

第7章 典型神经网络



7.1 单神经元网络

1 神经元模型

图7-1中 u_i 为神经元的内部状态, θ_i 为阈值, x_j 为输入信号, $j=1, \dots, n$, w_{ij} 为表示从单元 u_j 到单元 u_i 的连接权系数, s_i 为外部输入信号。

图7-1神经元模型可描述为:

7.1 单神经元网络

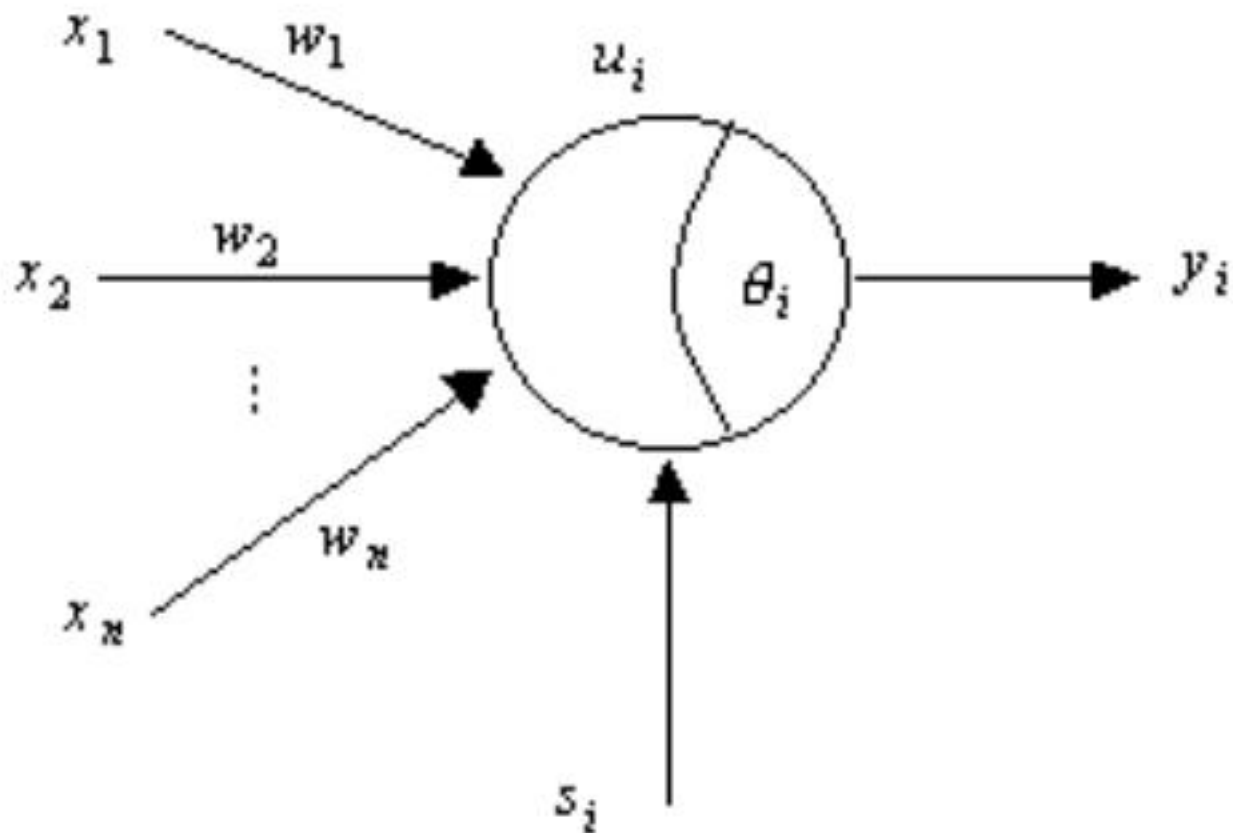


图7-1 单神经元模型

7.1 单神经元网络

$$Net_i = \sum_j w_{ij} x_j + s_i - \theta_i$$

$$u_i = f(Net_i)$$

$$y_i = g(u_i) = h(Net_i)$$

通常情况下，取 $g(u_i) = u_i$

即

$$y_i = f(Net_i)$$

7.1 单神经元网络

2.常用的神经元非线性特性有以下3种:

(1) 阈值型

$$f(Net_i) = \begin{cases} 1 & Net_i > 0 \\ 0 & Net_i \leq 0 \end{cases}$$

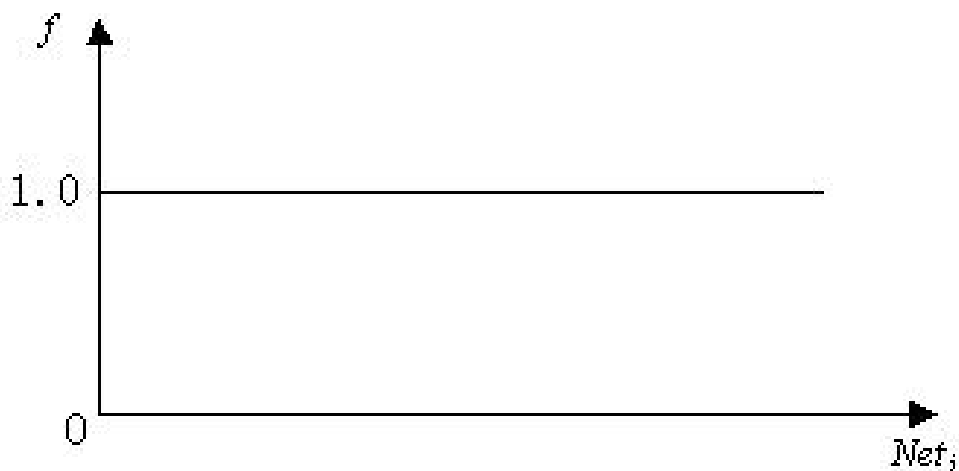


图7-2 阈值型函数

7.1 单神经元网络

(2) 分段线性型

$$f(Net_i) = \begin{cases} 0 & Net_i > Net_{i0} \\ kNet_i & Net_{i0} < Net_i < Net_{i1} \\ f_{\max} & Net_i \geq Net_{i1} \end{cases}$$

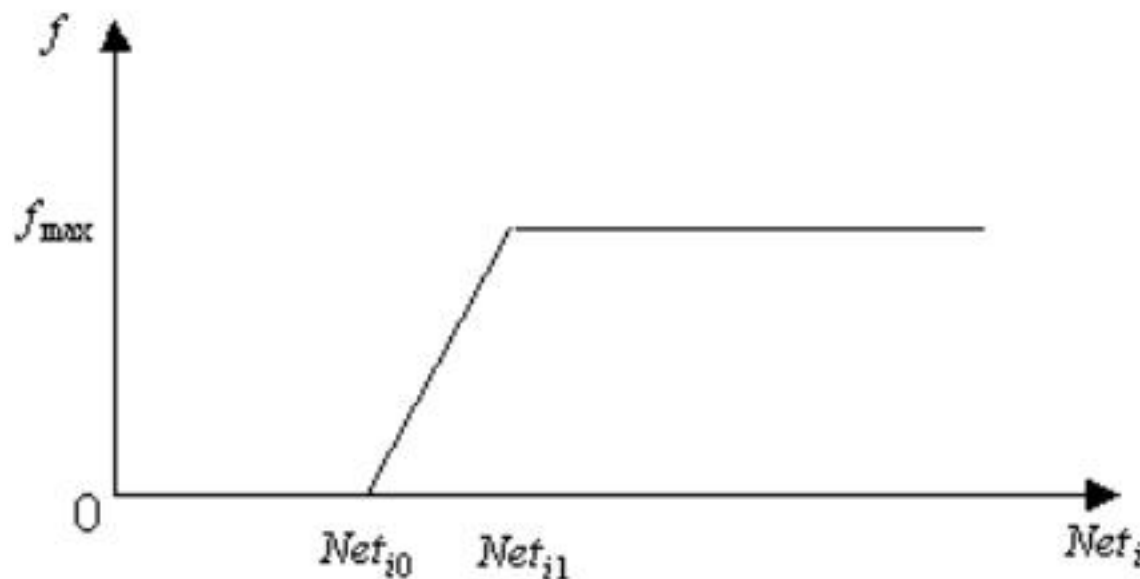


图7-3 分段线性函数

7.1 单神经元网络

(3) Sigmoid函数型

$$f(Net_i) = \frac{1}{1 + e^{-\frac{Net_i}{T}}}$$

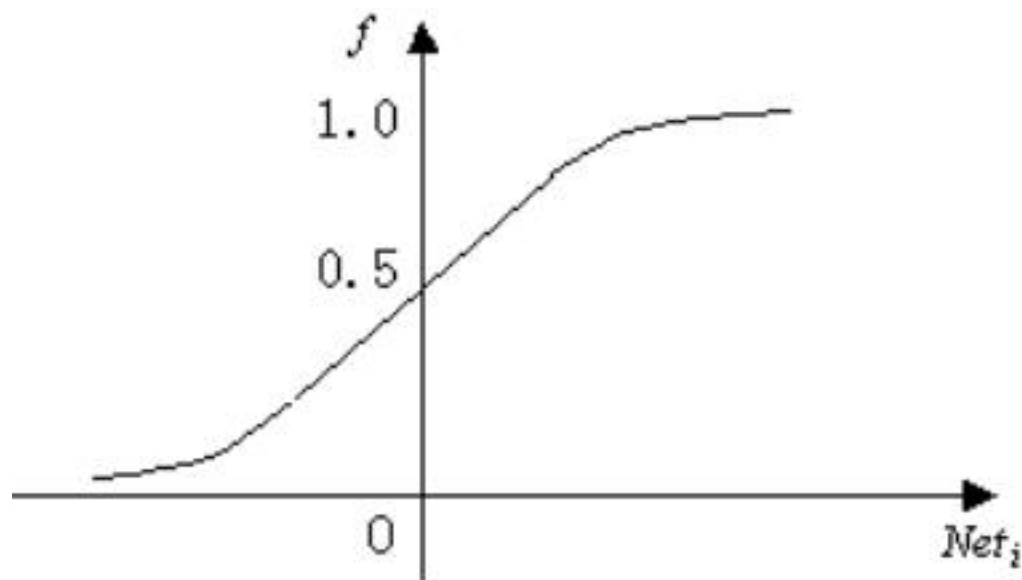


图7-4 Sigmoid函数

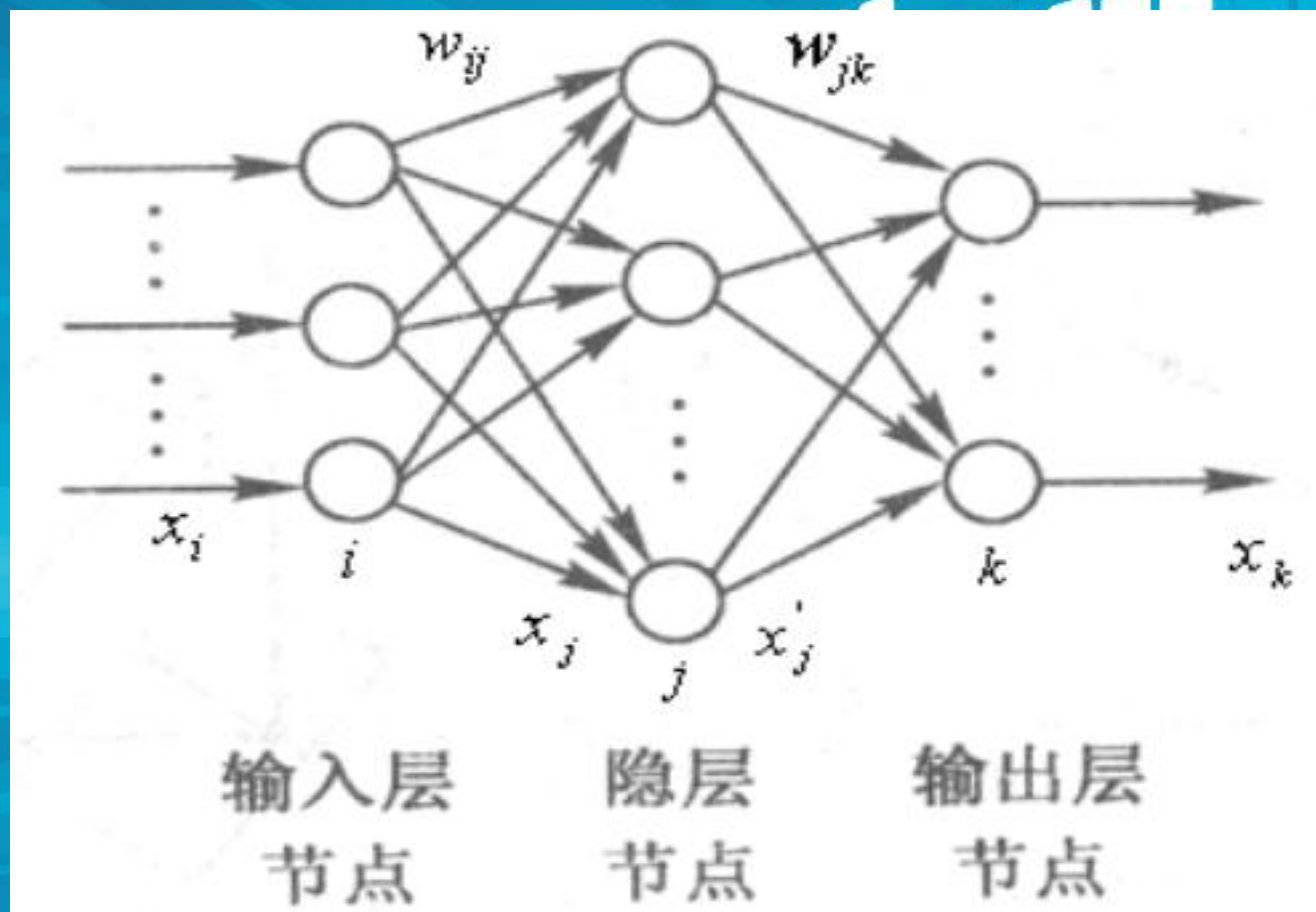
7.2 BP神经元网络

1986年，Rumelhart等提出了误差反向传播神经网络，简称BP网络（Back Propagation），该网络是一种单向传播的多层前向网络。

误差反向传播的BP算法简称BP算法，其基本思想是梯度下降法。它采用梯度搜索技术，以期使网络的实际输出值与期望输出值的误差均方值为最小。

7.2.1 BP网络结构

含一个隐含层的BP网络结构如图7-5所示，图中 i 为输入层神经元， j 为隐层神经元， k 为输出层神经元。



7.2.2 BP网络特点

- (1) 是一种多层网络，包括输入层、隐含层和输出层；
- (2) 层与层之间采用全互连方式，同一层神经元之间不连接；
- (3) 权值通过 δ 学习算法进行调节；
- (4) 神经元激发函数为S函数；
- (5) 学习算法由正向传播和反向传播组成；
- (6) 层与层的连接是单向的，信息的传播是双向的。

