Lecture 6

Radial basis functions neural networks

Neural networks offer a powerful framework for representing nonlinear mappings from several inputs to one or more outputs.

There are many different paradigms of neural networks. Some of them are nonparametric, e.g. PNN and k-nearest neighbors, and they do not involve the estimation of a set of parameters. Some of them are parametric, e.g. the linear discriminant function.

An important application of neural networks is regression. Instead of mapping the inputs into a discrete class label, the neural network maps the input variables into continuous values. A major class of neural networks is the radial basis function (RBF) neural network. We will look at the architecture of RBF neural networks, followed by its applications in both regression and classification.

The architecture of RBF neural network

Denote the input as $\mathbf{x}$, and the output as $y(\mathbf{x})$, the architecture of a RBF neural network is given by

$$y(\mathbf{x}) = \sum_{i=1}^{M} w_i \exp\left(-\frac{(\|\mathbf{x} - \mathbf{c}_i\|)^2}{2\sigma^2}\right)$$

if we choose to use Gaussian function as basis functions. Here $\mathbf{c}_i$ are called centers and σ is called the width. There are M basis functions centered at $\mathbf{c}_i$. w_i are called weights.

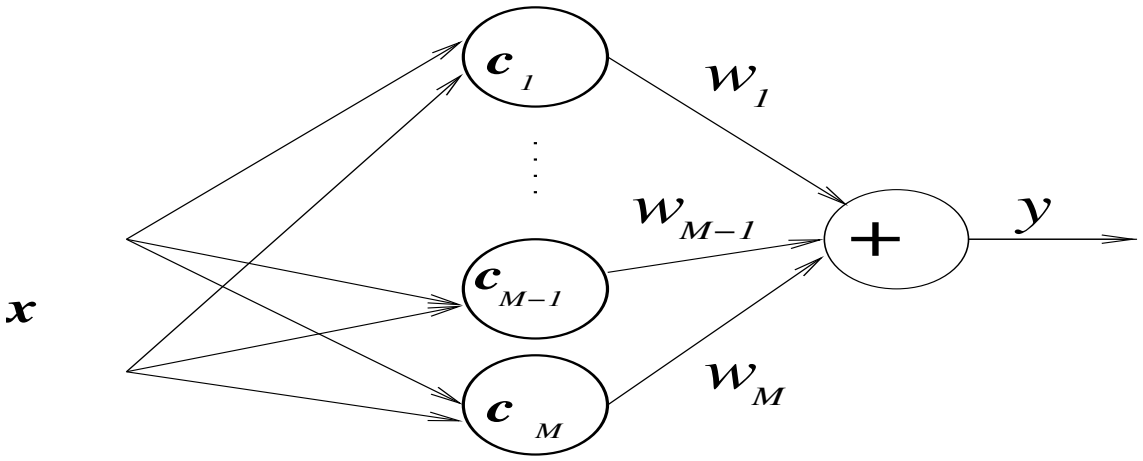


Figure: The architecture of RBF neural network.

