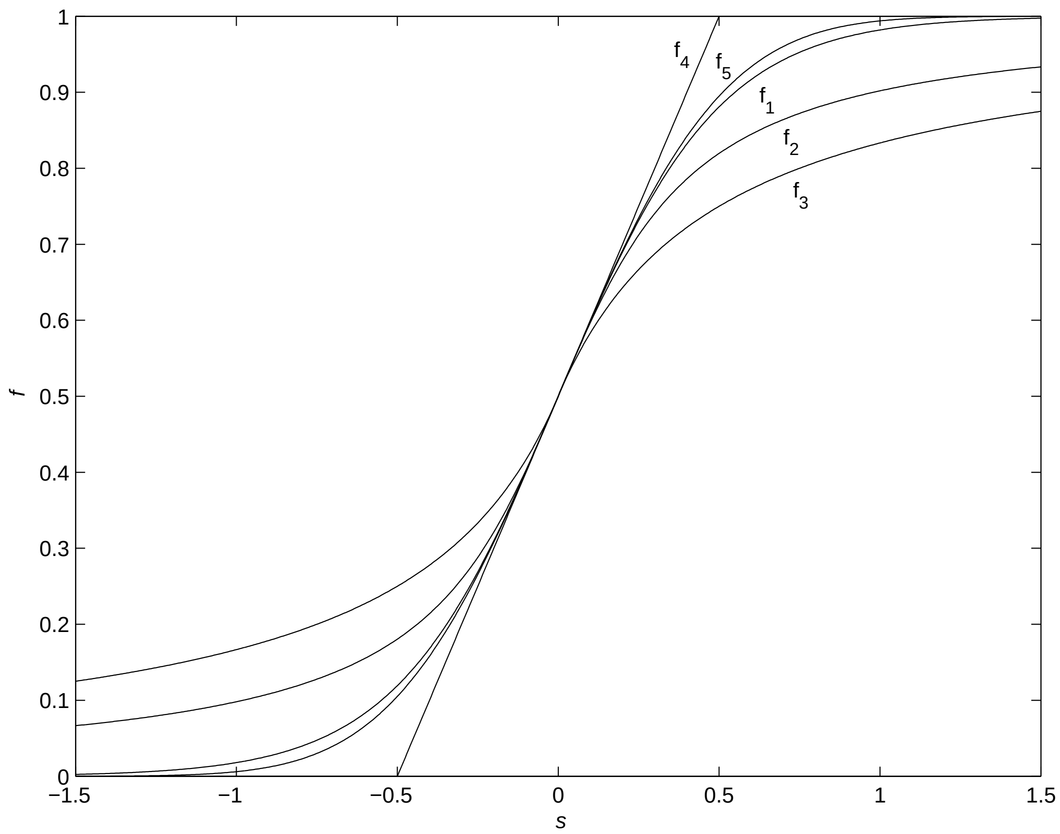
$$f_2(s) = \frac{1}{2} + \frac{1}{\pi} \arctan \pi s$$

$$f_3(s) = \frac{1}{2} + \frac{s}{1 + 2|s|}$$

$$f_4(s) = \min\left(1, \max\left(0, \frac{1}{2} + s\right)\right)$$

$$f_5(s) = \frac{1}{2} + \frac{1}{2} \operatorname{erf} \sqrt{\pi} s$$ where $$\operatorname{erf} x = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \, dt$$

Various sigmoid functions



RBF preliminaries

- consists of three layers: input, hidden and output one

- the signal processing in every output neuron is described by linear formula

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

where $n \in \mathbf{N}$ is number of neuron inputs, and $x_0 = 1$

RBF preliminaries

- the signal processing formula

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

Where $\sigma > 0$ is space factor.

- substitution $s_k = \frac{x_k - w_k}{\sqrt{2}\sigma}$

$$y = \mathrm{G}(s_1, \ldots, s_n) = \exp\left(-\sum_{k=1}^{n} s_k^2 \right)$$

Where $\mathrm{G} : \mathbf{R}^n \to (0;1]$ is *radial function.*

G is separable

- Let $g_R(s) = \exp(-s^2)$ then

$$y = G(s_1, \dots, s_n) = \exp\left(-\sum_{k=1}^{n} s_k^2\right) = \prod_{k=1}^{n} g_R(s_k)$$

- multiplicative construction of function G
- g is called *base function*

Generalized properties of function g

- Let $g : \mathbf{R} \to [0;1/4]$ be continuous function satisfying:
- $g(s) = g(-s)$
- $g(0) = 1/4$
- is non-increasing on $\mathbf{R}_0^+$
- $\lim_{s \to +\infty} g(s) = 0$
- $g''(0) = -2$

Other properties of base function

- g is non-decreasing on $\mathbf{R}_0^-$
- $\lim_{s \to -\infty} g(s) = 0$

Base functions

$$g_1(s) = \frac{1}{4}\exp\left(-4s^2\right)$$

$$g_2(s) = \frac{1}{4}\left(1 + 4s^2\right)^{-1}$$

$$g_3(s) = \frac{1}{4}\max\left(0,1 - 4s^2\right)$$

Perceptron approximation of RBF

- Having our favorite sigmoid function we can construct base function $g(s)$ as a product of $f(s)$ and $f(-s)$.

