

第8章 高级神经网络

The background is a vibrant blue with a series of white, semi-transparent grid patterns that resemble digital data or neural network layers. These grids are arranged in a way that suggests depth and movement, with some appearing closer and more defined than others. Radiating lines of varying shades of blue emanate from the right side, creating a sense of energy and forward motion. The overall aesthetic is high-tech and futuristic.

8.1 模糊RBF网络

在模糊系统中，模糊集、隶属度函数和模糊规则的设计是建立在经验知识基础上的。这种设计方法存在很大的主观性。将学习机制引到模糊系统中，使模糊系统能够通过不断学习来修改和完善隶属函数和模糊规则，是模糊系统的发展方向。

- 模糊系统与模糊神经网络既有联系又有区别，其联系表现为模糊神经网络在本质上是模糊系统的实现，其区别表现为模糊神经网络又具有神经网络的特性。
- 神经网络与模糊系统的比较见表8-1。模糊神经网络充分地利用了神经网络和模糊系统各自的优点，因而受到了重视。

	模糊系统	神经网络
获取知识	专家经验	算法实例
推理机制	启发式搜索	并行计算
推理速度	低	高
容错性	低	非常高
学习机制	归纳	调整权值
自然语言实现	明确的	不明显
自然语言灵活性	高	低

- 将神经网络的学习能力引到模糊系统中，将模糊系统的模糊化处理、模糊推理、精确化计算通过分布式的神经网络来表示是实现模糊系统自组织、自学习的重要途径。在模糊神经网络中，神经网络的输入、输出节点用来表示模糊系统的输入、输出信号，神经网络的隐含节点用来表示隶属函数和模糊规则，利用神经网络的并行处理能力使得模糊系统的推理能力大大提高。

- 模糊神经网络在本质上是将常规的神经网络赋予模糊输入信号和模糊权值，其学习算法通常是神经网络学习算法或其推广。模糊神经网络技术已经获得了广泛的应用，当前的应用主要集中在以下几个领域：模糊回归、模糊控制、模糊专家系统、模糊矩阵方程、模糊建模和模糊模式识别。
- 模糊神经网络是将模糊系统和神经网络相结合而构成的网络。利用RBF网络与模糊系统相结合，构成了模糊RBF网络。

