第 4 章　径向基函数神经网络

除了前面介绍的多层感知器外，径向基函数神经网络（Radial Basis Function Neural Network，简称 RBF 网）是另一类常用的三层前馈网络，也可用于函数逼近及分类。与 BP 网相比，RBF 网不仅有生理学基础，而且结构更简洁，学习速度也更快。本章 4.1 节先介绍 RBF 网的结构及工作原理；然后在 4.2 节和 4.3 节介绍该网络的生理学基础和数学基础；在 4.4 节，我们介绍 RBF 网的学习算法，包括基于聚类的学习算法，梯度学习算法，及正交最小二乘学习算法，包括它们的原理及实现；在 4.5 节，我们将介绍 RBF 网的学习动态，最后介绍了 RBF 网用于非线性函数逼近的一个例子。

4.1　RBF 网络结构和工作原理

径向基函数神经网络(RBF 网)是一种前馈神经网络，一般为三层结构，其如图 4.1 所示。

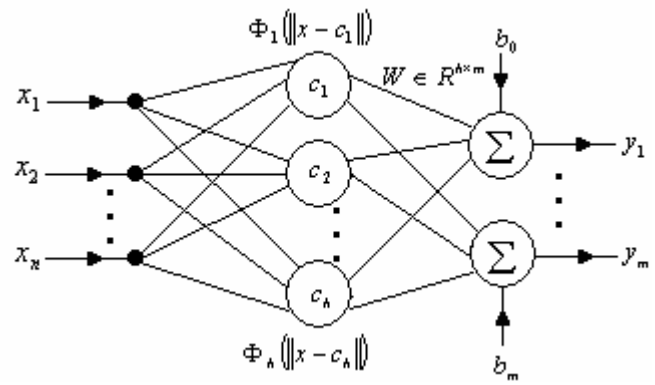

图 4.1　RBF 网结构

图 4.1 为 $n-h-m$ 结构的 RBF 网，即网络具有 n 个输入，h 个隐节点，m 个输出。其中 $\boldsymbol{x}=(x_1,\ x_2,\cdots,\ x_n)^T \in R^n$ 为网络输入矢量，$\boldsymbol{W} \in R^{h\times m}$ 为输出权矩阵，$b_0,\cdots,b_m$ 为输出单元偏移，$\boldsymbol{y}=[y_1,...,y_m]^T$ 为网络输出，$\Phi_i(*)$ 为第 i 个隐节点的激活函数。图中输出层节点中的 $\sum$ 表示输出层神经元采用线性激活函数，当然输出神经元也可以采用其它非线性激活函数，如 Sigmoid 函数。

我们知道，多层感知器（包括 BP 网）的隐节点基函数采用线性函数，激活函数则采用 Sigmoid 函数或硬极限函数。与多层感知器不同，RBF 网的最显著的特点是隐节点的

基函数采用距离函数（如欧氏距离），并使用径向基函数（如 Gaussian 函数）作为激活函数。径向基函数关于 n 维空间的一个中心点具有径向对称性，而且神经元的输入离该中心点越远，神经元的激活程度就越低。隐节点的这一特性常被称为“局部特性”。

因此 RBF 网的每个隐节点都具有一个数据中心，如图 4.1 中 c_i 就是网络中第 i 个隐节点的数据中心值，$\|*\|$ 则表示欧氏范数。

径向基函数 $\Phi_i(\cdot)$ 可以取多种形式，如下面的式（4.1）至式（4.3），其曲线形状如图 4.2 所示。

(1) Gaussian 函数

$$\Phi_i(t)=e^{-\frac{t^2}{\delta_i^2}} \tag{4.1}$$

(2) Reflected sigmoid 函数

$$\Phi_i(t)=\frac{1}{1+e^{\frac{t^2}{\delta^2}}} \tag{4.2}$$

(3) 逆 Multiquadric 函数

$$\Phi_i(t)=\frac{1}{\left(t^2+\delta_i^2\right)^{\alpha}},\quad \alpha>0 \tag{4.3}$$

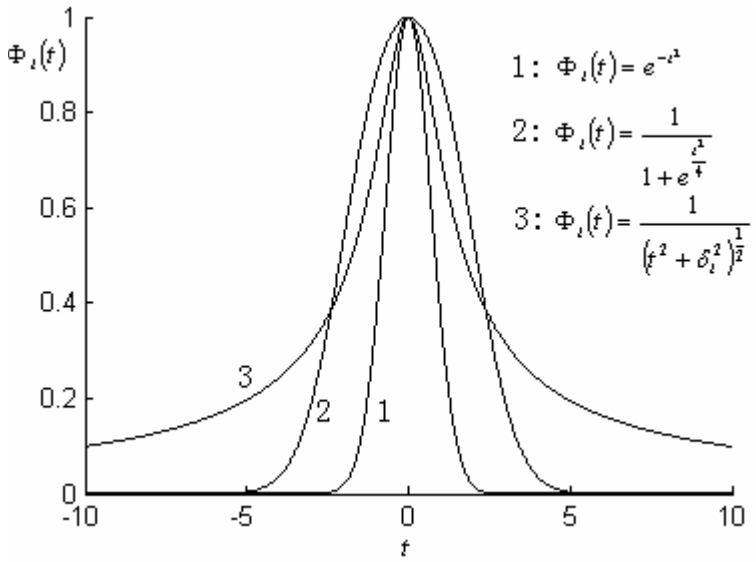

图 4.2　RBF 函数

式（4.1）至式（4.3）中的 δ_i 称为该基函数的扩展常数（Spread）或宽度。显然，δ_i 越小，径向基函数的宽度就越小，基函数就越具有选择性。

于是图 4.1 中，RBF 网的第 k 个输出可表示为：

$$y_k = \sum_{i=1}^{h} w_i \Phi_i(\|\boldsymbol{x} - \boldsymbol{c}_i\|) \tag{4.4}$$