- **Theorem 1:**

Let f be sigmoid function with $a = 0$, $b = 1$, $f'(0) = 1$

Then the function $g(s) = f(s) f(-s)$ is base function.

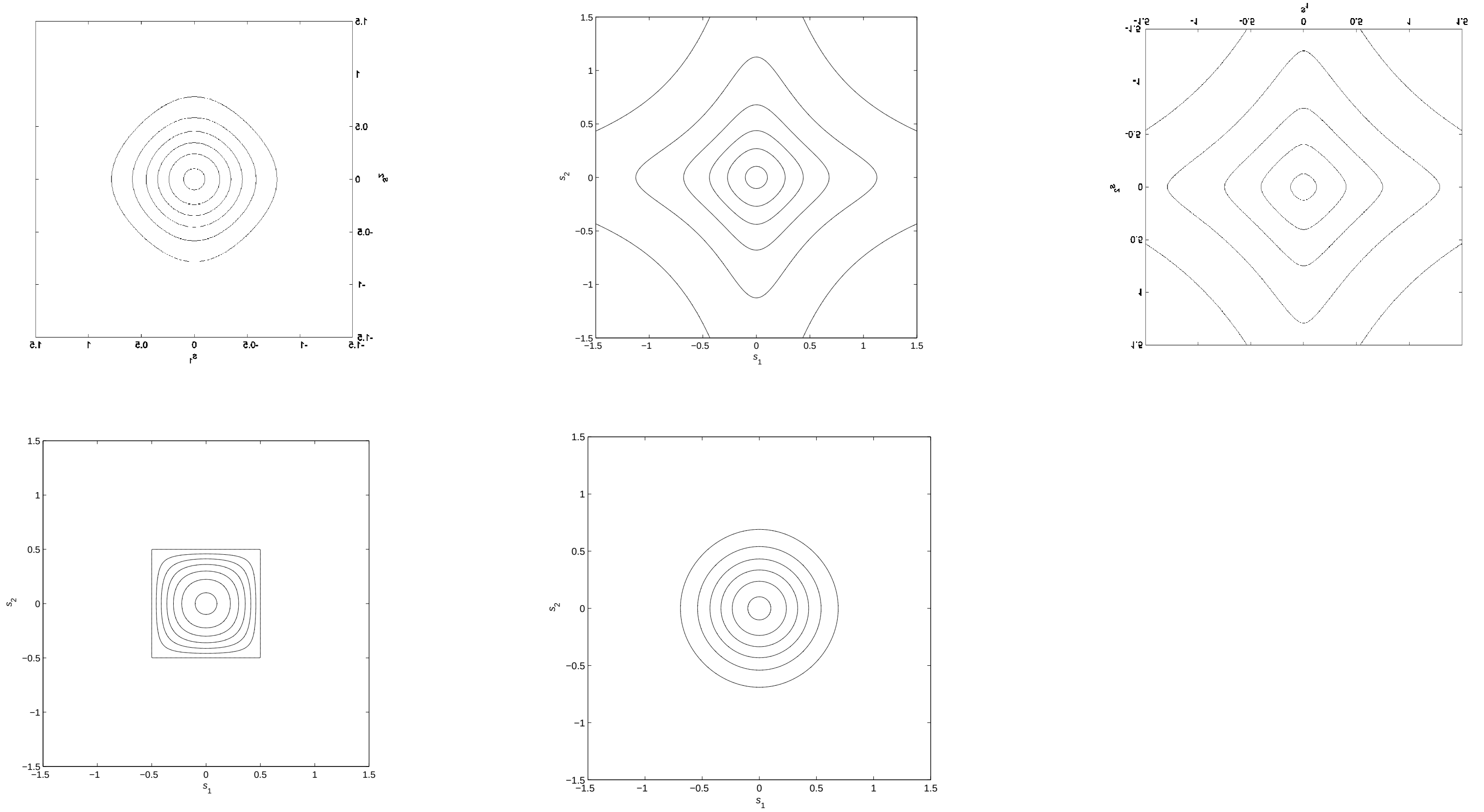
RBF neuron approximation

- Base function can be used for the approximation of RBF neuron using formula

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

- The non-radial shape can be demonstrated for $n=2$ and sigmoid functions $\mathrm{f}_1,\ldots,\mathrm{f}_5$

RBF neuron approximation



Theorem 2

Let

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

and

$$g(s) = f(s)f(-s)$$

Then f, g are sigmoid and radial base functions.