7.2.3 BP网络的逼近

BP网络逼近的结构如图7-6所示，图中 k 为网络的迭代步骤， $u(k)$ 和 $y(k)$ 为逼近器的输入。BP为网络逼近器， $y(k)$ 为被控对象实际输出， $y_n(k)$ 为BP的输出。将系统输出 $y(k)$ 及输入 $u(k)$ 的值作为逼近器BP的输入，将系统输出与网络输出的误差作为逼近器的调整信号。

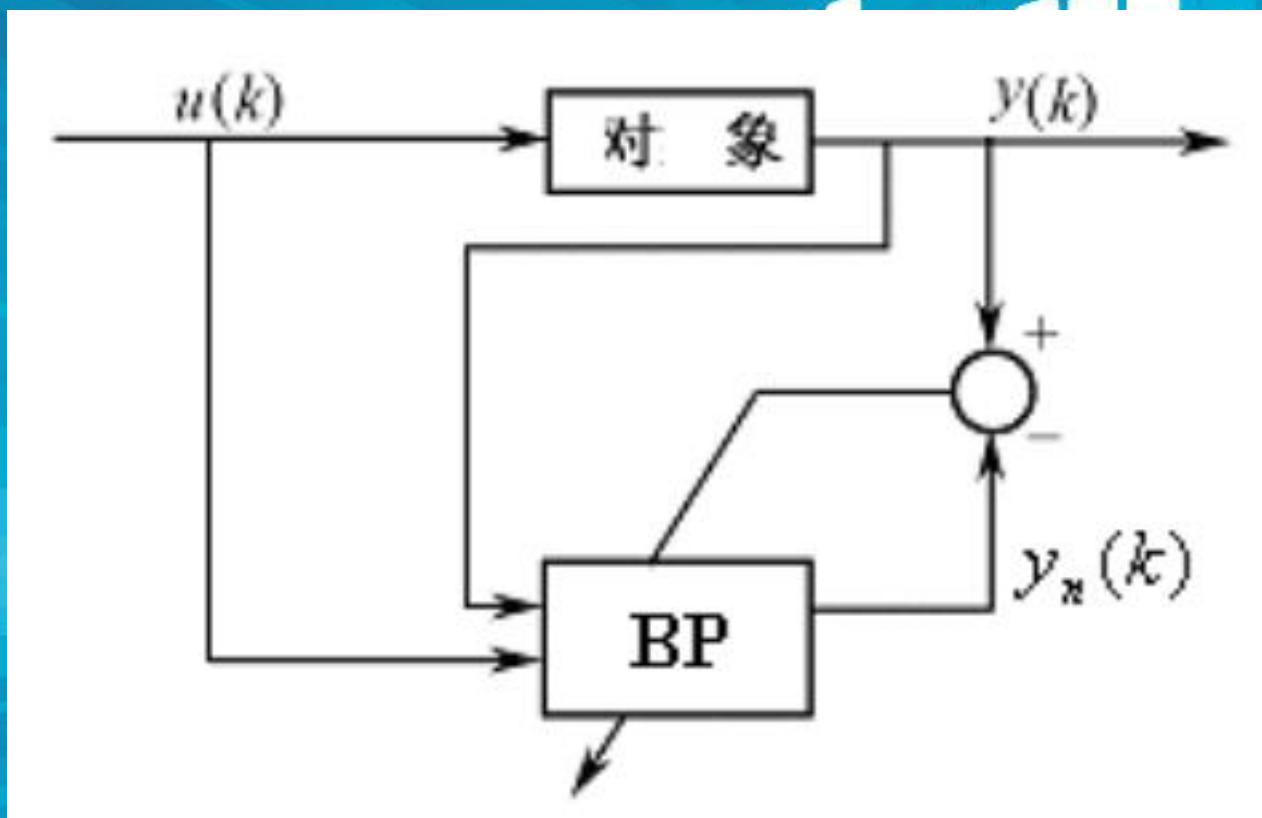


图7-6 BP神经网络逼近

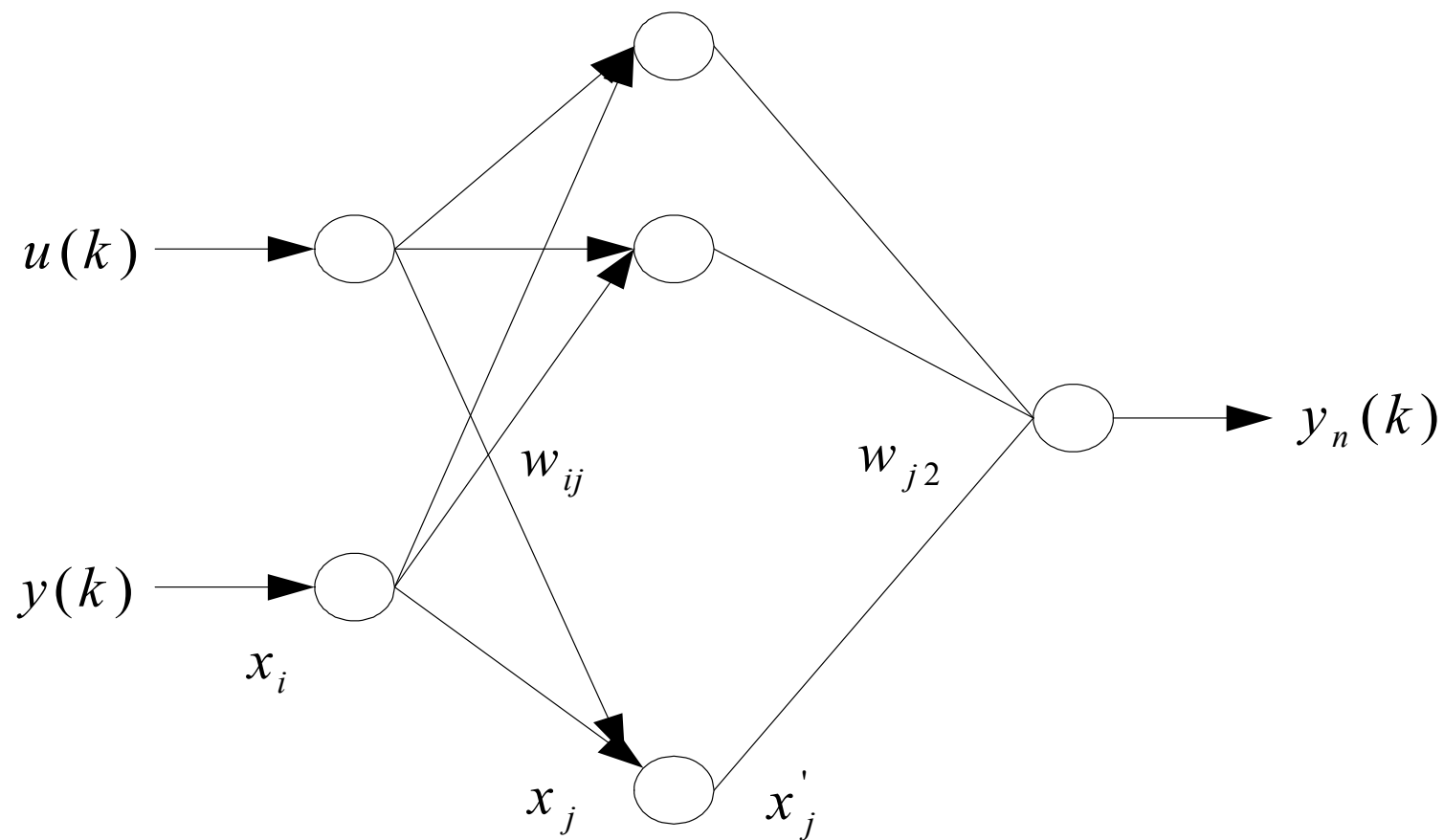


图7-7 用于逼近的BP网络

BP算法的学习过程由正向传播和反向传播组成。在正向传播过程中，输入信息从输入层经隐层逐层处理，并传向输出层，每层神经元（节点）的状态只影响下一层神经元的状态。如果在输出层不能得到期望的输出，则转至反向传播，将误差信号（理想输出与实际输出之差）按联接通路反向计算，由梯度下降法调整各层神经元的权值，使误差信号减小。

(1) 前向传播：计算网络的输出。

隐层神经元的输入为所有输入的加权之和：

$$x_j = \sum_i w_{ij} x_i$$

隐层神经元的输出采用S函数激发：

$$x'_j = f(x_j) = \frac{1}{1 + e^{-x_j}}$$

则

$$\frac{\partial x'_j}{\partial x_j} = x'_j (1 - x'_j)$$

输出层神经元的输出:

$$y_n(k) = \sum_j w_{j0} x'_j$$

网络输出与理想输出误差为:

$$e(k) = y(k) - y_n(k)$$

误差性能指标函数为:

$$E = \frac{1}{2} e(k)^2$$

(2) 反向传播：采用 δ 学习算法，调整各层间的权值。

根据梯度下降法，权值的学习算法如下：
输出层及隐层的连接权值学习算法为：

$$\Delta w_{j0} = -\eta \frac{\partial E}{\partial w_{j0}} = -\eta \cdot e(k) \cdot \frac{\partial y_n}{\partial w_{j0}} = -\eta \cdot e(k) \cdot x'_j$$

k+1时刻网络的权值为：

$$w_{j0}(k+1) = w_{j0}(k) + \Delta w_{j2}$$

隐层及输入层连接权值学习算法为：

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \cdot e(k) \cdot \frac{\partial y_n}{\partial w_{ij}}$$

其中

$$\frac{\partial y_n}{\partial w_{ij}} = \frac{\partial y_n}{\partial x'_j} \cdot \frac{\partial x'_j}{\partial x_j} \cdot \frac{\partial x_j}{\partial w_{ij}} = w_{j0} \cdot \frac{\partial x'_j}{\partial x_j} \cdot x_i = w_{j0} \cdot x'_j (1 - x'_j) \cdot x_i$$

k+1时刻网络的权值为：

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}$$

如果考虑上次权值对本次权值变化的影响，
需要加入动量因子 α ，此时的权值为：

$$w_{j0}(k+1) = w_{j0}(k) + \Delta w_{j0} + \alpha(w_{j0}(k) - w_{j0}(k-1))$$

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij} + \alpha(w_{ij}(k) - w_{ij}(k-1))$$

其中， α 为动量因子。 $\alpha \in [0,1]$

Jacobian 阵(即为对象的输出对控制输入的灵敏度信息)可由神经网络辨识得到, 其算法为:

$$\frac{\partial y(k)}{\partial u(k)} \approx \frac{\partial y_n(k)}{\partial u(k)} = \frac{\partial y_n(k)}{\partial x'_j} \times \frac{\partial x'_j}{\partial x_j} \times \frac{\partial x_j}{\partial x_1} = \sum_j w_{j0} x'_j (1 - x'_j) w_{1j}$$

其中取 $x_1 = u(k)$

7.2.4 BP网络的优缺点

BP网络的优点为：

- (1) 只要有足够多的隐层和隐层节点，BP网络可以逼近任意的非线性映射关系；
- (2) BP网络的学习算法属于全局逼近算法，具有较强的泛化能力。
- (3) BP网络输入输出之间的关联信息分布地存储在网络的连接权中，个别神经元的损坏只对输入输出关系有较小的影响，因而BP网络具有较好的容错性。

BP网络的主要缺点为：

- （1）待寻优的参数多，收敛速度慢；
- （2）目标函数存在多个极值点，按梯度下降法进行学习，很容易陷入局部极小值；
- （3）难以确定隐层及隐层节点的数目。目前，如何根据特定的问题来确定具体的网络结构尚无很好的方法，仍需根据经验来试凑。

- 由于BP网络具有很好的逼近非线性映射的能力，该网络在模式识别、图像处理、系统辨识、函数拟合、优化计算、最优预测和自适应控制等领域有着较为广泛的应用。
- 由于BP网络具有很好的逼近特性和泛化能力，可用于神经网络控制器的设计。但由于BP网络收敛速度慢，难以适应实时控制的要求。

7.2.5 BP网络逼近仿真实例

使用BP网络逼近对象：

$$y(k) = u(k)^3 + \frac{y(k-1)}{1 + y(k-1)^2}$$

BP网络逼近程序见chap7_1.m

7.2.6 BP网络模式识别

- 由于神经网络具有自学习、自组织和并行处理等特征，并具有很强的容错能力和联想能力，因此，神经网络具有模式识别的能力。

- 在神经网络模式识别中，根据标准的输入输出模式对，采用神经网络学习算法，以标准的模式作为学习样本进行训练，通过学习调整神经网络的连接权值。当训练满足要求后，得到的神经网络权值构成了模式识别的知识库，利用神经网络并行推理算法对所需要的输入模式进行识别。

- 当待识别的输入模式与训练样本中的某个输入模式相同时，神经网络识别的结果就是与训练样本中相对应的输出模式。当待识别的输入模式与训练样本中所有输入模式都不完全相同时，则可得到与其相近样本相对应的输出模式。当待识别的输入模式与训练样本中所有输入模式相差较远时，就不能得到正确的识别结果，此时可将这一模式作为新的样本进行训练，使神经网络获取新的知识，并存储到网络的权值矩阵中，从而增强网络的识别能力。