下面以两输入单输出函数逼近为例，简要介绍 RBF 网的工作机理。与输出节点相连的隐层第 i 个隐节点的所有参数可用三元组 $(\boldsymbol{c}_i, \delta_i, w_i)$ 表示。由于每个隐层神经元都对输入产生响应，且响应特性呈径向对称（即是一个个同心圆），于是 RBF 网的作用机理可用图 4.3 表示。图 4.3 表示输入区域中中有 6 个神经元，每个神经元都对输入 $\boldsymbol{x}$ 产生一个响应 $\Phi_i(\|\boldsymbol{x} - \boldsymbol{c}_i\|)$，而神经网络的输出则是所有这些响应的加权和。

由于每个神经元具有局部特性，最终整个 RBF 网最终也呈现“局部映射”特性，即 RBF 网是一种局部相应神经网络。这意味着如果神经网络有较大的输出，必定激活了一个或多个隐节点。

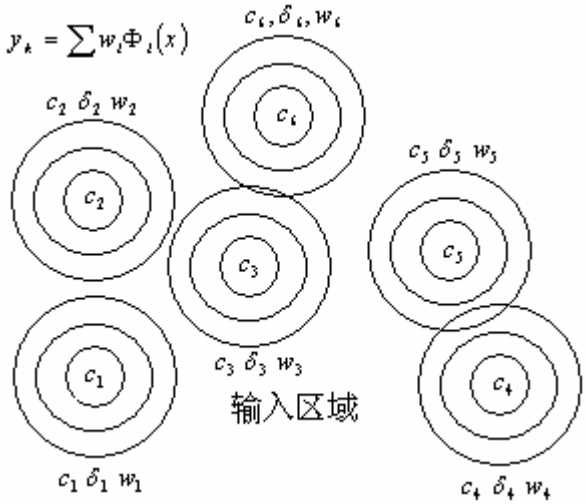

图 4.3　RBF 网的工作原理

4.2　RB F 网的生理学基础

事实上，RBF网的隐节点的局部特性主要是模仿了某些生物神经元的“内兴奋外抑制”（on-center off-surround）功能，灵长类动物的的视觉系统中就有这样的神经元。下面简要介绍人眼接收信息的过程并介绍近兴奋－远抑制功能。

眼是人接收来自外部信息的最主要的接收器官。外界物体的光线射入眼中，聚焦后在视网膜上成像，视网膜发出神经冲动达到大脑皮层视区，产生视觉。在所有的感官系统中，视网膜的结构最复杂。视网膜为感光系统，能感受光的刺激，发放神经冲动。它不仅有一级神经元(感光细胞),还有二级神经元(双极细胞)和三级神经无(神经节细胞)。

感光细胞与双极细胞形成突触联系，双极细胞外端与感光细胞相连，内端与神经节细胞相接，神经节细胞的轴突则组成视神经束。来自两侧的视神经在脑下垂体前方会合成视交叉。在这里组成每一根视神经的神经纤维束在进一步进入脑部之前被重新分组。从视神经交叉再发出的神经束叫作视束。在重新分组时，来自两眼视网膜右侧的纤维合成一束传向脑的右半部．来自两眼视网膜左侧的纤维合成另一束传向脑的左半部。这两束经过改组的纤维视束继续向脑内行进，大部分终止于丘脑的两个被分成外侧膝状体的神经核。

外膝体完成输入信息处理上的第一次分离,然后传送到大脑的第一视区和第二视区。外膝体属丘脑，是眼到视皮层的中继站，这就是视觉通路。视网膜上的感光细胞通过光化学反应和光生物化学反应，产生的光感受器电位和神经脉冲就是沿着视觉通路进行传播的。

从神经感受野可以作出员完善的分析。中枢神经元的感受野是指能影响某一视神经元反应的视网膜或视野的区域。可通过电生理学试验记录感受野的形状。电生理学试验可理解为有一根极小的电极置于某个细胞内的特定欲记录电位的区域，这样，当光束照到视网膜上，如果该细胞被激活，通过这一区域的电脉冲（Spike）就增加；反之，如果该细胞被抑制，通过这一区域的电脉冲（Spike）就减少。静止的细胞膜电位约 70 毫伏，因此我们可以用示波器观察到这些脉冲。

每个视皮层，外侧膝状体的神经元或视网膜神经细胞节细胞在视网膜上均有其特定的感受野。Hubel 与 Wiesel 比较了侧膝核与节细胞的感受野，结果发现它们非常相识，都呈圆形，并具有近兴奋远抑制(on-center off-surround)或远兴奋近抑制(off-center on surround)功能，如图 4.4 所示。

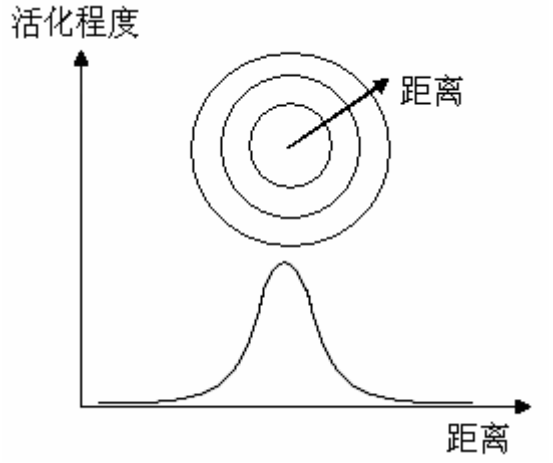

图 4.4　神经元的近兴奋远抑制现象

对于每一个这样的内兴奋外抑制神经元，我们可以用以下函数进行建模：

$$\Phi_i(\boldsymbol{x}) = G\left(\left\|\boldsymbol{x} - \boldsymbol{c}_i\right\|\right) \tag{4.5}$$