8.1.1 网络结构

采用图8-1所示的模糊神经网络系统，其模糊推理系统主要由输入层、模糊化层、模糊相联层、模糊后相连层和输出层构成。

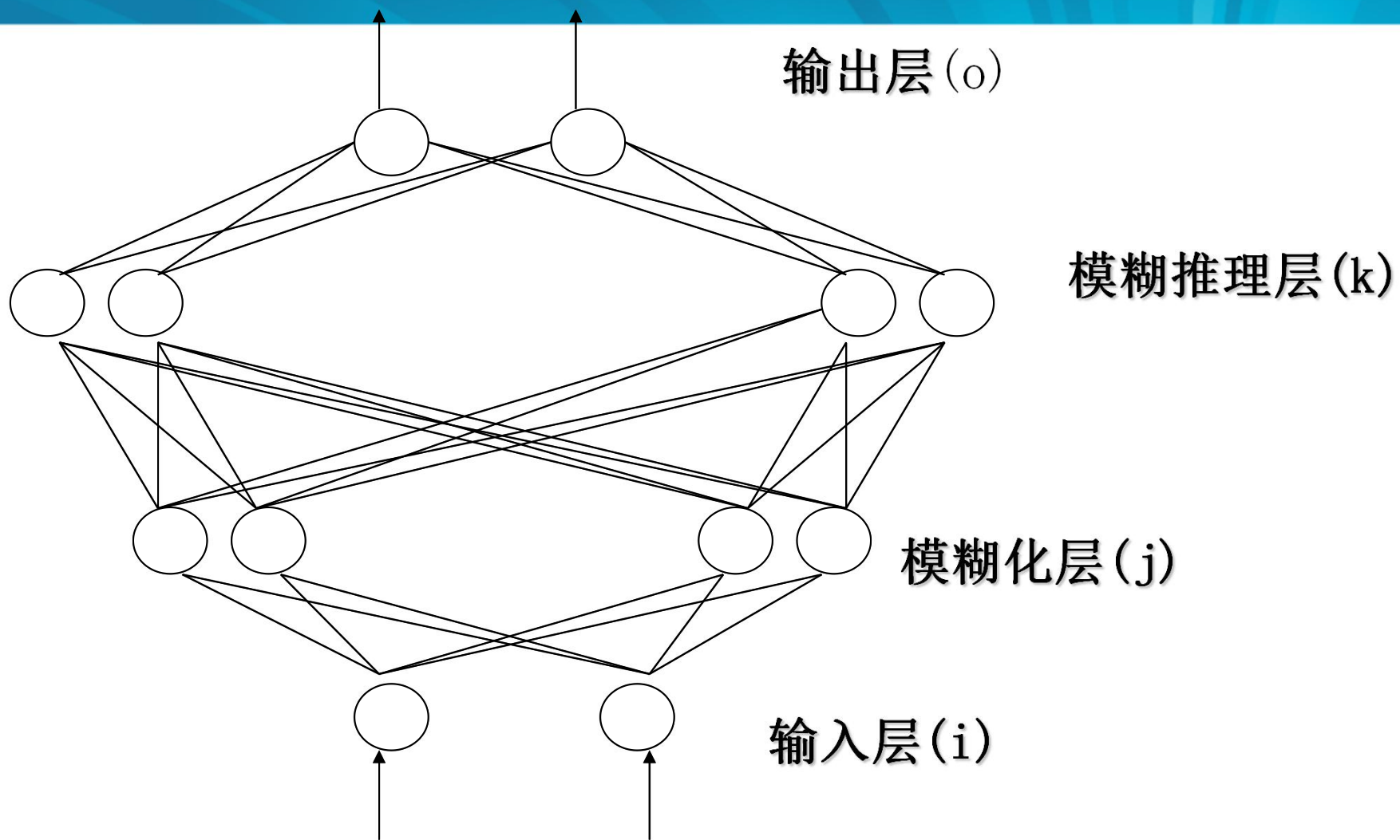


图8-1 模糊RBF神经网络结构

模糊RBF网络中信号传播及各层的功能表示如下

:

第一层：输入层

该层的各个节点直接与输入量的各个分量连接，将输入量传到下一层。对该层的每个节点*i*的输入输出表示为：

$$f_1(i) = x_i$$

第二层：隶属函数层，即模糊化层

该层的每个节点具有隶属函数的功能，采用高斯函数作为隶属函数。对第j个节点：

$$f_2(i, j) = \exp(\text{net}_j^2)$$
$$\text{net}_j^2 = -\frac{(f_1(i) - c_{ij})^2}{(b_j)^2}$$

其中 c_{ij} 和 b_j 分别是第i个输入变量的第j个模糊集合高斯函数的均值和标准差。

第三层：规则层，即模糊推理层

该层通过与模糊化层的连接来完成模糊规则的匹配，各个节点之间实现模糊运算，即通过各个模糊节点的组合得到相应的点火强度。每个节点j的输出为该节点所有输入信号的乘积，即