BP网络的训练过程如下：正向传播是输入信号从输入层经隐层传向输出层，若输出层得到了期望的输出，则学习算法结束；否则，转至反向传播。

以第 p 个样本为例，用于训练的BP网络结构如图7-11所示。

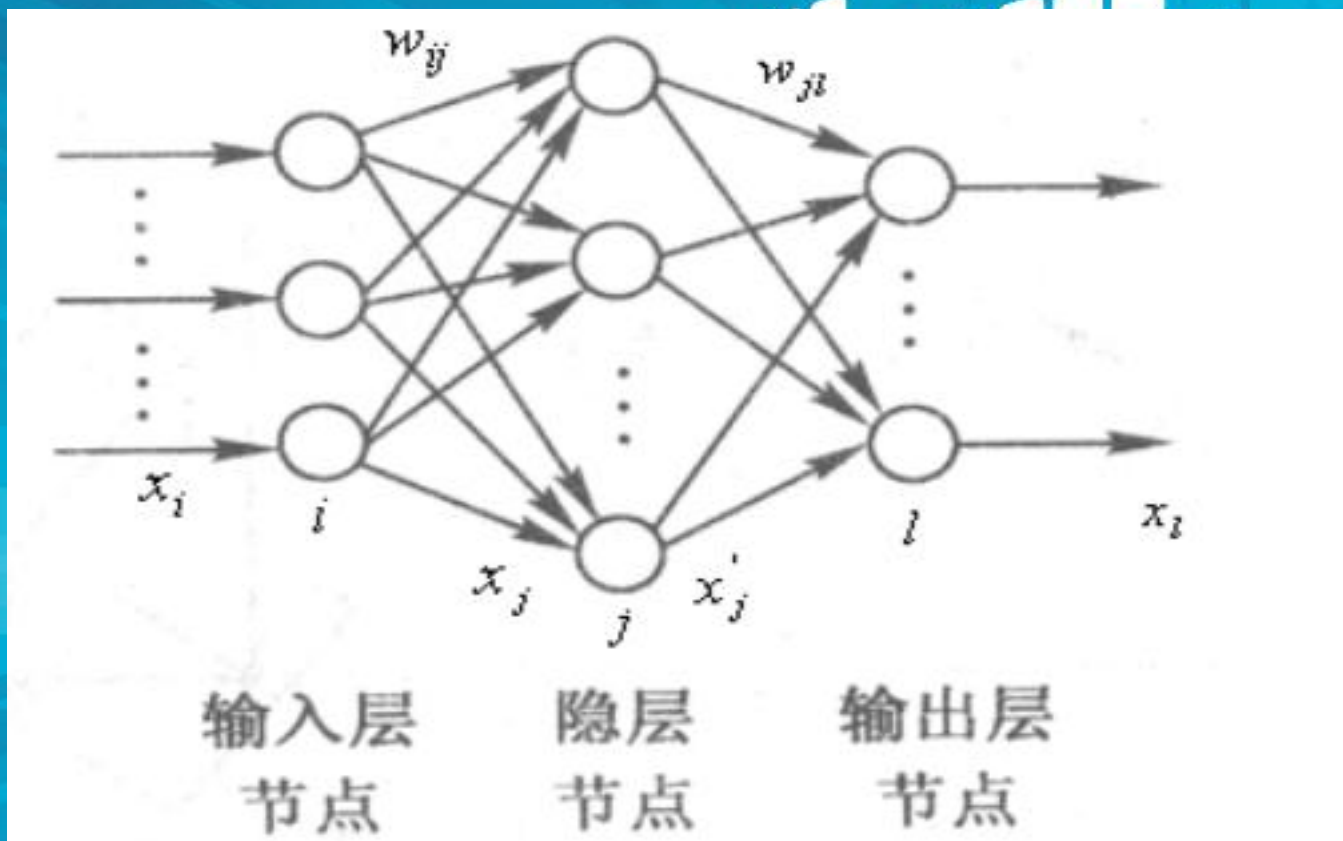


图7-11 BP神经网络结构

网络的学习算法如下：

(1) 前向传播：计算网络的输出。

隐层神经元的输入为所有输入的加权之和：

$$x_j = \sum_i w_{ij} x_i$$

隐层神经元的输出 x'_j 采用S函数激发 x_j ，

得

$$x'_j = f(x_j) = \frac{1}{1 + e^{-x_j}}$$

$$\frac{\partial x'_j}{\partial x_j} = x'_j (1 - x'_j)$$

输出层神经元的输出：

$$x_l = \sum_j w_{jl} x'_j$$

网络第 l 个输出与相应理想输出 x_l^0 的误差为：

$$e_l = x_l^0 - x_l$$

第 p 个样本的误差性能指标函数为：

$$E_p = \frac{1}{2} \sum_{l=1}^N e_l^2$$

其中 N 为网络输出层的个数。

(2) 反向传播：采用梯度下降法，调整各层间的权值。权值的学习算法如下：

输出层及隐层的连接权值 w_{jl} 学习算法为：

$$\Delta w_{jl} = -\eta \frac{\partial E_p}{\partial w_{jl}} = \eta e_l \frac{\partial x_l}{\partial w_{jl}} = \eta e_l x'_j$$

$$w_{jl}(k+1) = w_{jl}(k) + \Delta w_{jl}$$

$$\Delta w_{ij} = -\eta \frac{\partial E_p}{\partial w_{ij}} = \eta \sum_{l=1}^N e_l \frac{\partial x_l}{\partial w_{ij}}$$

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}$$

$$\frac{\partial x_l}{\partial w_{ij}} = \frac{\partial x_l}{\partial x'_j} \cdot \frac{\partial x'_j}{\partial x_j} \cdot \frac{\partial x_j}{\partial w_{ij}} = w_{il} \cdot \frac{\partial x'_j}{\partial x_j} \cdot x_i = w_{jl} \cdot x'_j (1 - x'_j) \cdot x_i$$

如果考虑上次权值对本次权值变化的影响，
需要加入动量因子 α ，此时的权值为：

$$w_{jl}(k+1) = w_{jl}(k) + \Delta w_{jl} + \alpha(w_{jl}(k) - w_{jl}(k-1))$$

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij} + \alpha(w_{ij}(k) - w_{ij}(k-1))$$

其中 α 为动量因子。 $\alpha \in [0,1]$

表7-1 训练样本

输 入			输 出	
1	0	0	1	0
0	1	0	0	0.5
0	0	1	0	1

BP 网络模式识别程序包括网络训练程序
chap7_2a.m和网络测试程序chap7_2b.m。

7.3 RBF神经网络

径向基函数(RBF-Radial Basis Function)神经网络是由J. Moody和C. Darken在80年代末提出的一种神经网络,它是具有单隐层的三层前馈网络。由于它模拟了人脑中局部调整、相互覆盖接收域(或称感受野-Receptive Field)的神经网络结构,因此,RBF网络是一种局部逼近网络,已证明它能任意精度逼近任意连续函数。

RBF网络特点

- (1) RBF网络的作用函数为高斯函数，是局部的，BP网络的作用函数为S函数，是全局的；
- (2) 如何确定RBF网络隐层节点的中心及基宽度参数是一个困难的问题；
- (3) 已证明RBF网络具有唯一最佳逼近的特性，且无局部极小。

7.3.1 RBF网络结构与算法

多输入单输出的RBF神经网络结构如图7-13所示。

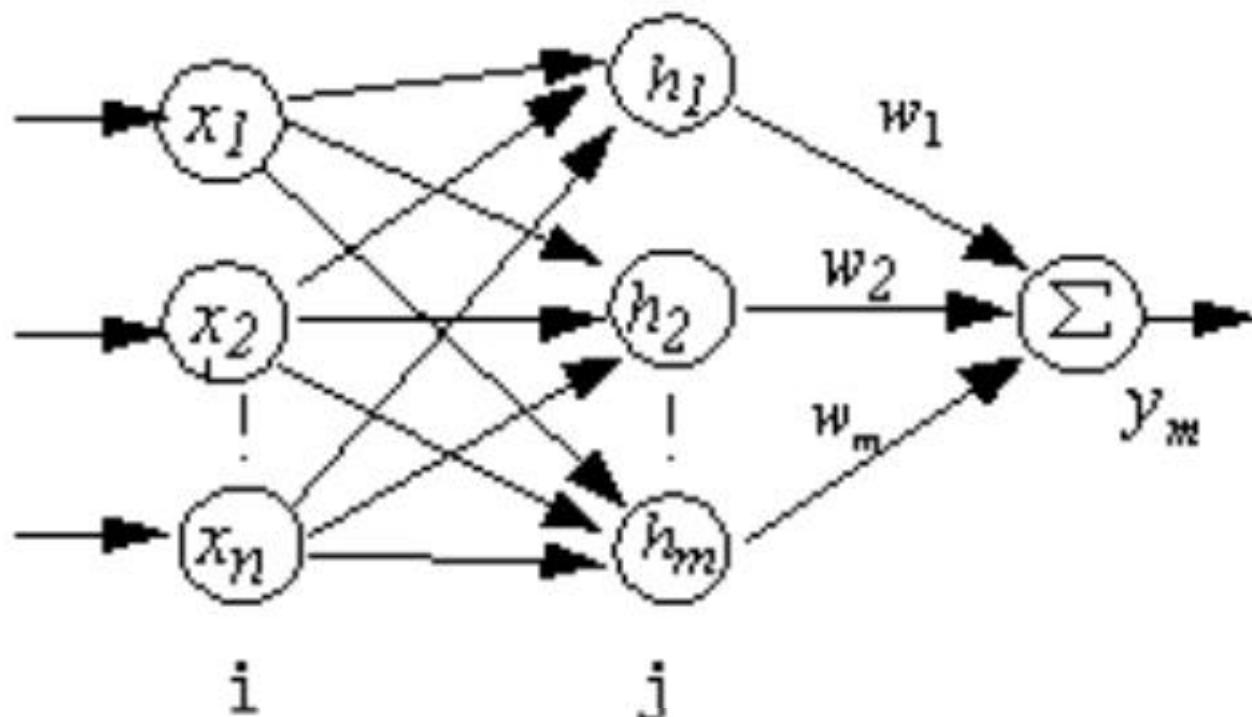


图7-13 RBF网络结构

h_j 在RBF神经网络中, $x = [x_1 \ x_2 \ \cdots \ x_n]^T$ 为网络输入, 为隐含层第 j 个神经元的输出, 即

$$h_j = \exp\left(-\frac{\|x - c_j\|^2}{2b_j^2}\right), j = 1, 2, \cdots m \quad (7.20)$$

式中, $c_j = [c_{j1}, \cdots, c_{jn}]$ 为第 j 个隐层神经元的中心点矢量值。

高斯基函数的宽度矢量为 $b = [b_1, \cdots, b_m]^T$

其中 $b_j > 0$ 为隐含层神经元 j 的高斯基函数的宽度。

网络的权值为

$$\boldsymbol{w} = [w_1, \dots, w_m]^T \quad (7.21)$$

RBF网络的输出为

$$y_m(t) = w_1 h_1 + w_2 h_2 + \dots + w_m h_m \quad (7.22)$$

由于RBF网络只调节权值，因此，RBF网络较BP网络有算法简单、运行时间快的优点。但由于RBF网络中，输入空间到输出空间是非线性的，而隐含空间到输出空间是线性的，因而其非线性能力不如BP网络。

7.3.2 RBF网络设计实例

(1) 结构为1-5-1的RBF网络

考虑结构为1-5-1的RBF网络，取网络输入 $x = x_1$ 为

$$\mathbf{b} = [b_1 \ b_2 \ b_3 \ b_4 \ b_5]^T \quad \mathbf{c} = [c_{11} \ c_{12} \ c_{13} \ c_{14} \ c_{15}]$$

令

$$\mathbf{h} = [h_1 \ h_2 \ h_3 \ h_4 \ h_5]^T \quad \mathbf{w} = [w_1 \ w_2 \ w_3 \ w_4 \ w_5]^T$$

则网络输出为 $y_m(t) = \mathbf{w}^T \mathbf{h} = w_1 h_1 + w_2 h_2 + w_3 h_3 + w_4 h_4 + w_5 h_5$