其中 $\boldsymbol{x}$ 为输入（光束照在视网膜上的位置），$\boldsymbol{c}_i$ 为感受野的中心，对应于视网膜上使

神经元最兴奋的光照位置。$G(\cdot)$的具体形式见前面的式（4.1）至式（4.3）。

4.3 RBF 网的数学基础

4.3.1 内插问题

假定共有 N 个学习样本，其输入为 $s=(\boldsymbol{X}_1,\ \boldsymbol{X}_2,\cdots,\ \boldsymbol{X}_N)$，相应的样本输出，即教师信号（单输出）为 $\boldsymbol{t}=(y_1,\ y_2,\cdots,\ y_N)$。所谓的多变量内插问题是指寻找函数 F，满足以下的内插条件：

$$y_i = F(\boldsymbol{X}_i) \tag{4.6}$$

这是一个非常经典的数学问题，可以有多种解决方案，采用 RBF 网也可以解决这一问题的学习问题。由式（4.4）可知，使用 RBF 网前必须确定其隐节点的数据中心（包括数据中心的数目、值、扩展常数）及相应的一组权值。RBF 网解决内插问题时，使用 N 个隐节点，并把所有的样本输入选为 RBF 网的数据中心，且各基函数取相同的扩展常数。于是 RBF 网从输入层到隐层的输出便是确定的。

现在我们再来确定网络的 N 个输出权值 $\boldsymbol{W}=(w_1,\ w_2,\cdots,\ w_N)^T$（待定）。我们只要把所有的样本再输入一遍，便可解出各 w_i 的值。假定当输入为 $\boldsymbol{X}_i$，$i=1,2,\cdots,\ N$ 时，第 j 个隐节点的输出为：

$$h_{ij} = \Phi_j(\left\|\boldsymbol{X}_i - \boldsymbol{c}_j\right\|) \tag{4.7}$$

其中 $\Phi_j(\cdot)$ 为该隐节点的激活函数，$\boldsymbol{c}_j = \boldsymbol{X}_j$ 为该隐节点 RBF 函数的数据中心。于是可定义 RBF 网的隐层输出阵为 $\boldsymbol{H}=\left[h_{ij}\right]$，$\boldsymbol{H}\in R^{N\times N}$。此时 RBF 网的输出为：

$$f(\boldsymbol{X}_i)=\sum_{j=1}^{N} h_{ij} w_j = \sum_{j=1}^{N} w_j \Phi(\left\|\boldsymbol{X}_i - \boldsymbol{c}_j\right\|) \tag{4.8}$$

令 $y_i = f(\boldsymbol{X}_i)$，$\boldsymbol{y}=\boldsymbol{t}^T=(y_1,\ y_2,\cdots,\ y_N)^T$，$\boldsymbol{W}=(w_1,\ w_2,\cdots,\ w_N)^T$，便得：

$$\boldsymbol{y}=\boldsymbol{H}\boldsymbol{W} \tag{4.8}$$

如果 $\boldsymbol{H}\in R^{N\times N}$ 可逆，即其列向量构成 R^N 中的一组基，则输出权矢量可由下式得到

$$\boldsymbol{W} = \boldsymbol{H}^{-1}\boldsymbol{y} \tag{4.9}$$

事实上，根据 Micchelli 定理[Micc1986]，如果隐节点激活函数采用上述的径向基函数，且 $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_N$ 各不相同，则隐层输出阵 $\boldsymbol{H}$ 的可逆性是可以保证的。

因此，如果把全部样本输入作为 RBF 网的数据中心，网络在样本输入点的输出就等于教师信号，此时网络对样本实现了完全内插，即对所有样本误差为 0。但上式方案存在以下问题：

(1) 通常情况下，样本数据较多，即 N 数值较大，上述方案中隐层输出阵 $\boldsymbol{H}$ 的条件数可能过大，求逆时可能导致不稳定。

(2) 如果样本输出含有噪声，此时由于存在过学习的问题，作完全内插是不合适的，而对数据作逼近可能更合理。

为了解决这些问题，可以采用下面的正则化网络。

4.3.2 正则化网络

假定 $S = \left\{ \left(\boldsymbol{X}_i, y_i \right) \in R^n \times R \middle| \ i = 1,2,..., N \right\}$ 为我们欲用函数 F 逼近的一组数据。传统的寻找逼近函数 F 的方法是通最小化过以下目标函数（标准误差项）实现的：

$$E_S(F) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - F(\boldsymbol{X}_i)\right)^2 \tag{4.10}$$

该函数体现了期望响应与实际响应之间的距离。而所谓的正则化方法[Tikh1977]，是指在标准误差项基础上增加了一个限制逼近函数复杂性的项（正则化项），该正则化项体现逼近函数的“几何”特性：

$$E_R(F) = \frac{1}{2}\|DF\|^2 \tag{4.11}$$

其中 D 为线性微分算子。于是正则化方法的总的误差项定义为：

$$E(F) = E_S(F) + \lambda E_R(F) \tag{4.12}$$

其中 λ 为正则化系数（正实数）。我们将在第 7 章介绍正则化理论，这里给出上述正则化问题的解为[Pogg1990]

$$F(X) = \sum_{i=1}^{N} w_i G(\boldsymbol{X}, \boldsymbol{X}_i) \tag{4.13}$$

其中 $G(\boldsymbol{X}, \boldsymbol{X}_i)$ 为自伴随算子 $\tilde{D}D$ 的 Green 函数，w_i 为权系数。Green 函数 $G(\boldsymbol{X}, \boldsymbol{X}_i)$ 的形式依赖于算子 D 的形式，如果 D 具有平移不变性和旋转不变性，则 Green 函数的值取决于 $\boldsymbol{X}$ 与 $\boldsymbol{X}_i$ 之间的距离，即

$$G(\boldsymbol{X},\boldsymbol{X}_i)=G\left(\left\|\boldsymbol{X}-\boldsymbol{X}_i\right\|\right) \tag{4.14}$$