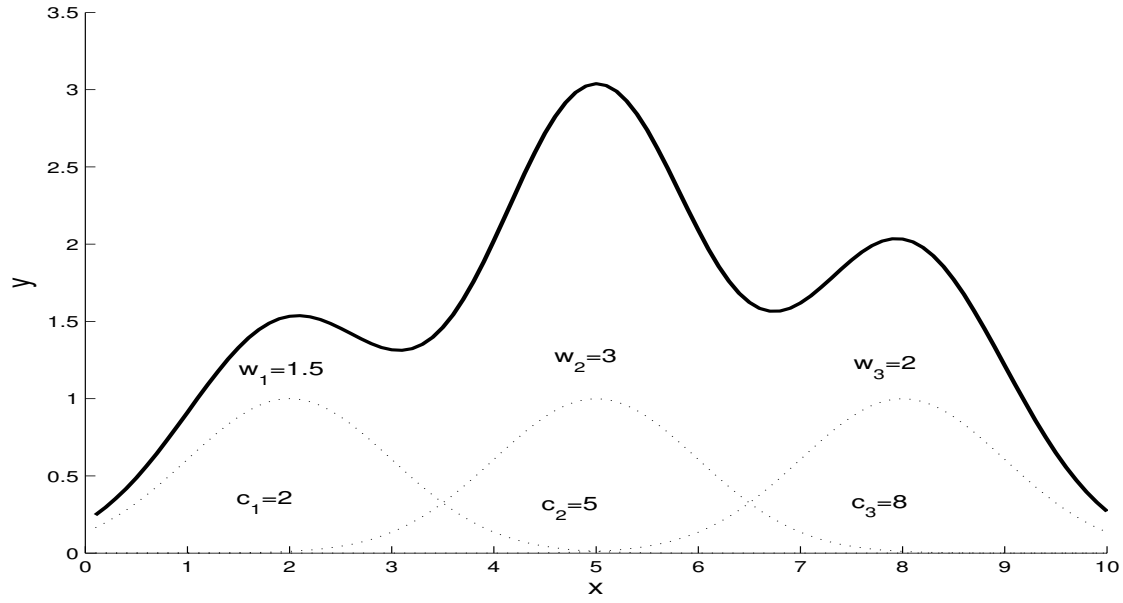
Curve fitting using RBF neural networks

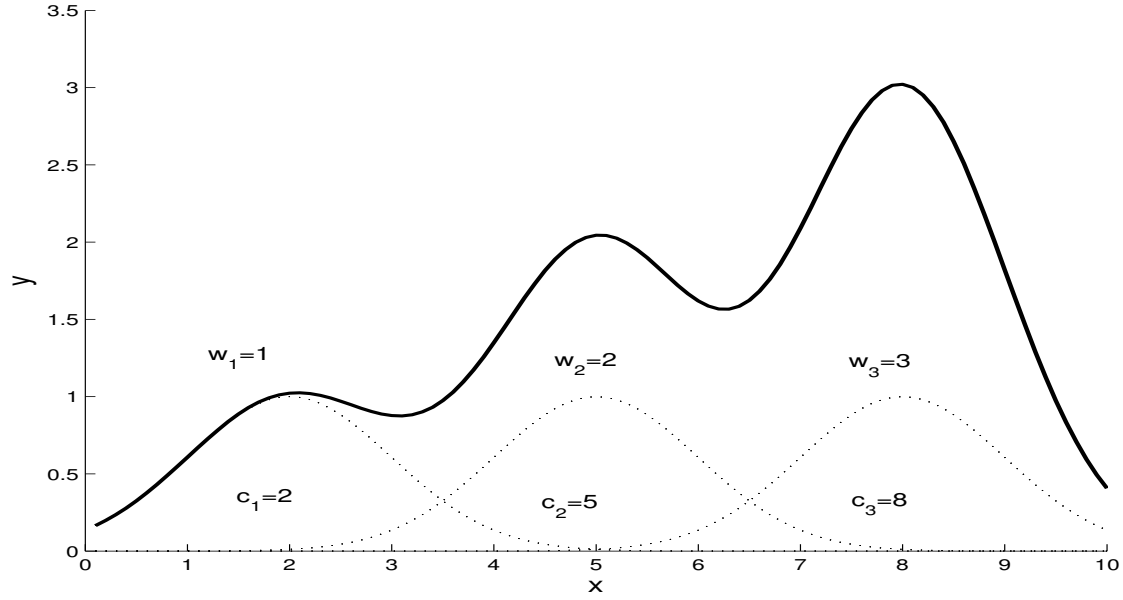
The applications of RBF neural networks in regression can be introduced by examples of curve fitting. Taking $\sigma = 1$, $c_1 = 2$, $c_2 = 5$ and $c_3 = 8$, the following figures plot the curves by varying the weights or centers in