取网络的输入为sint 时，网络的输出如图7-14所示，网络隐含层的输出如图7-15所示。仿真程序为chap7_3sim.mdl。

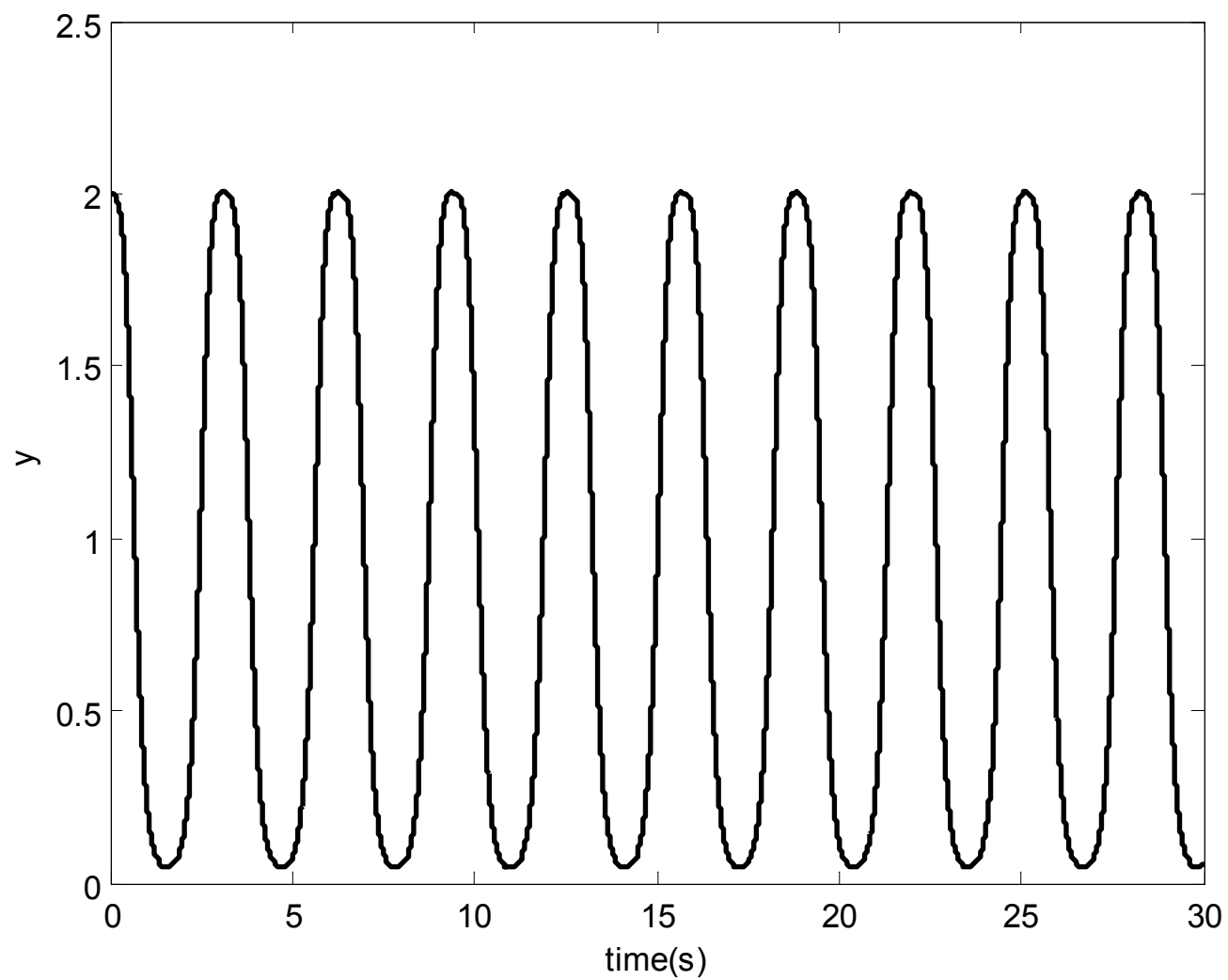


图7-14 RBF网络输出

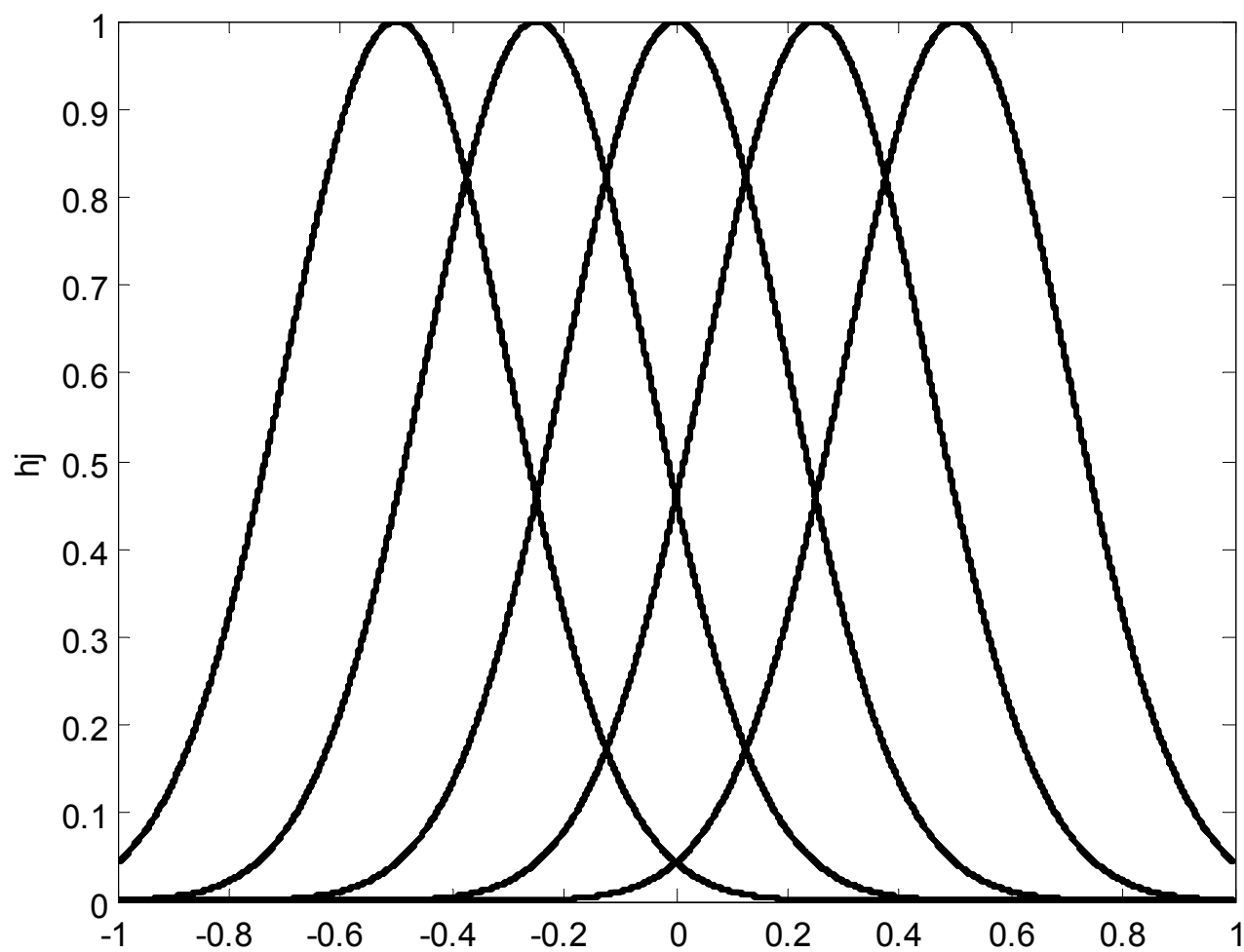


图7-15 RBF网络隐层的输出

(2) 结构为2-5-1的RBF网络

考虑结构为2-5-1的RBF网络，取网络输入为 $\mathbf{x} = [x_1, x_2]^T$

令

$$\mathbf{b} = [b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5]^T \quad \mathbf{c} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{25} \end{bmatrix}$$

$$\mathbf{h} = [h_1 \quad h_2 \quad h_3 \quad h_4 \quad h_5]^T \quad \mathbf{w} = [w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5]^T$$

网络输出为 $y_m(t) = \mathbf{w}^T \mathbf{h} = w_1 h_1 + w_2 h_2 + w_3 h_3 + w_4 h_4 + w_5 h_5$

取网络的输入为sint 时，网络的输出如图7-16所示，网络隐含层的输出如图7-17和图7-18所示。仿真程序为chap7_4sim.mdl。

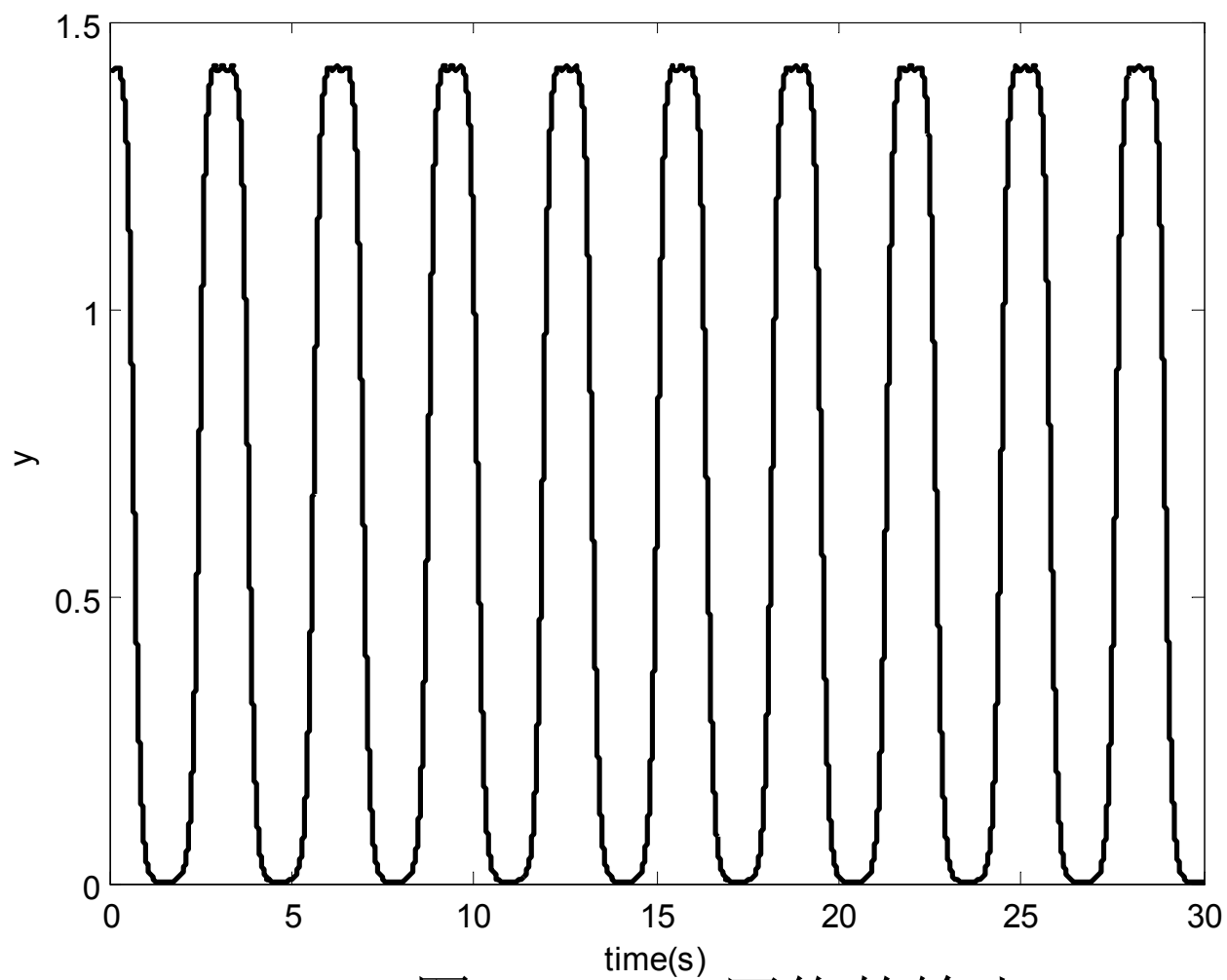


图7-16 RBF网络的输出

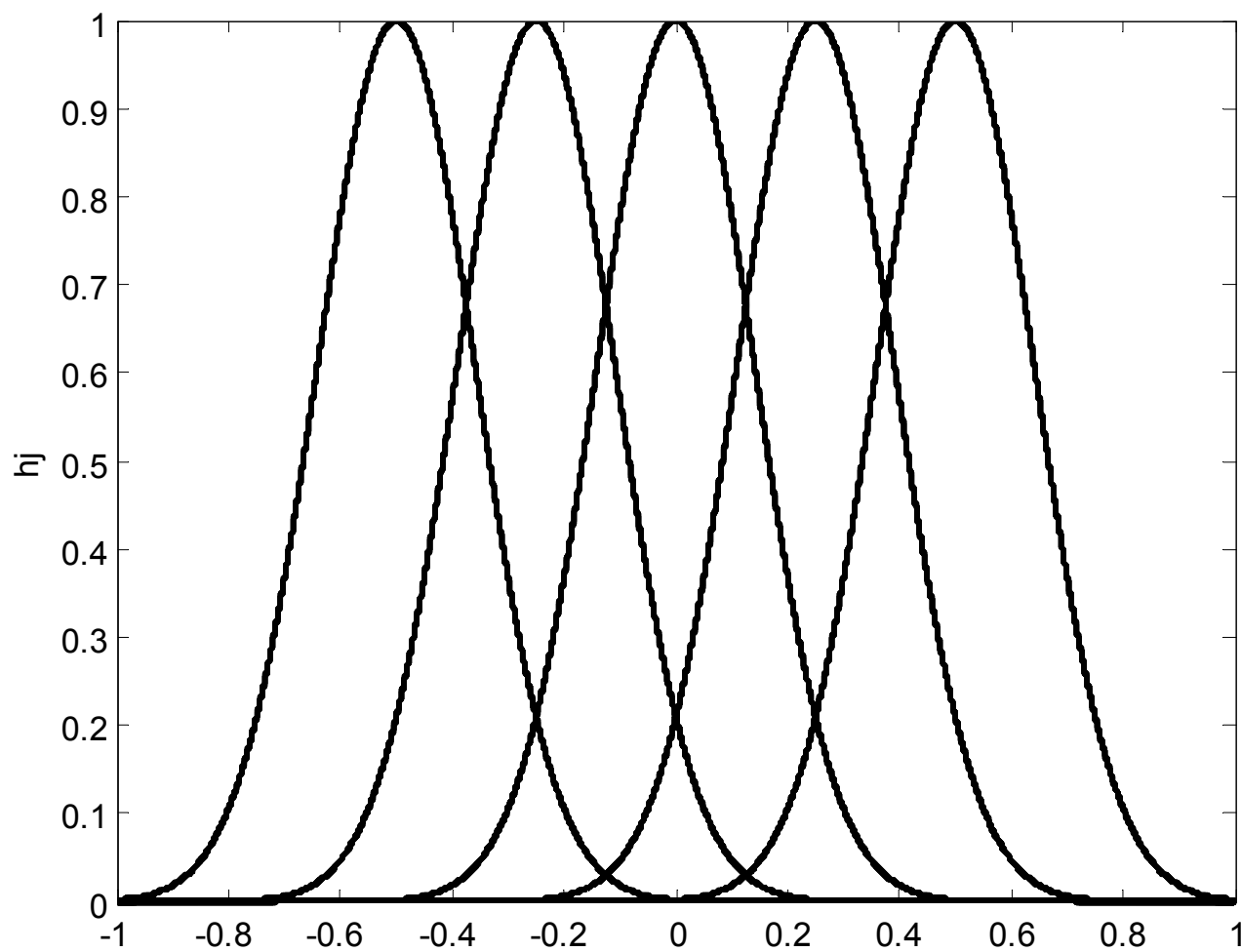


图7-17 第一个输入的隐层神经网络输出

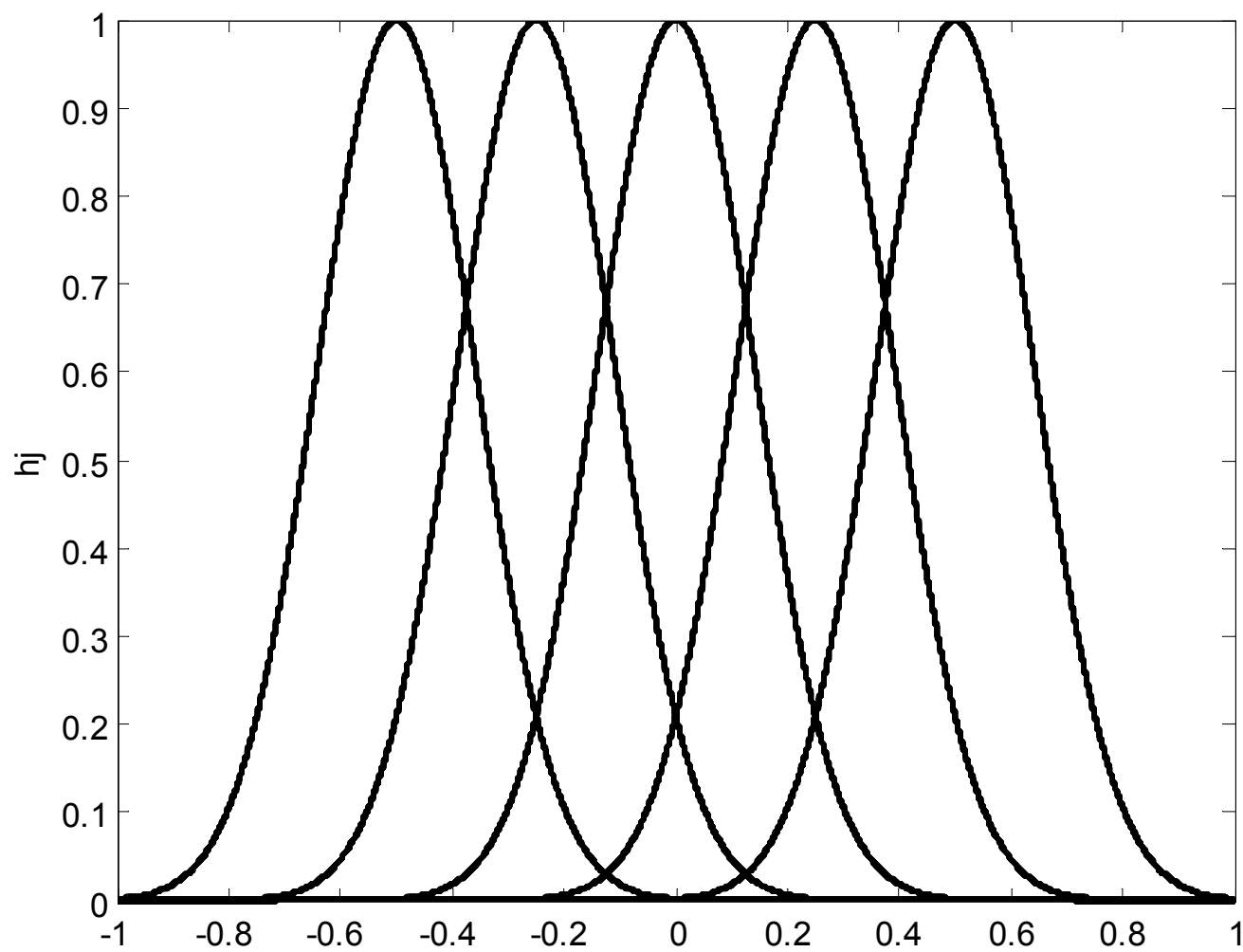
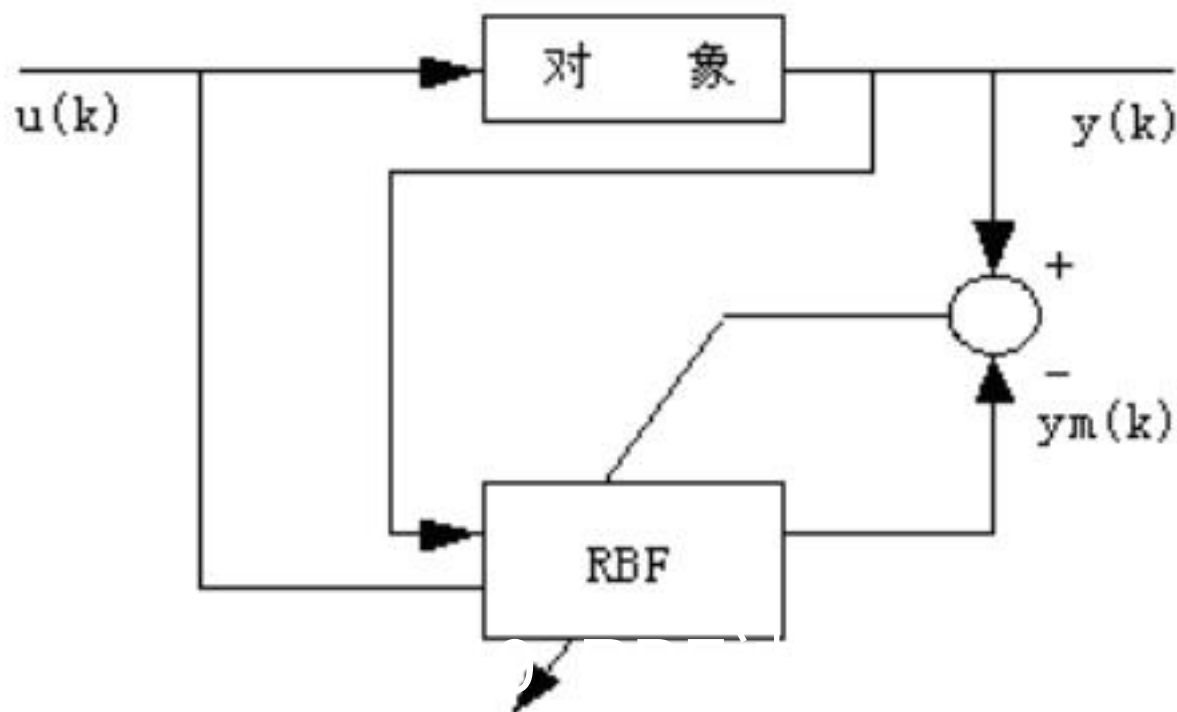