如果选择不同的算子 D（应具有平移和旋转不变性），便可得到不同的 Green 函数，包括 Gaussian 函数这样的最常用的径向基函数。

按照上述方式得到的网络称为正则化网络，其拓扑结构如下图 4.5 所示。

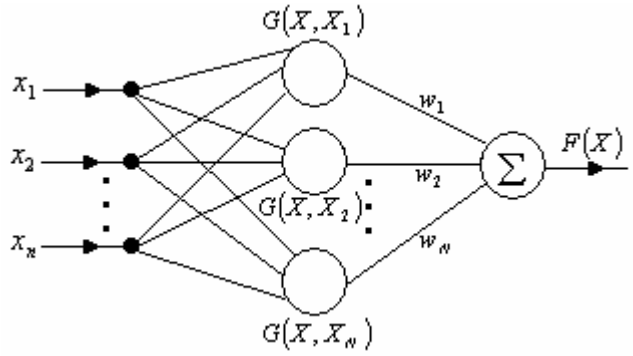

图 4.5　正则化网络

该正则化网络具有以下特点：

(1) 具有万能逼近能力，即只要有足够的隐节点，正则化网络能逼近紧集上的任意连续函数；

(2) 具有最佳逼近特性，即任给未知的非线性函数 f，总可以找到一组权系数，在该组系数下正则化网络对 f 的逼近优于其它系数；

(3) 正则化网络得到的解是最优的，即通过最小化式（4.12），得到了同时满足对样本的逼近误差和逼近曲线平滑性的最优解。

4.4　RBF 网的常用学习算法

正则化网络要求网络的隐节点数等于训练样本数，在训练样本较多时该方法显然不能令人满意，因此有必要寻找更适合工程化应用的 RBF 网学习方法。由式（4.4）可见，给定了训练样本，RBF 网的学习算法应该解决以下问题：结构设计，即如何确定网络隐节点数 h；确定各径向基函数的数据中心 $\boldsymbol{c}_i$ 及扩展常数 δ_i；输出权值修正。一般情况下，如果知道了网络的隐节点数、数据中心和扩展常数，RBF 网从输入到输出就成了一个线性方程组，此时权值学习可采用最小二乘法求解。

根据数据中心的取值方法，RBF 网的设计方法可分为两大类。

1) 数据中心从样本输入中选取

这种方法中数据中心从样本输入中选，如 OLS 算法、ROLS 算法、进化优选算法等。这类算法的特点是数据中心一旦获得便不再改变，而隐节点的数目或者一开始就固定，

或者在学习过程中动态调整。

2） 数据中心动态调节方法

这类方法中数据中心的位置在学习过程中是动态调节的,如各种基于动态聚类(最常用的是 K-means 聚类或 Kohenen 提出的 SOFM 方法[Koho1984])方法、梯度训练方法、资源分配网络(RAN)等。

这些方法各有优缺点。第一类算法较容易实现，且能在权值学习的同时确定隐节点的数目，并保证学习误差不大于给定值，但数据中心从样本输入中选取是否合理，值得进一步讨论。另外，OLS 算法并不一定能设计出具有最小结构的 RBF 网，也无法确定基函数的扩展常数。第二类方法中，聚类方法的优点是能根据各聚类中心之间的距离确定各隐节点的扩展常数，缺点是确定数据中心时只用到了样本输入信息，而没有用到样本输出信息；另外聚类方法也无法确定聚类的数目(RBF 网的隐节点数)。由于 RBF 网的隐节点数对其泛化能力有极大的影响，所以寻找能确定聚类数目的合理方法，是聚类方法设计 RBF 网时需首先解决的问题。

下面我们主要介绍 RBF 学习的聚类方法、梯度学习方法，及正交最小二乘（OLS）优选算法。这些算法也是最常用的 RBF 网学习算法，其它的算法，如资源分配网络（RAN）、进化优选算法等，将在以后各章中介绍。

在以下 RBF 网学习算法中，$\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_N$ 为样本输入，相应的样本输出（教师信号）为 y_1，y_2，…，y_N，网络中第 j 个隐节点的激活函数为 $\Phi_j(\cdot)$。

4.4.1 聚类方法

该方法是最经典的 RBF 网学习算法，由 Moody 与 Darken 提出[MoDa1989]。其思路是先用无监督学习（用 $k-means$ 算法对样本输入进行聚类）方法确定 RBF 网中 h 个隐节点的数据中心，并根据各数据中心之间的距离确定隐节点的扩展常数，然后用有监督学习（梯度法）训练各隐节点的输出权值。

假设 k 为迭代次数，第 k 次叠代时的聚类中心为 $\boldsymbol{c}_1(k)$，$\boldsymbol{c}_2(k),...,\boldsymbol{c}_h(k)$，相应的聚类域为 $w_1(k)$，$w_2(k),...,w_h(k)$。$k-means$ 聚类算法确定 RBF 网数据中心 $\boldsymbol{c}_i$ 和扩展常数 δ_i 的步骤如下：

(1) 算法初始化：选择 h 个不同的初始聚类中心，并令 $k=1$。初始聚类中心的方法很多，比如说从样本输入中随机选取，或者选择前 h 个样本输入，但这 h 个初始数据中心必须取不同值。

(2) 计算所有样本输入与聚类中心的距离 $\left\|\boldsymbol{X}_j - \boldsymbol{c}_i(k)\right\|, i=1,2,...,h,\ j=1,2,...,N$。