$$y(x) = \sum_{i=1}^{3} w_i \exp\left(-\frac{(x - c_i)^2}{2}\right)$$

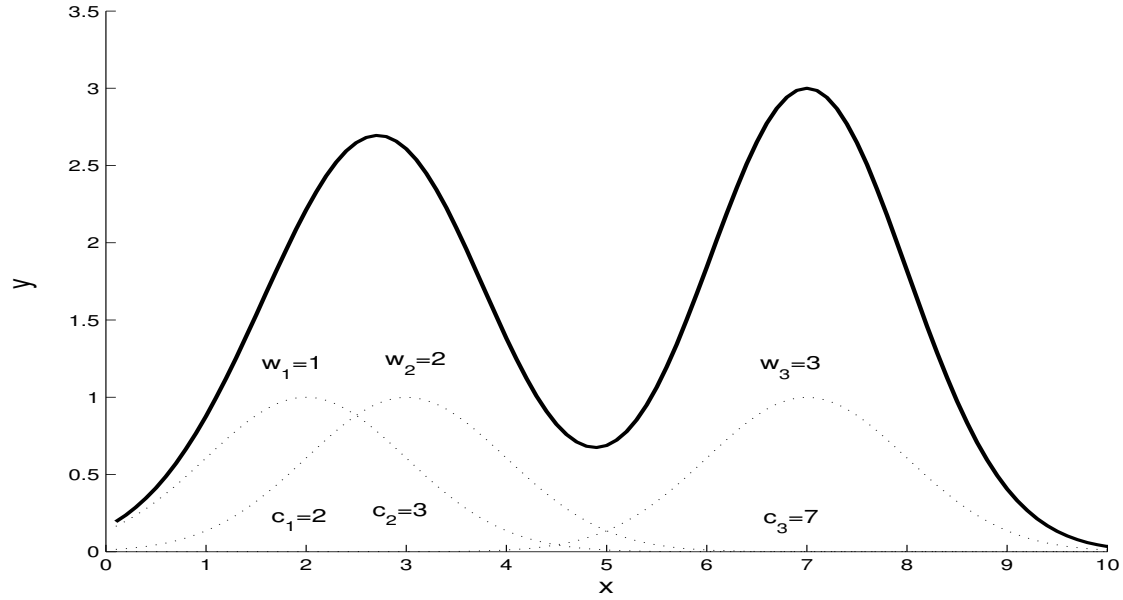
The shape of curves are adjustable by changing the weights or centers. This suggests that we can use RBF to approximate any unknown nonlinear function given observational data samples.

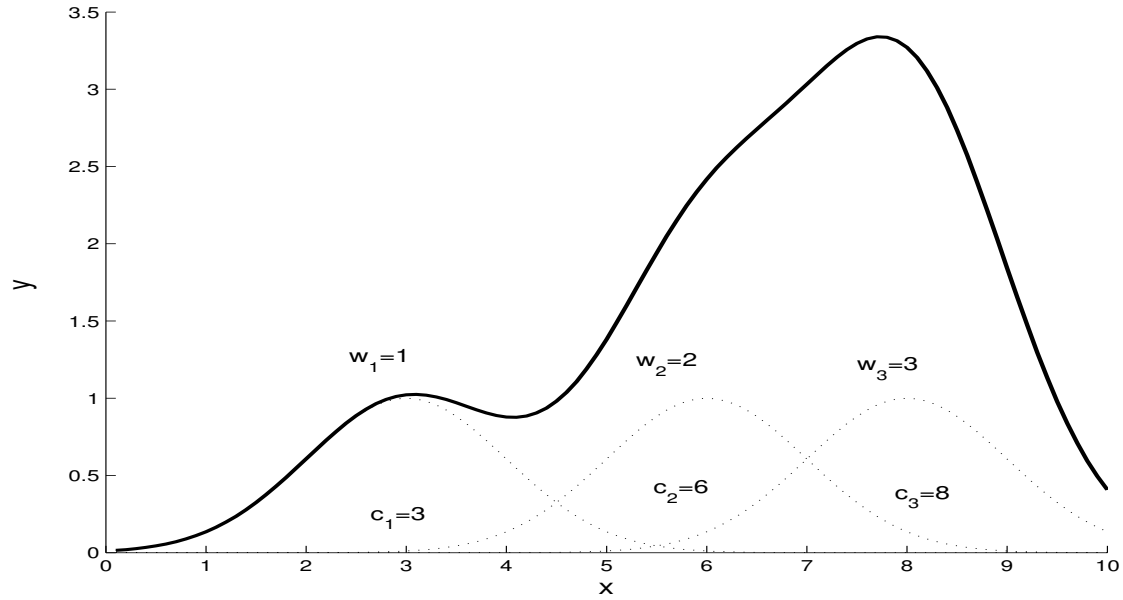
A set of n pairs $(\mathbf{x}_1, t_1)$,..., $(\mathbf{x}_n, t_n)$ is given, where $\mathbf{x}_i$ takes real values and t_i also takes values. The RBF neural network is trained using the data set. The objective is that $y(\mathbf{x}_i)$ is as close to t_i as possible.





Figures above; RBF functions for the same centers and different weights.





Figures above; RBF functions for same weights and different centers.