图7-18 第二个输入的隐层神经网络输出

7.3.3 RBF网络的逼近

7.3.3.1 基本原理

采用RBF网络对模型进行逼近，结构如图7-19所示。



网络逼近的误差指标为

$$E(t) = \frac{1}{2} (y(t) - y_m(t))^2 \quad (7.23)$$

根据梯度下降法，权值按以下方式调节

$$\Delta w_j(t) = -\eta \frac{\partial E}{\partial w_j} = \eta (y(t) - y_m(t)) h_j \quad (7.24)$$

$$w_j(t) = w_j(t-1) + \Delta w_j(t) + \alpha (w_j(t-1) - w_j(t-2))$$

其中 $\eta \in (0,1)$ 为学习速率， $\alpha \in (0,1)$ 为动量因子。

在RBF网络设计中，需要注意的是将 c_j 和 b 值设计在网络输入有效的映射范围内，否则高斯基函数将不能保证实现有效的映射，导致RBF网络失效。如果将 c_j 和 b 的初始值设计在有效的映射范围内，则只调节网络的权值便可实现RBF网络的有效学习。

7.3.3.2 仿真实例

实例1:连续系统

采用RBF网络对如下模型进行逼近

$$G(s) = \frac{133}{s^2 + 25s}$$

网络结构为2-5-1，取 $x(1)=u(t)$ ， $x(2)=y(t)$ ， $\alpha = 0.05$
 $\eta = 0.5$ 。网络的初始权值取0至1之间的随机值。考虑到网络的第一个输入范围为 $[0, 1]$ ，离线测试可得第二个输入范围为 $[0, 10]$ ，取高斯基函数的参数取值为

$$c_j = \begin{bmatrix} -1 & -0.5 & 0 & 0.5 & 1 \\ -10 & -5 & 0 & 5 & 10 \end{bmatrix}^T \quad b_j = 1.5 \quad j = 1, 2, 3, 4, 5$$

网络的第一个输入为 $u(t)=\sin t$ ，仿真中，只调节权值 w ，取固定的 c_j 和 b ，仿真结果如图7-20所示。仿真程序为chap7_5.m。

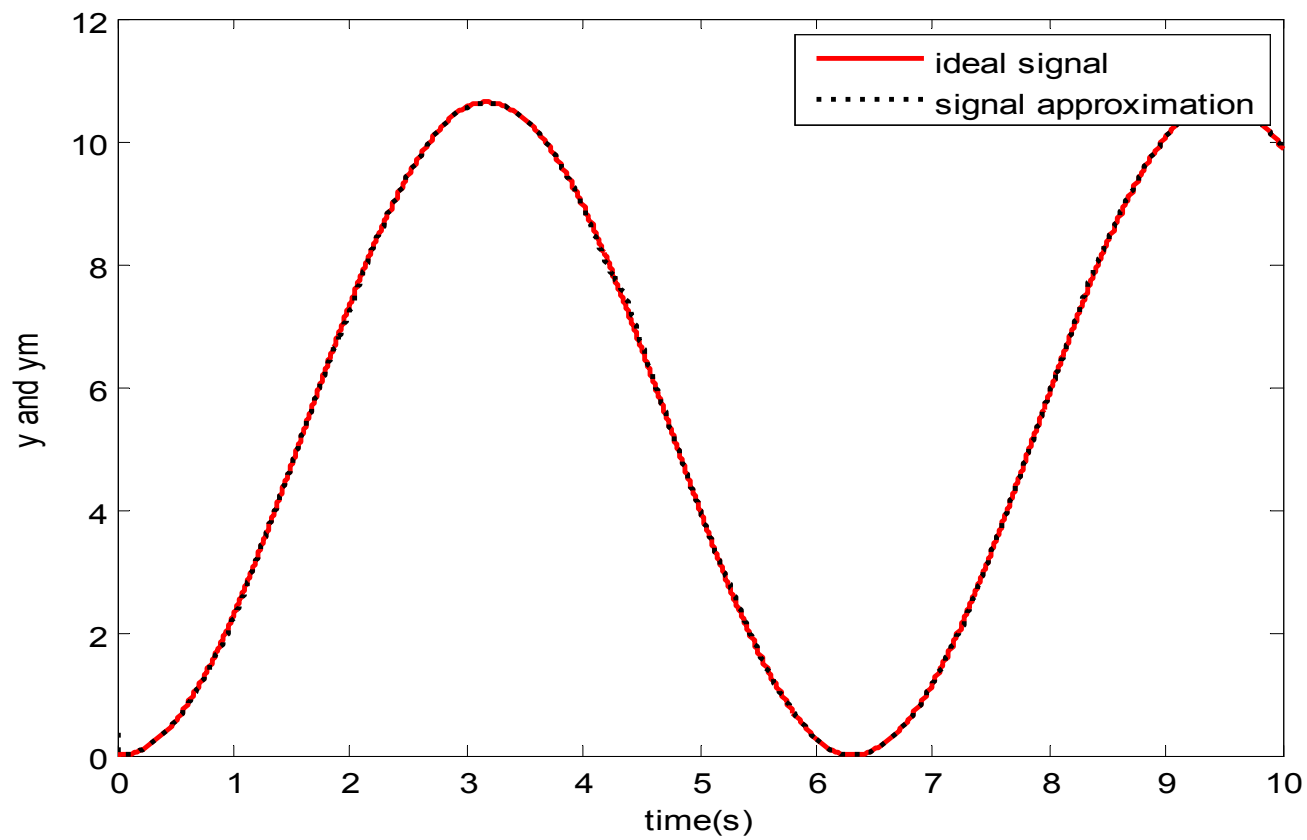


图7-20 基于权值调节的RBF网络逼近

实例2:离散系统

采用RBF网络对如下离散模型进行逼近

$$y(k) = u(k)^3 + \frac{y(k-1)}{1 + y(k-1)^2}$$

网络的第一个输入为 $u(t)=\sin t$ ， $t = k \times T$ ， $T=0.001$ 。

网络结构为2-5-1，取 $x(1)=u(t)$ ， $x(2)=y(t)$ ， $\alpha = 0.05$ ， $\eta = 0.15$ 。网络的初始权值取0至1之间的随机值。考虑到网络的第一个输入范围为 $[0,1]$ ，离线测试可得第二个输入范围为 $[0,10]$ ，取高斯基函数的参数取值为

$$\mathbf{c}_j = \begin{bmatrix} -1 & -0.5 & 0 & 0.5 & 1 \end{bmatrix}^T, \mathbf{b}_j = 3.0, j = 1, 2, 3, 4, 5。$$

仿真中，调节权值 w ，取固定的 \mathbf{c}_j 和 \mathbf{b} ，仿真结果如图7-21所示。由仿真结果可见，采用梯度下降法可实现很好的逼近效果，其中高斯基函数的参数值 \mathbf{c}_j 和 \mathbf{b} 的取值很重要。仿真程序为chap7_6.m。

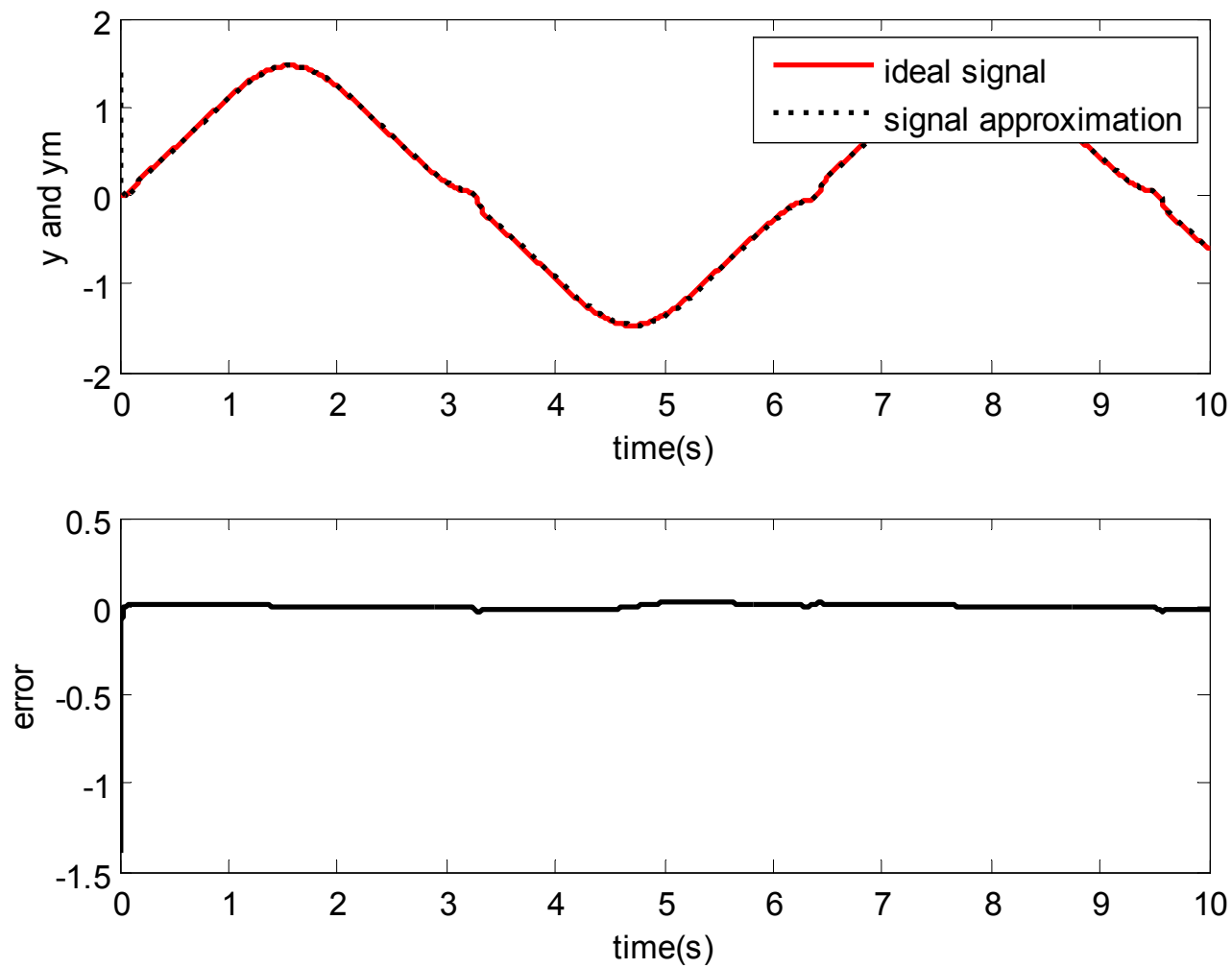


图7-21 基于权值调节的RBF网络逼近

7.3.4 高斯基函数的参数对RBF网络逼近的影响

由高斯函数的表达式可知，高斯基函数受参数 c_j 和 b_j 的影响， c_j 和 b_j 的设计原则如下：

(1) b_j 为隐含层第 j 个神经元高斯基函数的宽度。 b_j 值越大，表示高斯基函数越宽。高斯基函数宽度是影响网络映射范围的重要因素，高斯基函数越宽，网路对输入的映射能力越大，否则，网路对输入的映射能力越小。一般将 b_j 值设计为适中的值。