(3) 对样本输入 $\boldsymbol{X}_j$，按最小距离原则对其进行分类：即当

$i(\boldsymbol{X}_j)=\min_i\|\boldsymbol{X}_j-\boldsymbol{c}_i(k)\|, i=1,2,...,h$ 时， $\boldsymbol{X}_j$ 即被归为第 i 类，即 $\boldsymbol{X}_j\in w_i(k)$。

(4) 重新计算各类的新的聚类中心：$\boldsymbol{c}_i(k+1)=\frac{1}{N_i}\sum_{\boldsymbol{x}\in w_i(k)}\boldsymbol{x}, i=1,2,...,h$，其中 N_i 为第 i 个聚类域 $w_i(k)$ 中包含的样本数。

(5) 如果 $\boldsymbol{c}_i(k+1)\neq\boldsymbol{c}_i(k)$，转(2)，否则聚类结束，转(6)。

(6) 根据各聚类中心之间的距离确定各隐节点的扩展常数。隐节点的扩展常数取 $\delta_i=\kappa d_i$，其中 d_i 为第 i 个数据中心与其他最近的数据中心之间的距离，即 $d_i=\min_{j\neq i}\|\boldsymbol{c}_j-\boldsymbol{c}_i(k)\|$，$\kappa$ 称重叠系数。

一旦各隐节点的数据中心和扩展常数确定了，输出权矢量 $\boldsymbol{w}=[w_1,w_2,...,w_h]^T$ 就可以用有监督学习方法（如梯度法）训练得到，但更简洁的方法是使用最小二乘方法（LMS）直接计算。假定当输入为 $\boldsymbol{X}_i$，$i=1,2,\cdots$，N 时，第 j 个隐节点的输出如前面式（4.7）所示，则隐层输出阵为

$$\hat{\boldsymbol{H}}=[h_{ij}] \tag{4.15}$$

则 $\hat{\boldsymbol{H}}\in R^{N\times h}$。如果 RBF 网的当前权值为 $\boldsymbol{w}=[w_1,w_2,...,w_h]^T$（待定），则对所有样本，网络输出矢量为

$$\hat{\boldsymbol{y}}=\hat{\boldsymbol{H}}\boldsymbol{w} \tag{4.16}$$

令 $\varepsilon=\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|$ 为逼近误差，则如果给定了教师信号 $\boldsymbol{y}=[y_1,y_2,...,y_N]^T$ 并确定了 $\hat{\boldsymbol{H}}$，便可通过最小化下式求出网络的输出权值 $\boldsymbol{w}$：

$$\varepsilon=\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|=\|\boldsymbol{y}-\hat{\boldsymbol{H}}\boldsymbol{w}\| \tag{4.17}$$

通常 $\boldsymbol{w}$ 可用最小二乘法（LMS）求得：

$$\boldsymbol{w}=\hat{\boldsymbol{H}}^{+}\boldsymbol{y} \tag{4.18}$$

其中，$\hat{\boldsymbol{H}}^{+}$ 为 $\hat{\boldsymbol{H}}$ 的伪逆：

$$\hat{\boldsymbol{H}}^{+}=(\hat{\boldsymbol{H}}^{T}\hat{\boldsymbol{H}})^{-1}\hat{\boldsymbol{H}}^{T} \tag{4.19}$$

4.4.2 梯度训练方法

RBF 网的梯度训练方法[Hayk2001]与 BP 算法训练多层感知器的原理类似，也是通过最小化目标函数实现对各隐节点数据中心、扩展常数和输出权值的调节。这里给出一种带遗忘因子的单输出 RBF 网学习方法，此时神经网络学习的目标函数为

$$E = \frac{1}{2}\sum_{j=1}^{N}\beta_j e_j^2 \tag{4.20}$$

其中 β_j 为遗忘因子，误差信号 e_j 定义为

$$\begin{aligned} e_j &= y_j - F(\boldsymbol{X}_j) \\ &= y_j - \sum_{i=1}^{h} w_i \Phi_i(\boldsymbol{X}_j) \end{aligned} \tag{4.21}$$

由于神经网络函数 $F(\boldsymbol{X})$ 对数据中心 $\boldsymbol{c}_i$、扩展常数 r_i 和输出权值 w_i 的梯度分别为：

$$\nabla_{c_i} F(\boldsymbol{X}) = \frac{2w_i}{r_i^2}\Phi_i(\boldsymbol{X})(\boldsymbol{X} - \boldsymbol{c}_i) \tag{4.22}$$

$$\nabla_{r_i} F(\boldsymbol{X}) = \frac{2w_i}{r_i^3}\Phi_i(\boldsymbol{X})\|\boldsymbol{X} - \boldsymbol{c}_i\|^2 \tag{4.23}$$

$$\nabla_{w_i} f(\boldsymbol{X}) = \Phi_i(\boldsymbol{X}) \tag{4.24}$$

考虑所有训练样本和遗忘因子的影响，$\boldsymbol{c}_i$、r_i 和 w_i 的调节量为：

$$\Delta\boldsymbol{c}_i = \eta\frac{w_i}{r_i^2}\sum_{j=1}^{N}\beta_j e_j \Phi_i(\boldsymbol{X}_j)(\boldsymbol{X}_j - \boldsymbol{c}_i) \tag{4.25}$$

$$\Delta r_i = \eta\frac{w_i}{r_i^3}\sum_{j=1}^{N}\beta_j e_j \Phi_i(\boldsymbol{X}_j)\|\boldsymbol{X}_j - \boldsymbol{c}_i\|^2 \tag{4.26}$$

$$\Delta w_i = \eta\sum_{j=1}^{N}\beta_j e_j \Phi_i(\boldsymbol{X}_j) \tag{4.27}$$