For example, suppose that we have a set of 10 data samples (x_1, t_1),..., (x_{10}, t_{10}) as given in the following Table. The data set was generated by using $t = \sin(2\pi x)$.

i	1	2	3	4	5
x_i	0.1	0.2	0.3	0.4	0.5
t_i	0.5878	0.9511	0.9511	0.5878	0.0000

i	6	7	8	9	10
x_i	0.6	0.7	0.8	0.9	1
t_i	-0.5878	-0.9511	-0.9511	-0.5878	0.0000

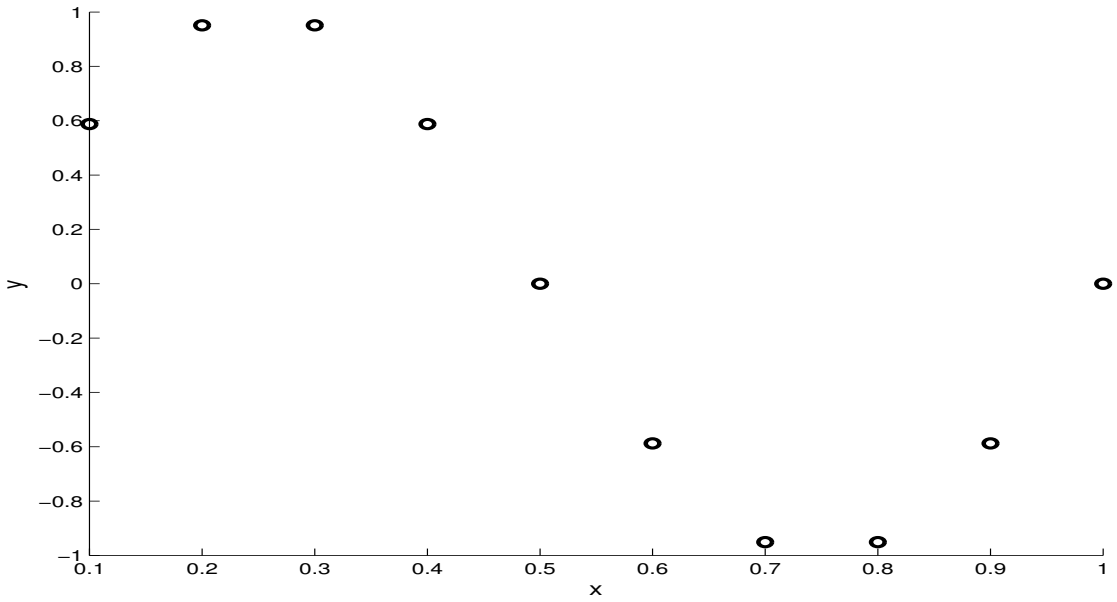


Figure: Graphical illustration of the data samples in above Table.

In general, the training of RBF neural networks includes the determination of locations of centers $\mathbf{c}_i$, and the estimation of the weights w_i. Note that σ also needs to be set, e.g. as a constant $\sigma = 1$.

We initially concentrate on the estimation of weights w_i, for given centers $\mathbf{c}_i$. The methods on the determination of centers will be discussed later.

For this example we preset 4 centers as $c_1 = 0.2$, $c_2 = 0.4$, $c_3 = 0.6$ and $c_4 = 0.8$. Here we also set $\sigma = 1$. This gives us four basis functions. $\exp\left(-\frac{(x-0.2)^2}{2}\right)$, $\exp\left(-\frac{(x-0.4)^2}{2}\right)$, $\exp\left(-\frac{(x-0.6)^2}{2}\right)$ and $\exp\left(-\frac{(x-0.8)^2}{2}\right)$.

Over the given ten data samples, form the matrix Φ given by

$$\Phi = \begin{pmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} & \phi_{1,4} \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} & \phi_{3,4} \\ \vdots & \vdots & \dots & \vdots \\ \phi_{9,1} & \phi_{9,2} & \phi_{9,3} & \phi_{9,4} \\ \phi_{10,1} & \phi_{10,2} & \phi_{10,3} & \phi_{10,4} \end{pmatrix}$$

with

$$\phi_{i,1} = \exp\left(-\frac{(x_i - 0.2)^2}{2}\right), \quad i = 1, 2, 3..., 10$$

$$\phi_{i,2} = \exp\left(-\frac{(x_i - 0.4)^2}{2}\right), \quad i = 1, 2, 3..., 10$$

$$\phi_{i,3} = \exp\left(-\frac{(x_i - 0.6)^2}{2}\right), \quad i = 1, 2, 3..., 10$$

$$\phi_{i,4} = \exp\left(-\frac{(x_i - 0.8)^2}{2}\right), \quad i = 1, 2, 3..., 10$$