$$f_3(j) = \prod_{j=1}^N f_2(i,j) \qquad N = \prod_{i=1}^n N_i$$

第四层：输出层

该层的每个节点的输出为该节点所有输入信号的加权和，即

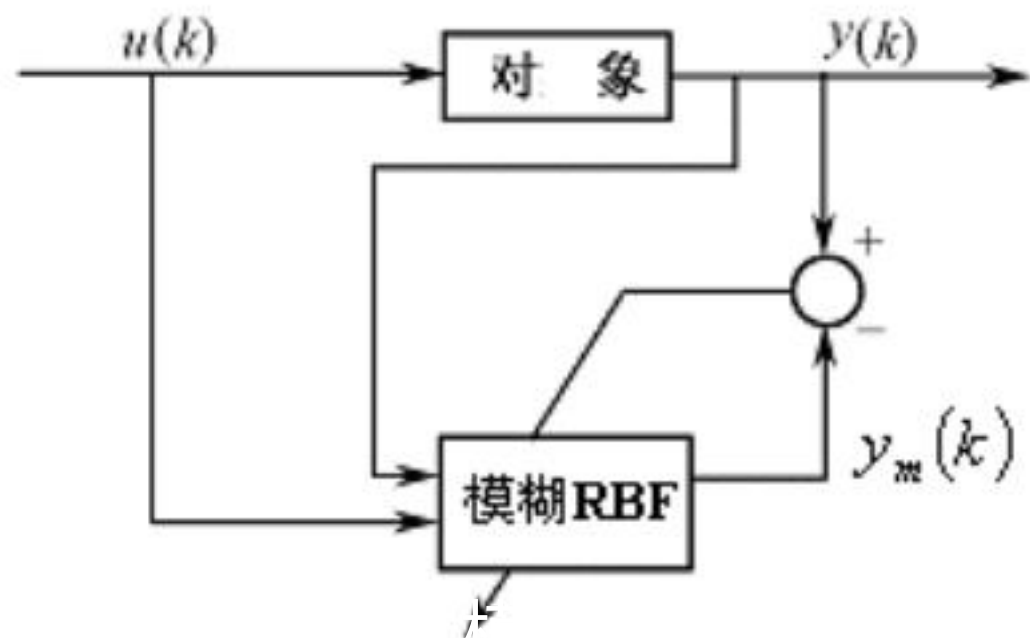
$$f_4(l) = W \cdot f_3 = \sum_{j=1}^N \omega(l, j) \cdot f_3(j)$$

其中 l 为输出层节点的个数， W 为输出节点与第三层各节点的连接权矩阵。

在此模糊神经网络中，可调参数有三类：一类为规则的权系数 c_{ij} 第二类和第三类为高斯函数的均值和标准差 b_j ，即输入隶属函数的参数。

8.1.2 基于模糊RBF网络的逼近算法

采用模糊RBF网络逼近对象，取网络结构为2-4-1，如图8-2所示。



- 取 $y_m(k) = f_4$, $y_m(k)$ 和 $y(k)$ 分别表示网络输出和理想输出。网络的输入 x_1 和 x_2 为 $u(k)$ 和 $y(k)$, 网络的输出为 $y_m(k)$, 则网络逼近误差为:

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

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

采用梯度下降法来修正可调参数, 定义目标函数为:

网络的学习算法如下：

输出层的权值通过如下方式来调整：

$$\Delta w(k) = -\eta \frac{\partial E}{\partial w} = -\eta \frac{\partial E}{\partial e} \frac{\partial e}{\partial y_m} \frac{\partial y_m}{\partial w} = \eta e(k) f_3$$

则输出层的权值学习算法为：

$$w(k) = w(k-1) + \Delta w(k) + \alpha(w(k-1) - w(k-2))$$

其中 η 为学习速率， α 为动量因子。

隶属函数参数通过如下方式调整

$$\Delta c_{ij} = -\eta \frac{\partial E}{\partial c_{ij}} = -\eta \frac{\partial E}{\partial net_j^2} \frac{\partial net_j^2}{\partial c_{ij}} = -\eta \delta_j^2 \frac{2(x_i - c_{ij})}{b_{ij}^2}$$

$$\Delta b_j = -\eta \frac{\partial E}{\partial b_j} = -\eta \frac{\partial E}{\partial net_j^2} \frac{\partial net_j^2}{\partial b_j} = \eta \delta_j^2 \frac{2(x_i - c_{ij})}{b_j^3}$$

其中

$$\delta_j^2 = \frac{\partial E}{\partial net_j^2} = -e(k) \frac{\partial y_m}{\partial net_j^2} = -e(k) \frac{\partial y_m}{\partial f_3} \frac{\partial f_3}{\partial f_2} \frac{\partial f_2}{\partial net_j^2} = -e(k) w f_3$$

隶属函数参数学习算法为：

$$c_{ij}(k) = c_{ij}(k-1) + \Delta c_{ij}(k) + \alpha(c_{ij}(k-1) - c_{ij}(k-2))$$

$$b_j(k) = b_j(k-1) + \Delta b_j(k) + \alpha(b_j(k-1) - b_j(k-2))$$

8.1.3 仿真实例

使用模糊RBF网络逼近对象：

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

其中采样时间为1ms。

模糊RBF网络逼近程序见chap8_1.m。