其中 $\Phi_i(\boldsymbol{X}_j)$ 为第 i 个隐节点对 $\boldsymbol{X}_j$ 的输出，η 为学习率。

4.4.3 正交最小二乘（OLS）学习算法

选择 RBF 网数据中心的另一种方法是由 Chen 等人提出的正交最小二乘算法（简称 OLS 算法）[ChCo1991]。该方法从样本输入中选取数据中心，思路如下：如果把所有样本输入均作为数据中心，并令各扩展常数取相同值，则根据前面 4.3.1 节提到的 Micchelli 定理，隐层输出阵 $\boldsymbol{H} \in R^{N\times N}$ 是可逆的，于是目标输出 $\boldsymbol{y}$ 可以由 $\boldsymbol{H}$ 的 N 个列向量线性表出。但是，$\boldsymbol{H}$ 的 N 个列向量对 $\boldsymbol{y}$ 的能量贡献显然是不同的，因此我们可以从 $\boldsymbol{H}$ 的 N 个列向量中按能量贡献大小依次找出 $M \le N$ 个向量构成 $\hat{\boldsymbol{H}} \in R^{N\times M}$，直至满足给定误差 ε，即

$$\left\| \boldsymbol{y} - \hat{\boldsymbol{H}}\boldsymbol{w}_0 \right\| < \varepsilon \tag{4.28}$$

其中 $\boldsymbol{w}_0$ 是使 $\left\| \boldsymbol{y} - \hat{\boldsymbol{H}}\boldsymbol{w} \right\|$ 最小的最优权矢量 $\boldsymbol{w}$ 的值。显然，选择不同的 $\hat{\boldsymbol{H}}$，式（4.28）的逼近误差是不同的。是否选择了一个最优的 $\hat{\boldsymbol{H}}$，直接影响着 RBF 网的性能。而一旦确定了 $\hat{\boldsymbol{H}}$，也就确定了 RBF 网的数据中心。

下面简要介绍这里提到的能量贡献的计算原理。假定目标输出 $\boldsymbol{y}$ 可由 $\boldsymbol{N}$ 个互相正交的矢量（不一定是单位矢量）$\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N$ 线性表出，即

$$\boldsymbol{y} = \sum_{i=1}^{N} a_i \boldsymbol{x}_i \tag{4.29}$$

上式右乘 $\boldsymbol{x}_i^T$ 后，我们有

$$\boldsymbol{x}_i^T \boldsymbol{y} = a_i \left\| \boldsymbol{x}_i \right\|^2, \quad i = 1,2,...,N \tag{4.30}$$

于是我们有

$$\boldsymbol{y}^T \boldsymbol{y} = \sum_{i=1}^{N} a_i \left\| \boldsymbol{x}_i \right\|^2 \tag{4.31}$$

$$1 = \sum_{i=1}^{N} \frac{a_i \left\| \boldsymbol{x}_i \right\|^2}{\boldsymbol{y}^T \boldsymbol{y}} \tag{4.32}$$

因此选择 M 个基矢量时的能量总贡献为

$$g_A = \sum_{i=1}^{M} g_i = \sum_{i=1}^{M} \frac{a_i \left\| \boldsymbol{x}_i \right\|^2}{\boldsymbol{y}^T \boldsymbol{y}} \tag{4.33}$$

注意注意上式中$0 \le g_A \le 1$。M 越大，g_A就越大，逼近精度就越高。如果$M = N$，即选择了所有的基矢量，则逼近精度最高，此时$g_A = 1$。

式（4.29）中的各$\boldsymbol{x}_i$是相互正交的，而$\boldsymbol{H}$的各列并不正交，因此正交最小二乘算法(OLS)对$\boldsymbol{H}$的列的选择是在对$\boldsymbol{H}$作 Gram-Schmidt 正交化的过程中实现的。

Gram-Schmidt 正交化选择数据中心的步骤如下：

(1) 计算隐节点输出阵$\boldsymbol{H}$，并令$\boldsymbol{H}$的N个列向量为$\boldsymbol{P}_1^1, \boldsymbol{P}_1^2, \cdots, \boldsymbol{P}_1^N$，它们构成$N$维欧氏空间$E_N^H$；

(2) 把输出数据矢量$\boldsymbol{y}$投影到$\boldsymbol{P}_1^1, \boldsymbol{P}_1^2, \cdots, \boldsymbol{P}_1^N$上，如果$\boldsymbol{y}$与某一个$\boldsymbol{P}_1^k$具有最大的夹角，即$\dfrac{\boldsymbol{y}^T \boldsymbol{P}_1^k}{\left\|\boldsymbol{y}\right\| \left\|\boldsymbol{P}_1^k\right\|}$的绝对值达最大(表示该$\boldsymbol{P}_1^k$对$\boldsymbol{y}$有最大能量贡献)，则把$\boldsymbol{P}_1^k$对应的样本输入选为第一个数据中心，$\boldsymbol{P}_1^k$构成一维欧氏空间$E_1$。

(3) 用 4.4.1 节提到的广义逆方法计算网络的输出权值（包括偏移），并得到网络对样本的训练误差。如果误差小于目标值则终止算法，否则对前一步中剩下的$N-1$个向量作 Gram-Schmidt 正交化，使之正交于E_1，得到$\boldsymbol{P}_2^1, \boldsymbol{P}_2^2, \cdots, \boldsymbol{P}_2^{N-1}$；

(4) 找出与$\boldsymbol{y}$有最大投影的$\boldsymbol{P}_2^j$，选择与之对应的样本输入为第二个数据中心；计算输出权值和训练误差，并判断是否终止算法。