(2) c_j 为隐含层第 j 个神经元高斯基函数中心点的坐标向量。 c_j 值离输入越近，高斯函数对输入越敏感，否则，高斯函数对输入越不敏感；

(3) 中心点坐标向量 c_j 应使高斯基函数在有效的输入映射范围内。例如，RBF网络输入为 $[-3, +3]$ ，则 c_j 为 $[-3, +3]$ 。

仿真中，应根据网络输入值得范围来设计 c_j 和 b_j ，从而保证有效的高斯基函数映射，如图7-22为 5 个高斯基函数。仿真程序为chap7_7.m。

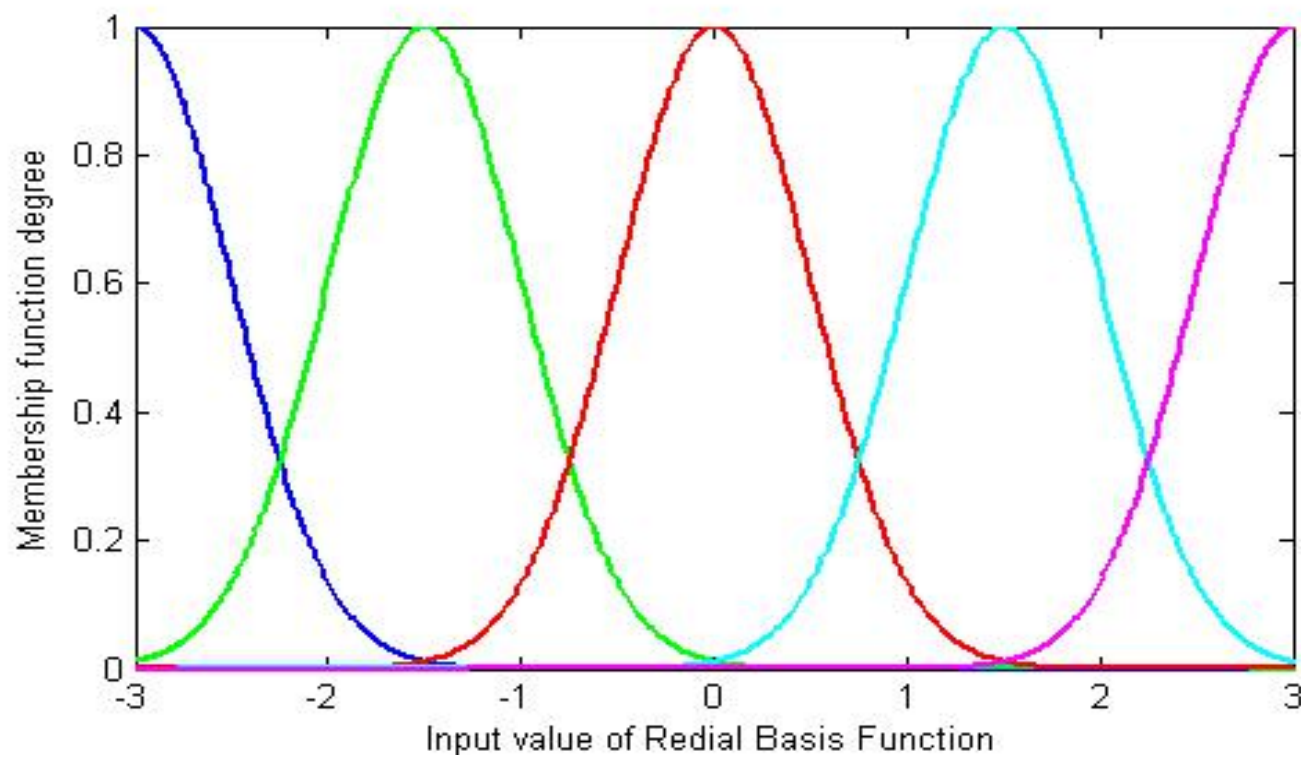


图7-22 5个高斯隶属函数

采用RBF网络对如下离散模型进行逼近

$$y(k) = u(k)^3 + \frac{y(k-1)}{1 + y(k-1)^2}$$

仿真中，取RBF网络输入为 $0.5 \sin(2\pi t)$ ，网络结构为2-5-1，通过改变高斯基函数 c_j 和 b_j 值，可分析 c_j 和 b_j 对RBF网络逼近性能的影响，具体说明如下：

- (1) 合适的 b_j 和 c_j 值对RBF网络逼近的影响 ($M_b=1, M_c=1$);
- (2) 不合适的 b_j 和 c_j 合适的 值对RBF网络逼近的影响 ($M_b=2, M_c=1$);

(3) 合适的 b_j 与不合适的 c_j 值对RBF网络逼近的影响 ($M_b=1$, $M_c=2$);

(4) 不合适的 b_j 和 c_j 值对RBF网络逼近的影响 ($M_b=2$, $M_c=2$)。

仿真结果如图7-23至图7-26所示，由仿真结果可见，如果选取的参数 c_j 和 b_j 不合适，RBF网络逼近性能将得不到保证。仿真程序为chap7_8.m，程序见附件。

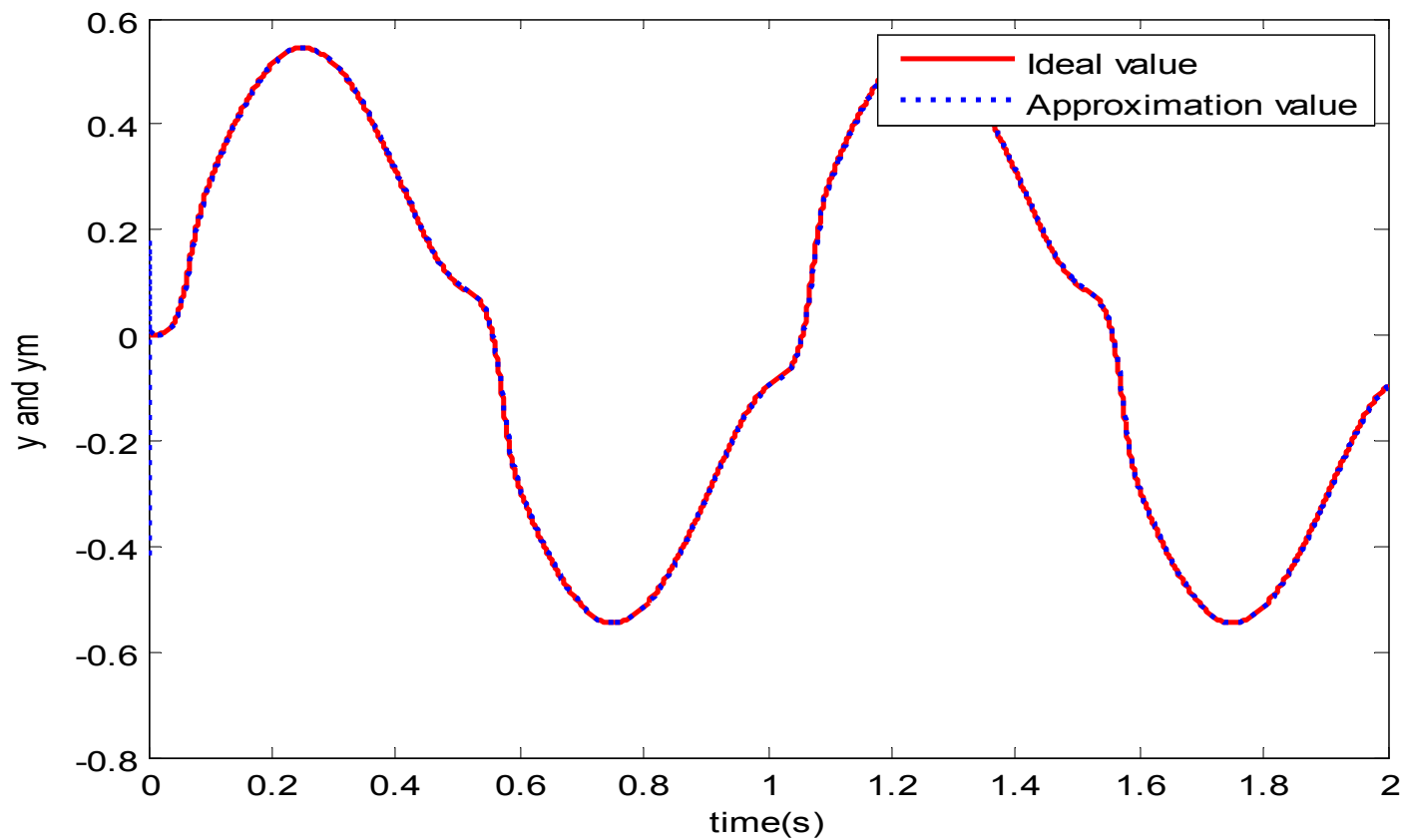


图7-23 合适 b_j 与 c_j 值的RBF网络逼近RBF ($M_b=1$, $M_c=1$)

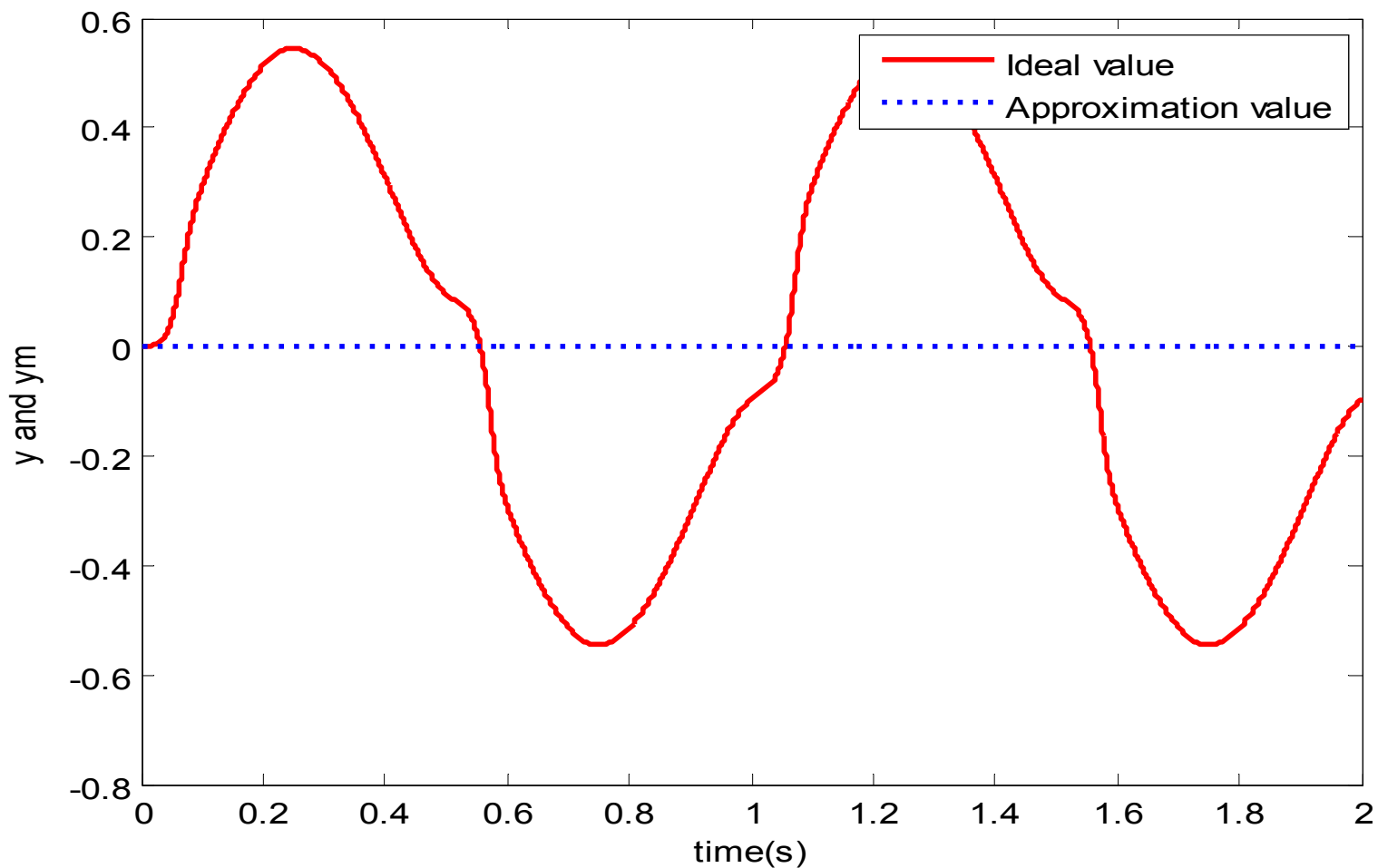


图7-24 不合适 b_j 与合适 c_j 值的RBF网络逼近 ($M_b=2$, $M_c=1$)

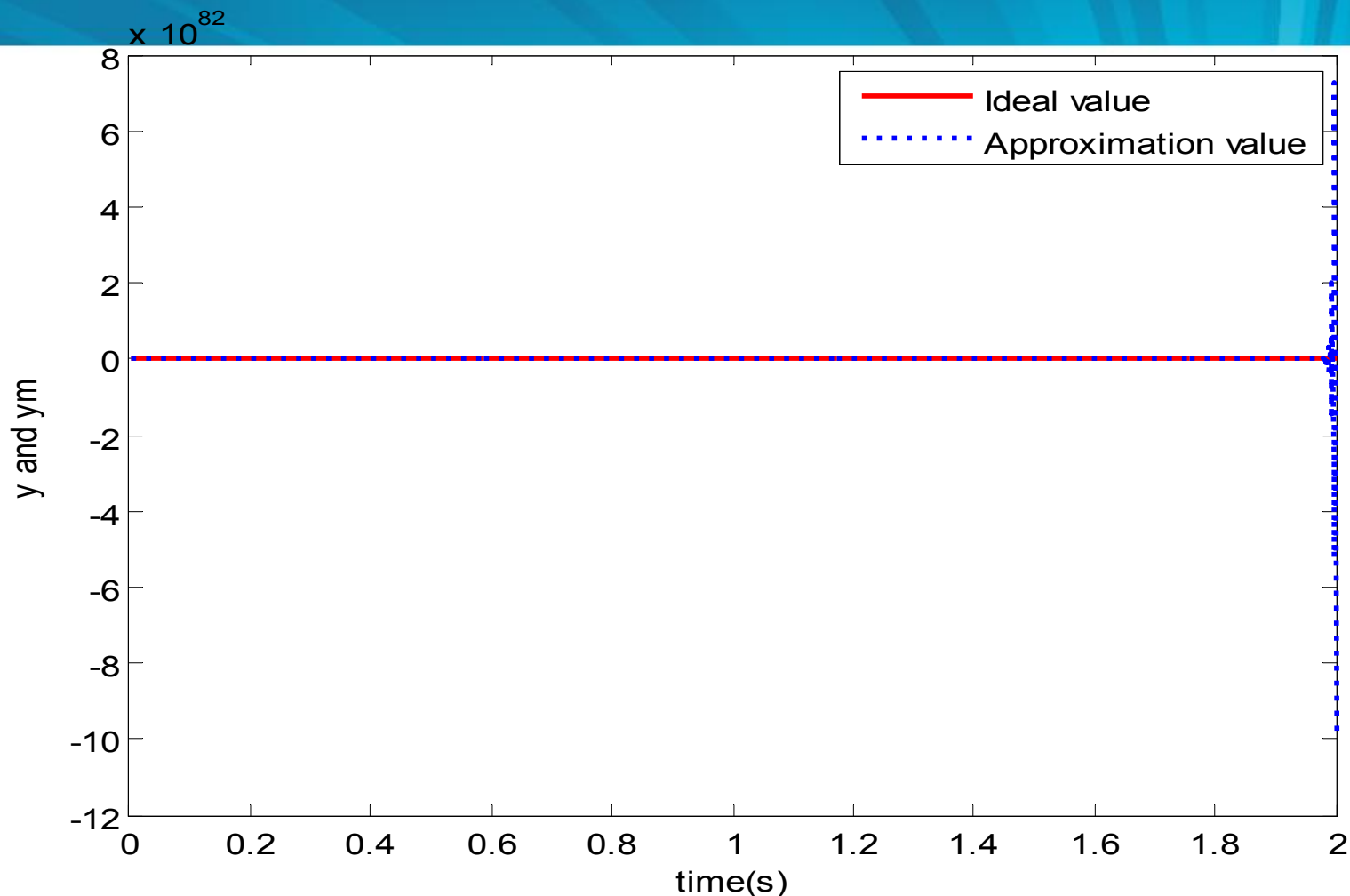


图7-25 合适的 b_j 与不合适的 c_j 值的RBF网络逼近 ($M_b=1$, $M_c=2$)