We can write ten linear equations

$$\begin{cases} \phi_{1,1}w_1 + \phi_{1,2}w_2 + \phi_{1,3}w_3 + \phi_{1,4}w_4 = t_1 \\ \phi_{2,1}w_1 + \phi_{2,2}w_2 + \phi_{2,3}w_3 + \phi_{2,4}w_4 = t_2 \\ \phi_{3,1}w_1 + \phi_{3,2}w_2 + \phi_{3,3}w_3 + \phi_{3,4}w_4 = t_3 \\ \qquad\qquad\qquad\qquad \cdots\cdots\cdots \\ \phi_{10,1}w_1 + \phi_{10,2}w_2 + \phi_{10,3}w_3 + \phi_{10,3}w_4 = t_{10} \end{cases}$$

which is

$$\underbrace{\begin{pmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} & \phi_{1,4} \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} & \phi_{3,4} \\ \vdots & \vdots & \cdots & \vdots \\ \phi_{9,1} & \phi_{9,2} & \phi_{9,3} & \phi_{9,4} \\ \phi_{10,1} & \phi_{10,2} & \phi_{10,3} & \phi_{10,4} \end{pmatrix}}_{\Phi} \underbrace{\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix}}_{\mathbf{w}} = \underbrace{\begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_{10} \end{pmatrix}}_{\mathbf{t}}$$

or simply

$$\Phi\mathbf{w} = \mathbf{t}$$

There are 10 equations to solve 4 unknown parameters (Φ is not square matrix). There is no exact solution. The least squares estimate is calculated as

$$\mathbf{w} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{t}$$

It is found that

$$\mathbf{w} = [-3083.3, 8903.8, -8892.6, 3071.6]^T$$

The resultant RBF neural network is ready for prediction for any x

$$y(x) = \sum_{i=1}^{4} w_i \exp\left(-\frac{(x - c_i)^2}{2}\right)$$

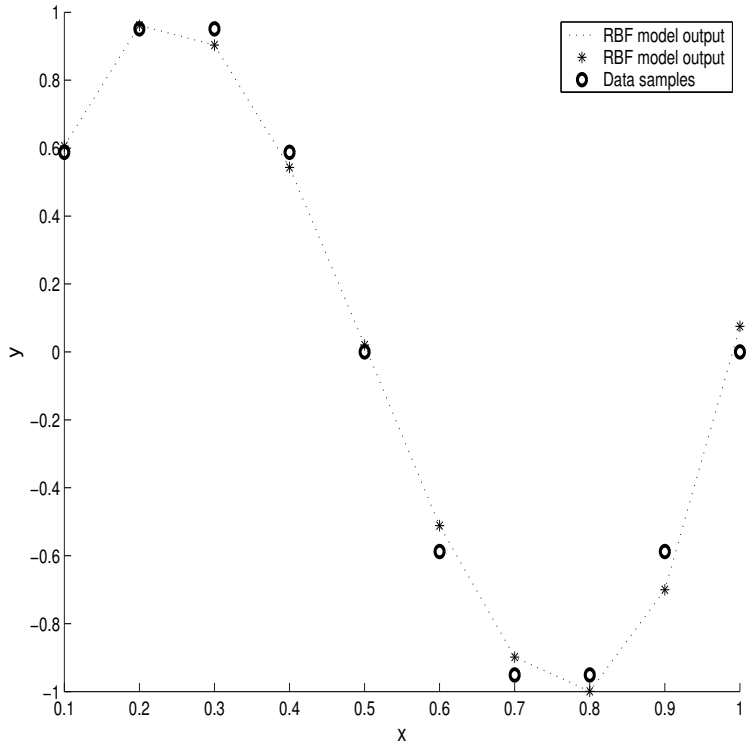


Figure: The results of resultant RBF neural network for curve fitting. Code $sinEX.m$ have been used.

A summary of the construction of RBF neural network for regression.

1. Determine the number and the centers $\mathbf{c}_i$;

2. Calculate $\phi_{i,j}$ for all training data samples;

3. Form matrix Φ and $\mathbf{t}$;

4. Calculate

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$$

5. Use the resultant RBF network

$$y(\mathbf{x}) = \sum_{i=1}^{M} w_i \exp\left(-\frac{(\|\mathbf{x} - \mathbf{c}_i\|)^2}{2\sigma^2}\right)$$

for prediction on any new sample $\mathbf{x}$.