(5) 重复以上步骤，直至找到M个数据中心，使网络的训练误差小于给定值。

注意，上述算法可以自动设计满足精度要求的网络结构。我们的仿真发现，尽管其结构不一定是最优的，但网络规模确实是相对较小的。

4.5 RBF 网的学习动态

梯度学习算法是 RBF 网训练时最常用的算法，这一节讨论了参数采用梯度算法训练时 RBF 网的学习动态，即学习过程中各隐节点参数的变化情况。主要结论包括：当网络中不存在冗余隐节点时，各隐节点将移向样本输入的加权聚类中心；当网络中出现冗余隐节点时，这些冗余隐节点将出现萎缩、衰减、外移或重合现象。通过给出 RBF 网不存

在冗余隐节点和存在冗余隐节点时参数变化的几个算例，演示了 RBF 网的学习动态。最后根据得到的结论，对资源分配网络(RAN)和最小 RBF 网(MRBF)这两类启发式 RBF 网学习算法的合理性进行了解释，并给出了提高 RBF 网性能的一些建议。

RBF 的设计方法很多，除了上一节介绍的一些离线学习方法外，还有一些在线学习方法。Platt 提出的资源分配网络(RAN)[Platt1991]就是一种著名的在线 RBF 网学习算法，该算法在学习过程中根据需要动态分配隐节点，不分配隐节点时则用梯度法调节各隐节点的数据中心和输出权值；Yingwei 等人的 MRBF 方法[YiSu1997a，YiSu1997b]则在 RAN 的基础上增加了隐节点的删除操作。但这些方法都是启发式设计方法，尽管仿真和实际应用均说明了这些方法的有效性，但其合理性仍需要进一步解释。

神经网络的学习动态，即学习过程中各隐节点参数的变化情况，是一个令人感兴趣的研究课题。理解神经网络的学习动态，对理解神经网络的动力学行为，改进神经网络泛化能力，指导神经网络结构设计等具有积极意义。Wilson 等人[WiBl1997]研究了用于分类的多层感知器（MLP）网络的学习动态，并对现有的 MLP 训练算法提出了改进性能并简化结构的很好的建议。对采用线性隐节点的两层线性网络来说，当使用梯度下降算法时，权值的最终解仅仅是解曲面上最靠近初始权值的点[PoIo1992]。对一般的以 sigmoid 函数为激活函数的神经网络，最终权值将收敛到目标函数的某个局部最小点，且该最小点最接近于初始状态时的目标函数值[AtJi1997]。

对 RBF 网来说，由于局部特性的影响，其学习动态将明显不同于线性网络或 BP 网。如果 RBF 网学习过程中各隐节点的数据中心、扩展常数和输出权值是可调的，则学习过程中各隐节点参数的变化，最终体现为各隐节点数据中心的位置、扩展常数（或宽度）的影响范围及其输出权值的变化。由于梯度法是最常用的 RBF 网训练算法，本文研究了各隐节点参数用梯度法训练时 RBF 网的学习动态，得到了以下结果：当网络中不存在冗余隐节点时，各隐节点数据中心将移向所有样本输入的加权聚类中心；而当网络中存在冗余隐节点时，这些冗余隐节点将出现萎缩、衰减、外移或重合现象，从而可以被合并或删除。我们通过给出 RBF 网中非冗余隐节点和冗余隐节点参数变化的几个算例，演示了 RBF 网的学习动态。在最后的讨论中，我们用得到的结论，对资源分配网络及最小 RBF 网这两种常用的启发式在线 RBF 网学习算法的作用机理进行了解释，说明这些方法的合理性，并给出了提高 RBF 网性能的一些建议。

4.5.1 定义

考虑一个 $m-h-1$ 结构的 RBF 网，其中 m 和 h 分别为网络的输入节点和隐节点数。神经网络具有多输出的情况可以简单类推，这里不再介绍。假定第 i 个隐节点（简称为隐节点 i）的激活函数为 $\phi_i(\boldsymbol{x})$，其形式采用最常用的 Gaussian 型径向基函数，即 $\phi_i(\boldsymbol{x})=e^{-\frac{\|\boldsymbol{x}-\boldsymbol{c}_i\|^2}{r_i^2}}$，其中 $\boldsymbol{x}\in R^m$ 为 RBF 网输入，$\boldsymbol{c}_i\in R^m$ 为隐节点 i 的数据中心，r_i 为该隐节点的扩展常数(也称为宽度)。于是 RBF 网模型为：

$$f(\boldsymbol{x},\boldsymbol{\theta}_{(h)})=\sum_{i=1}^{h}w_i\phi_i(\boldsymbol{x}) \tag{4.34}$$

其中，w_i 为隐节点 i 的输出权值，$\boldsymbol{\theta}_{(h)}$ 表示 RBF 网中所有 h 个隐节点的参数（包

括各隐节点的数据中心、扩展常数及输出权值）组成的矢量，$f\left(\boldsymbol{x},\boldsymbol{\theta}_{(h)}\right)\in R$ 表示各隐节点参数为 $\boldsymbol{\theta}_{(h)}$，输入为 $\boldsymbol{x}$ 时网络的输出。我们定义整个网络的输出权矢量为 $\boldsymbol{w}=\left[w_1,w_2,...,w_h\right]^T$。

下面我们以 $Net_{(h)}$ 表示含 h 个隐节点组成的 RBF 网，并以三元组 $\theta_i=\left\{\boldsymbol{c_i},w_i,r_i\right\}$ 表示网络中第 i 个隐节点的参数。由于一个 RBF 网可由其所有隐节点参数唯一标志，于是网络中所有参数也可表示为 $\boldsymbol{\theta}_{(h)}=\left\{\theta_i,i=1,2,...,h\right\}$。