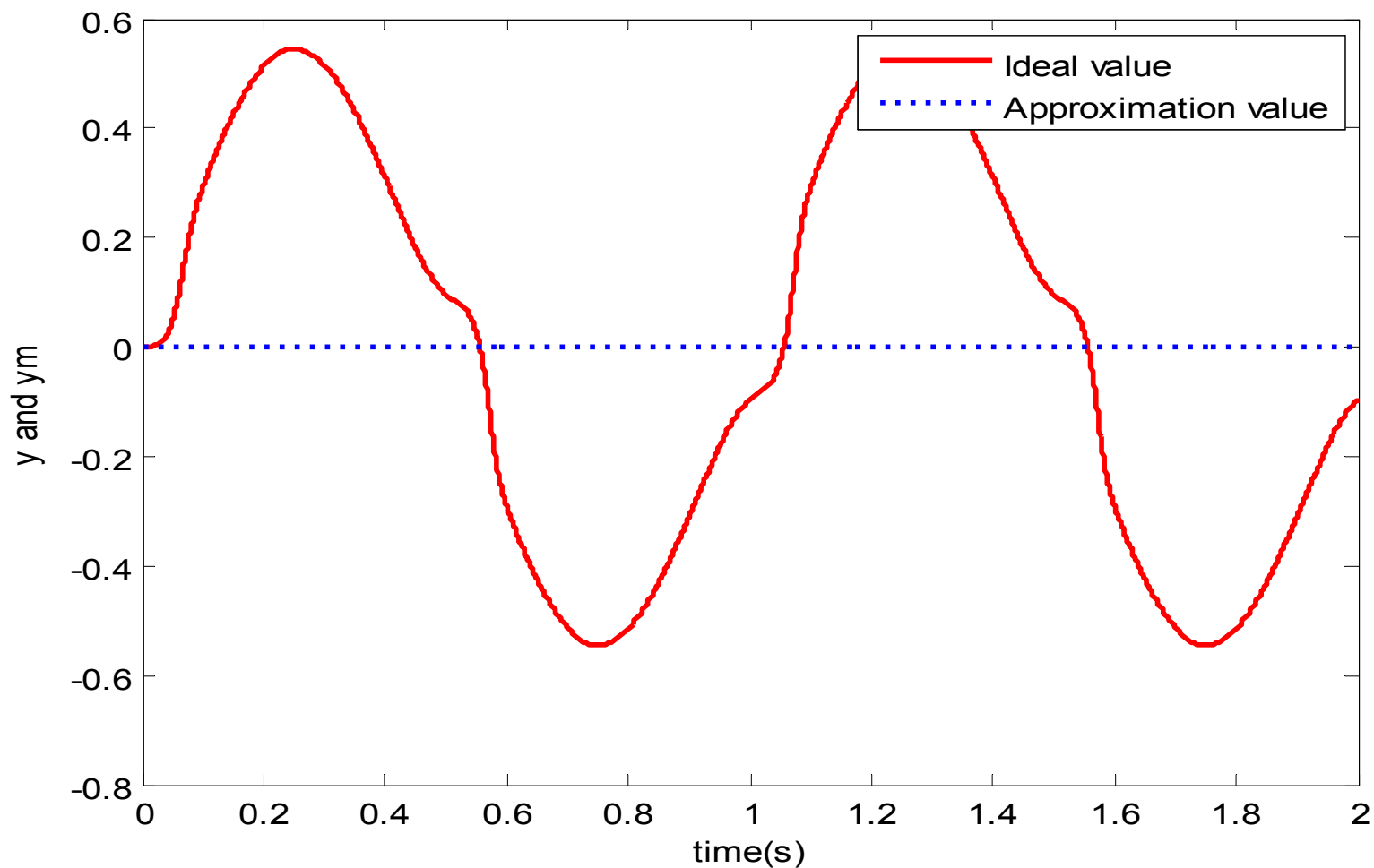


图7-26 不合适 b_j 和 c_j 值的RBF网络逼近($M_b=2$, $M_c=2$)

7.3.5 隐含层节点数对RBF网络逼近的影响

由高斯函数的表达式可见，逼近误差除了与高斯函数的中心点坐标 c_j 和宽度参数 b_j 有关，还与隐含层神经元节点数量有关。

采用RBF网络对如下离散模型进行逼近

$$y(k) = u(k)^3 + \frac{y(k-1)}{1 + y(k-1)^2}$$

仿真中，取 $\alpha = 0.05$ ， $\eta = 0.3$ 。神经网络权值的初始值取零，取高斯基函数参数 $b_j = 1.5$ 。取RBF网络的输入为 $u(k) = \sin t$ 和 $y(k)$ ，网络结构取 $2-m-1$ ， m 为隐含层节点数。为了表明隐含层节点数对网络逼近的影响，分别取 $m=1$ ， $m=3$ ， $m=7$ ，所对应的 c_j 分别取 $c_j=0$ ， $c_j = \frac{1}{3}[-1 \ 0 \ 1]^T$ 和

$$c_j = \frac{1}{9} \begin{bmatrix} -3 & -2 & -1 & 0 & 1 & 2 & 3 \\ -3 & -2 & -1 & 0 & 1 & 2 & 3 \end{bmatrix}^T$$

仿真结果如图7-27至图7-32所示。由仿真结果可见，随着隐含层神经元节点数的增加，逼近误差下降。同时，随着隐含层神经元节点数的增加，为了防止梯度下降法的过度调整造成学习过程发散，应适当降低学习速率 η 。仿真程序为 chap7_9.m。

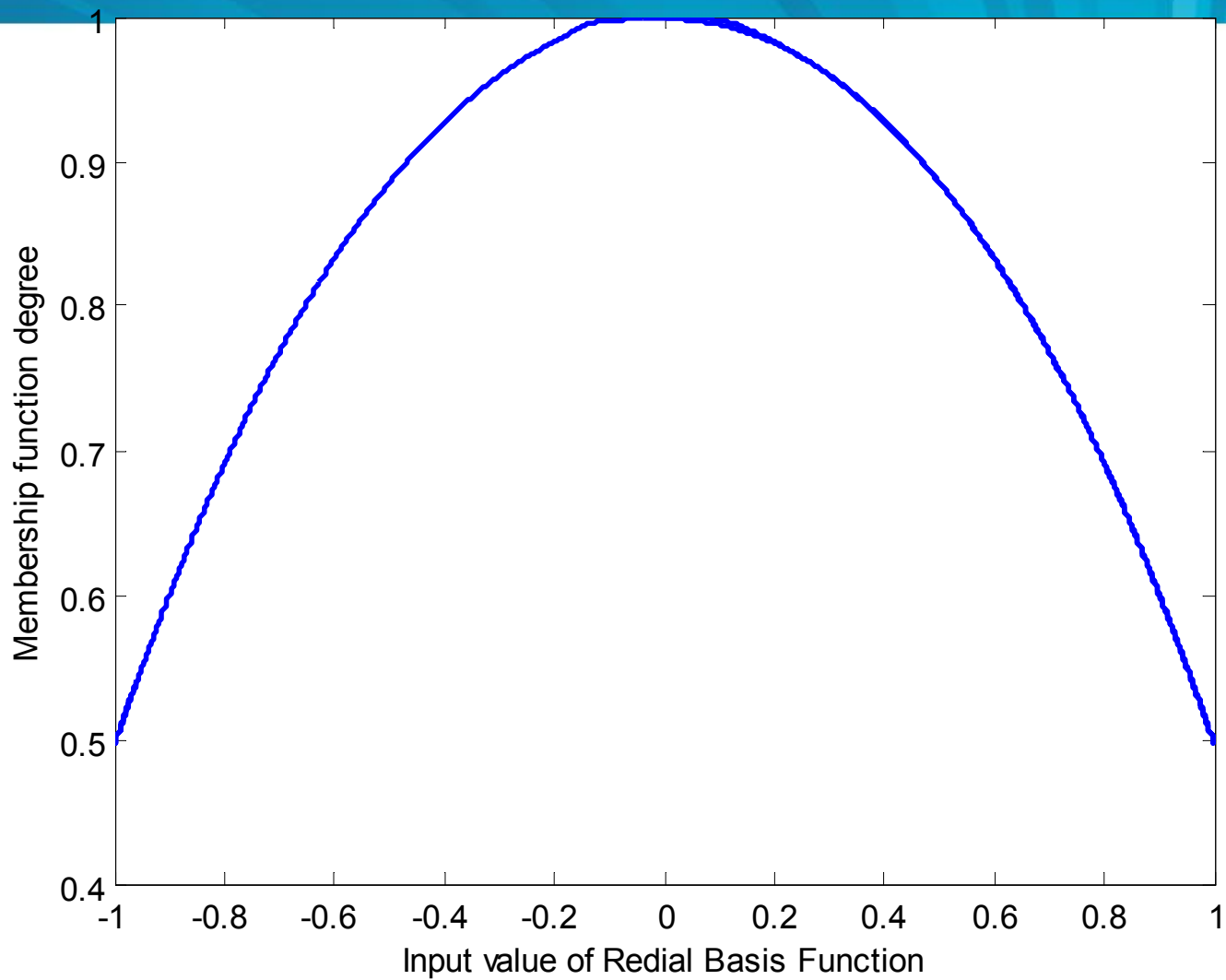


图7-27 单个隐含神经元的高斯基函数($m=1$)

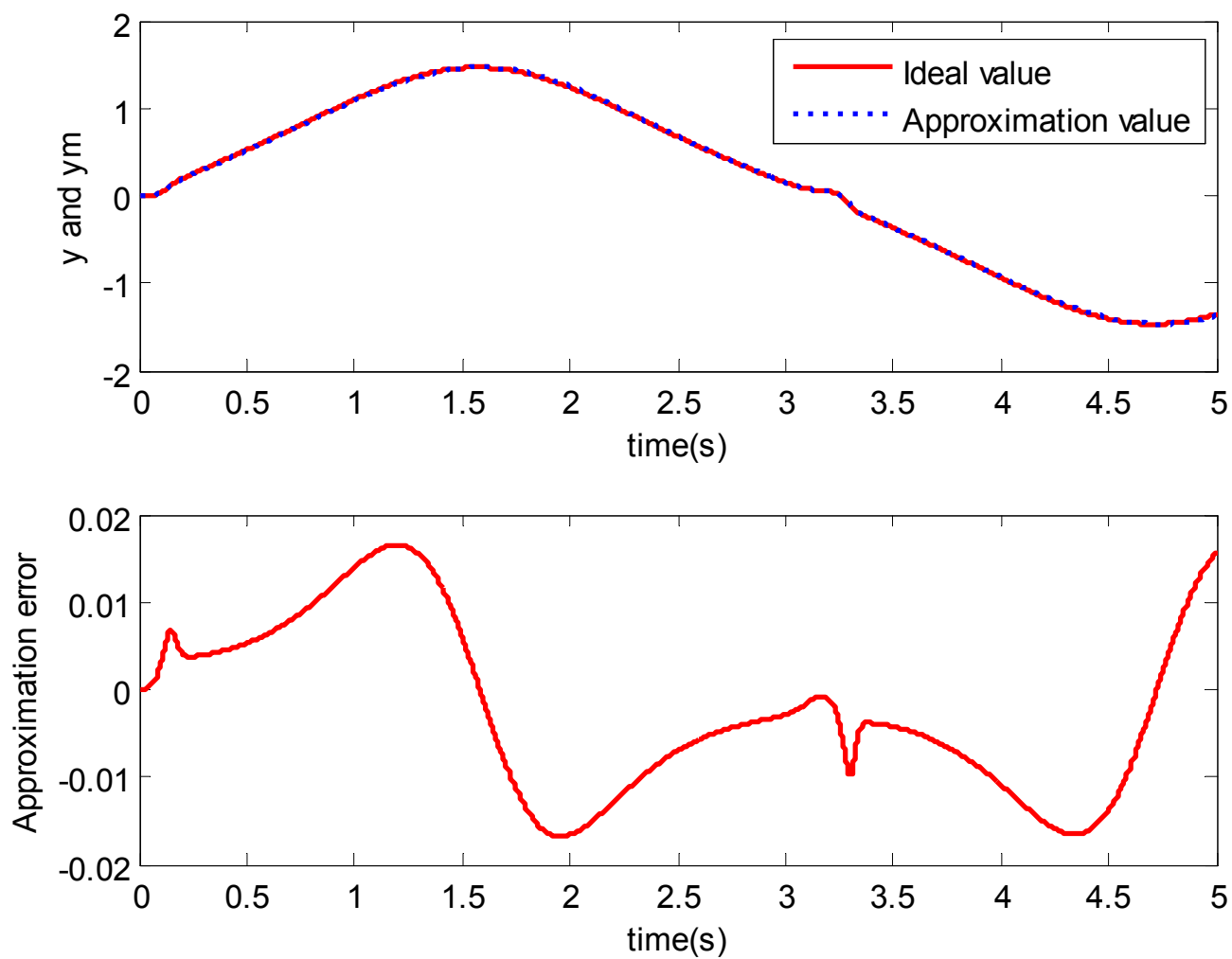


图7-28 含有单个隐含层神经元的逼近 ($m=1$)