New sigmoid vs. logistic function

- The difference between the new sigmoid function

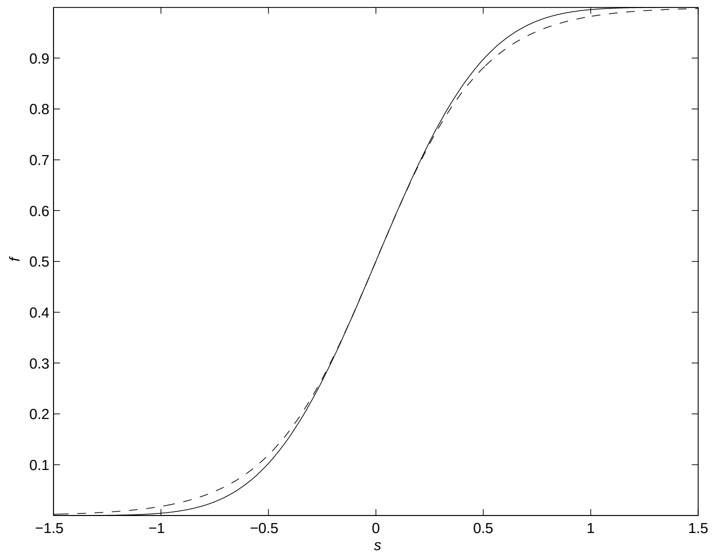
$$f_6(s) = \frac{1 + \mathrm{sign}(s)\sqrt{1 - \exp\left(-4s^2\right)}}{2}$$

and traditional perceptron characteristics

$$f_1(s) = \frac{1 + \tanh 2s}{2}$$

is rather symbolic then dramatic.

New sigmoid vs. logistic function



$$D(s) = \left| f_6(s) - f_1(s) \right| \qquad D_{max} = 0.0207995$$

Perceptron Radial Basis Function ANN

- Two types of processing neurons

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

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

is called *linear neuron.*

Perceptron Radial Basis Function ANN

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

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

is called *multiplicative perceptron*.

PRBFL

- 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 $k = 1, \ldots, L$.

- Let $2j$ layer consists of linear neurons and $2j+1$ of multiplicative perceptrons.

- Then the network is called *Perceptron Radial Basis Function ANN* and denoted as $\text{PRBF} - N_1 - N_2 - \ldots - N_L$ or $\text{PRBF}L$.

PRBF properties

Any linear ANN can be realized as PRBF 2.

Any RBF network can be realized as PRBF4.

PRBF network is able to realize any OLAM and RBF network and approximate any MLP2, MLP3 and MLL networks with logistic perceptrons.

PRBF learning

- PRBF network is a function

$$\mathbf{y} = \mathrm{PRBF}\left(\mathbf{x}, \mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L-1)}\right)$$

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

PRBF learning

- Then the sum of squares

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

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

PRBF learning

- The minimization of $\mathrm{SSQ}\left(\mathbf{W}^{(1)},\ldots,\mathbf{W}^{(L-1)}\right)$ is mixed integer non-linear programming task which is difficult to solve.

- local optimization
- random shooting (Monte Carlo)
- stochastic gradient learning
- differential evolution
- annealing
- combination of previous principles

PRBF testing

- Matlab testing environment
- 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.

PRBF testing

- MLP with characteristics $f_1(s) = \frac{1 + \tanh 2s}{2}$

  RBF with characteristics $g_1(s) = \frac{1}{4}\exp\left(-4s^2\right)$

  And PRBF with three input neurons and single output neuron were tested and compared.

- Various number of hidden neurons (up to 5)

Results

- *ni* – nuber 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 variances
- *sy* – model error as

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

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
PRBF	3	1	1	8	276	0.117532	3.812577
PRBF	3	2	1	15	269	0.117688	3.725774
PRBF	3	3	1	22	262	0.116791	3.573745
PRBF	3	4	1	29	255	**0.115388**	3.395183
PRBF	3	5	1	36	248	0.115343	3.299403

Conclusions

- The results show that PRBF was the best model in the case of model error minimization.

- The PRBF 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 PRBF network as very efficient tool for data modeling.

Thank you for your attention.