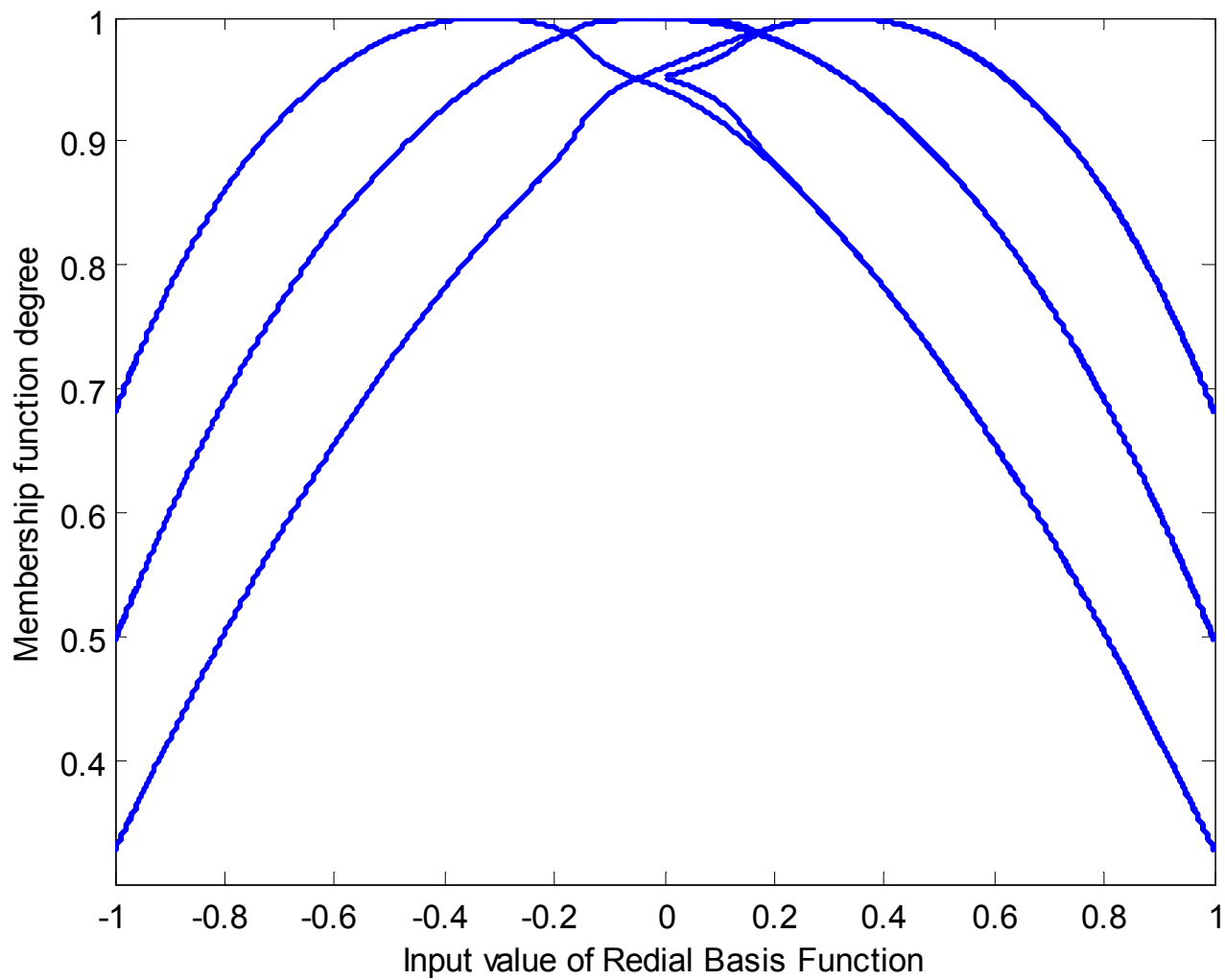


图7-29 3个隐含层神经元的高斯基函数($m=3$)

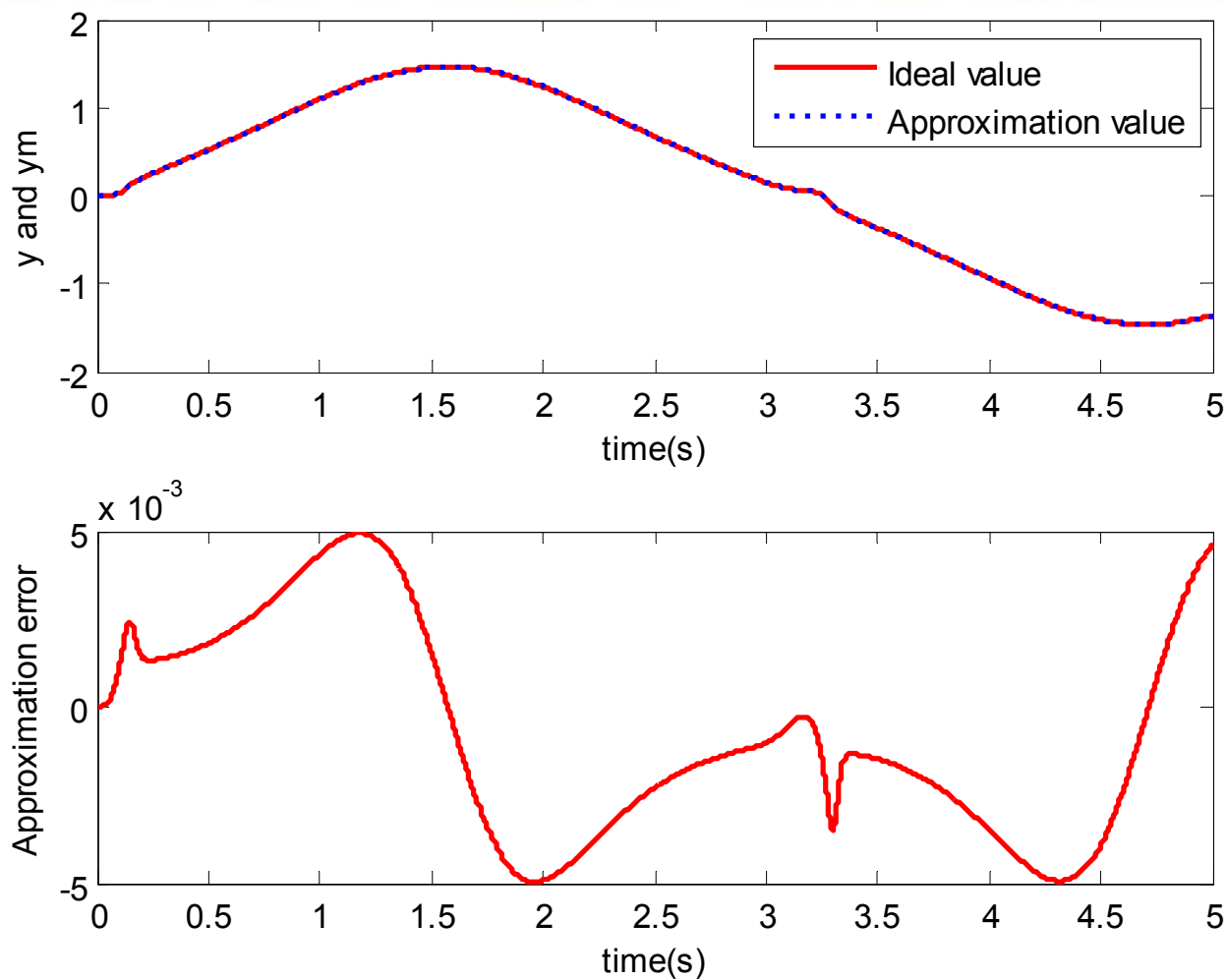


图7-30 含有3个隐含层神经元的逼近 ($m=3$)

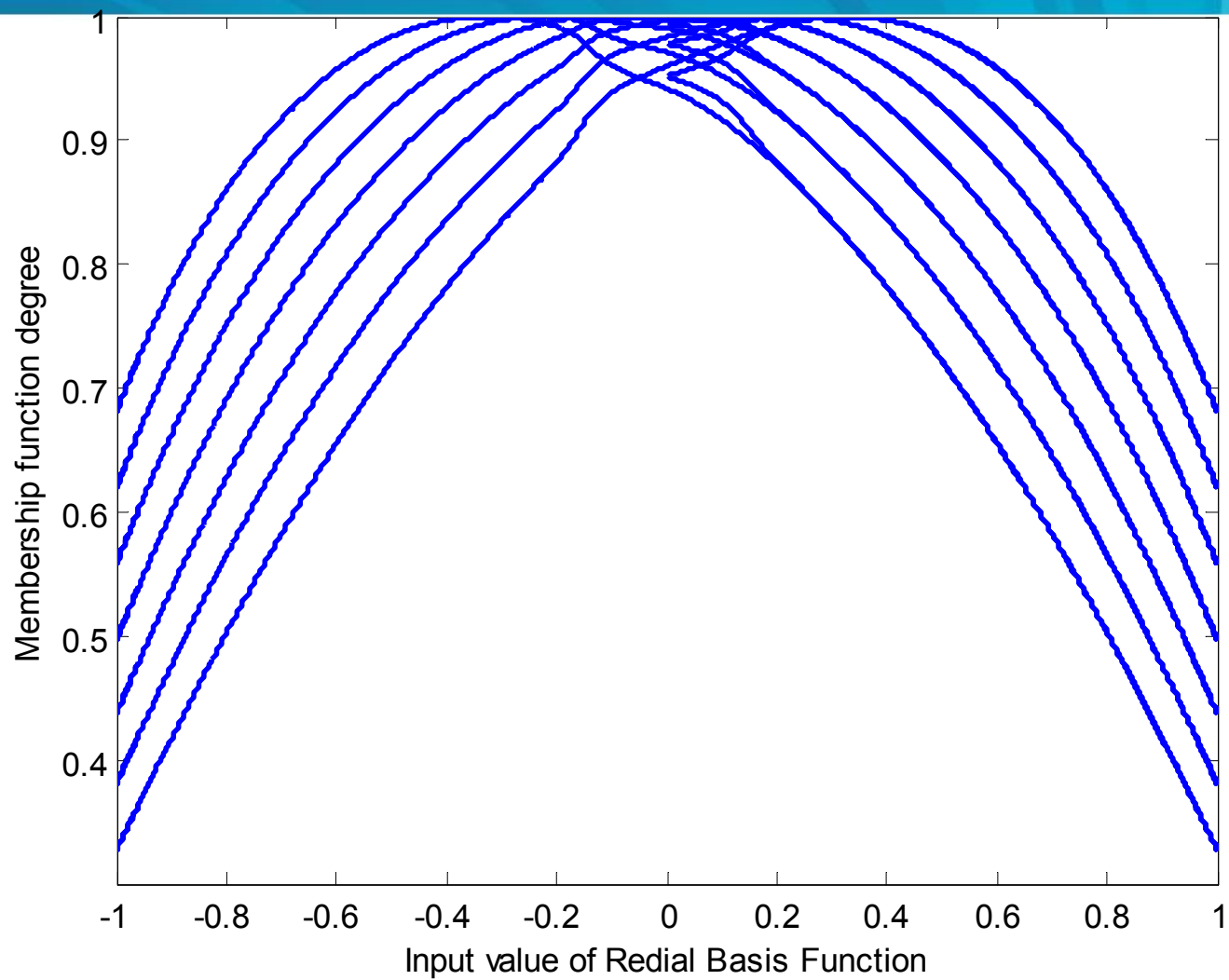


图7-31 7个隐含层神经元的高斯基函数($m=7$)

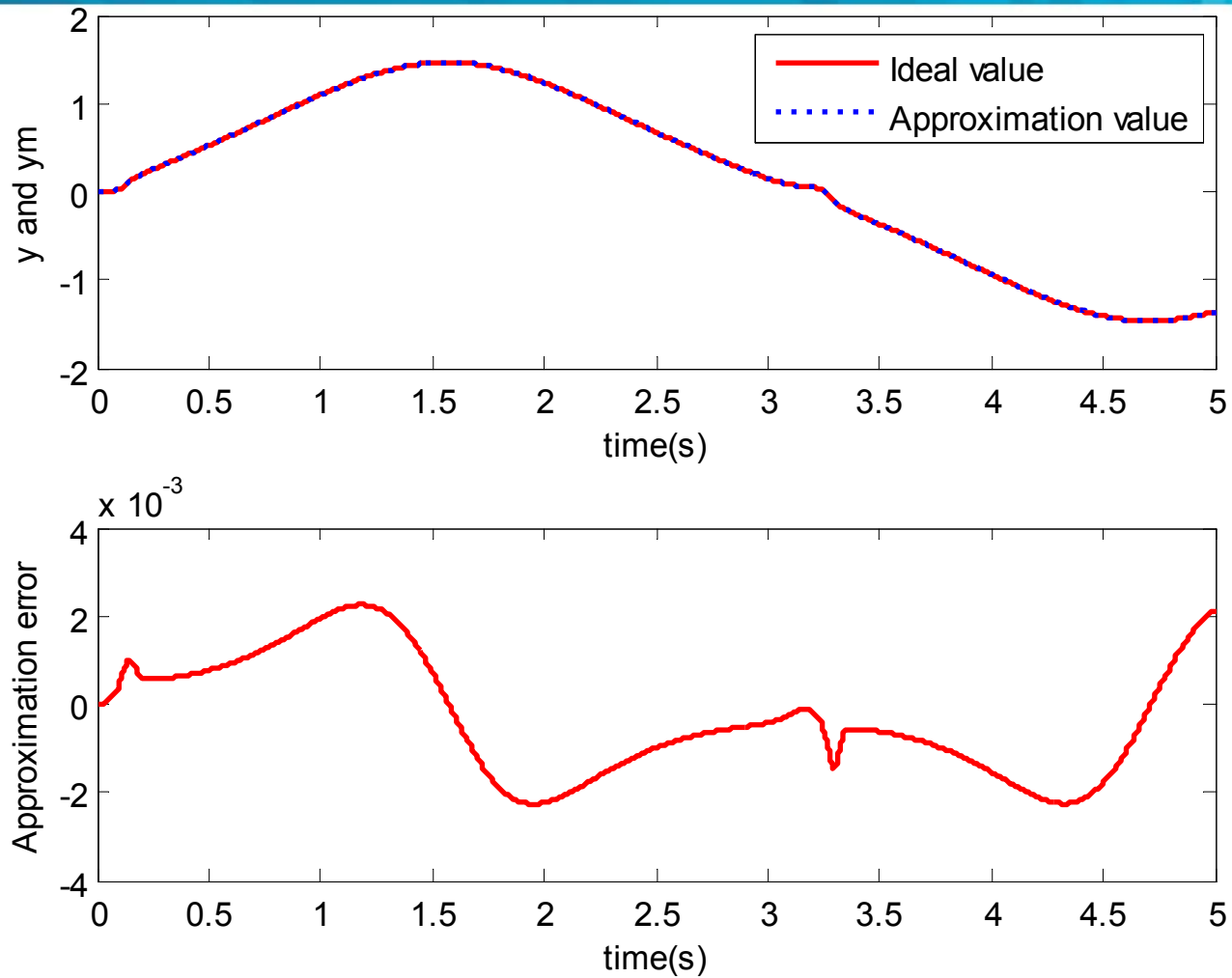


图7-32 含有7个隐含层神经元的逼近 ($m=7$)

7.3.6 控制系统设计中RBF网络的逼近

R B F 网络可对任意未知非线性函数进行任意精度的逼近^[1, 2]。在控制系统设计中，采用RBF网络可实现对未知函数的逼近。

例如，为了估计未知函数 $f(x)$ ，可采用如下RBF网络算法进行逼近

$$h_j = g\left(\|x - c_{ij}\|^2 / b_j^2\right)$$
$$f = W^{*T} h(x) + \varepsilon \quad (7.25)$$

其中 x 为网络输入， i 表示输入层节点， j 为隐含层节点， $h = [h_1, h_2, \dots, h_n]^T$ 为隐含层的输出， W^* 为理想权值， ε 为网络的逼近误差， $\varepsilon \leq \varepsilon_N$

在控制系统设计中，可采样 R B F 网络对未知函数 f 进行逼近。一般可采用系统状态作为网络的输入，网络输出为

$$\hat{f}(x) = \hat{W}^T h(x) \quad (7.26)$$

其中 \hat{W} 为估计权值。

在实际的控制系统设计中，为了保证网络的输入值处于高斯基函数的有效范围，应根据网络的输入值实际范围确定高斯基函数中心点坐标向量 c 值，为了保证高斯基函数的有效映射，需要将高斯基函数的宽度 b_j 取适当的值。 \hat{W} 的调节是通过闭环的 Lyapunov 函数的稳定性分析中进行设计的。