假定欲学习的目标函数 $g\left(x\right)$ 的定义域为 $D\subset R^m$。RBF 网的学习问题，是通过选择适当数目的隐节点(包括确定它们的数据中心、扩展常数、输出权值)，使神经网络函数 $f\left(\boldsymbol{x},\boldsymbol{\theta}_{(h)}\right)$ 在 D 内以给定精度逼近 $g\left(\boldsymbol{x}\right)$，即满足以下的性能指标函数：

$$P\left(\boldsymbol{\theta}_{(h)}\right)=\left\|f\left(\boldsymbol{x},\boldsymbol{\theta}_{(h)}\right)-g\left(\boldsymbol{x}\right)\right\|<\varepsilon \tag{4.35}$$

其中 $\left\|f\left(\boldsymbol{x},\boldsymbol{\theta}_{(h)}\right)-g\left(\boldsymbol{x}\right)\right\|=\int_{x\in D}\left(f\left(\boldsymbol{x},\boldsymbol{\theta}_{(h)}\right)-g\left(\boldsymbol{x}\right)\right)^2 d\boldsymbol{x}$，学习精度 ε 是一个较小的正数。在实际应用中，我们只能得到目标函数 $g\left(\boldsymbol{x}\right)$ 的一组样本，此时式（4.35）用下式代替：

$$P\left(\boldsymbol{\theta}_{(h)}\right)=\sum_{i=1}^{N}\left(y_i-f\left(\boldsymbol{x}_i,\boldsymbol{\theta}_{(h)}\right)\right)^2<\varepsilon \tag{4.36}$$

其中 $\left(\boldsymbol{x}_i,y_i\right)$，$i=1,2,...,N$ 为训练样本，每个训练样本都是目标函数的一个实现，其中 $\boldsymbol{x}_i\in D$ 为样本输入，$y_i\in R$ 为样本输出。在本文的讨论中，y_i 可以含有噪声。

我们称 N 个样本输入组成样本输入阵为 $\boldsymbol{X}=\left[\boldsymbol{x_1},\boldsymbol{x_2},...,\boldsymbol{x}_N\right]^T$，相应的样本输出（目标输出，或教师输出）矢量为 $\boldsymbol{Y}=\left[y_1,y_2,...,y_N\right]^T$。隐节点 i 对所有样本的输出矢量为 $\boldsymbol{o_i}=\left[o_{i1},o_{i2},...,o_{iN}\right]^T$，其中 $o_{ij}=e^{-\frac{\left\|x_j-c_i\right\|^2}{r_i^2}}$ 为输入第 $\boldsymbol{x}_j$ 时隐节点 i 的输出。定义含 h 个隐节点的神经网络输出对目标输出的误差矢量为：

$$\boldsymbol{e_h}=\boldsymbol{Y}-\sum_{i=1}^{h}w_i\boldsymbol{o_i} \tag{4.37}$$

显然，给定样本后，e_h 也是神经网络参数的函数。因此（4.36）式中的性能指标函数可以表示为 $P\left(\boldsymbol{\theta}_{(h)}\right)=\boldsymbol{e_h^T e_h}$。

如果学习后的神经网络参数满足 $P\left(\boldsymbol{\theta}_{(h)}\right)=0$，则称网络对所有样本实现了完全学习。

给定一组样本后，对一个固定隐节点数的 RBF 网而言，其性能指标函数 $P\left(\boldsymbol{\theta}_{(h)}\right)$ 的曲面形状是极为复杂的。$P\left(\boldsymbol{\theta}_{(h)}\right)$ 有许多局部最小点，由于 RBF 网的对称性（通过交换任意两个隐节点的参数），每个局部最小点都将出现 $h!$ 次。本文用到的梯度算法是最普通的，只要求：如果 $\boldsymbol{\theta}_{(h)}^*$ 是 $P\left(\boldsymbol{\theta}_{(h)}\right)$ 的一个局部最小点，且当前 RBF 网的隐节点参数 $\boldsymbol{\theta}_{(h)}$ 在 $\boldsymbol{\theta}_{(h)}^*$ 的邻域内，即 $\left\|\boldsymbol{\theta}_{(h)}-\boldsymbol{\theta}_{(h)}^*\right\|<\delta$（$\delta$ 为小的正数），则当算法收敛时，总有 $\boldsymbol{\theta}_{(h)}=\boldsymbol{\theta}_{(h)}^*$。

为便于后面描述，我们再引入以下概念：

最小结构网络：在能对给定目标函数或训练样本实现给定精度学习的 RBF 网中，称具有最少隐节点数的网络为最小结构网络。

隐节点重合 :算法收敛时某两个隐节点具有相同的数据中心和扩展常数的现象称为隐节点重合。如果这两个隐节点还具有相同的输出权值则称为隐节点完全重合。

隐节点外移：算法收敛前某隐节点 i 对所有样本的输出矢量 $\boldsymbol{o}_i \neq 0$，而算法收敛时该隐节点数据中心已位于样本输入区域 D 外，且其扩展常数也已经不足以影响 D 内的样本，导致 $\boldsymbol{o}_i = 0$ 的现象称为隐节点外移。

隐节点衰减：算法收敛前某隐节点 i 的输出权值不为零，而算法收敛时 $w_i = 0$ 的现象称为隐节点衰减。

隐节点萎缩：算法收敛前某隐节点 i 对所有样本的输出矢量 $\boldsymbol{o}_i \neq 0$，而算法收敛时该隐节点数据中心虽位于样本输入区域 D 内，且其扩展常数较小，已经不足以影响附近的样本，导致 $\boldsymbol{o}_i = 0$ 的现象称为隐节点萎缩。

4.5.2 主要结论

在训练的初始阶段，当神经网络还没有充分学习，或者神经网络的隐节点数远不足以拟合所有训练样本，神经网络都将不会出现冗余隐节点；而随着学习的进行，如果网络的初始规模较大，则网络中可能出现冗余隐节点。下面我们分别以当前网络不存在冗余隐节点和存在冗余隐节点两种情况，介绍 RBF 网的学习动态。

1) 当前网络不存在冗余隐节点的情况

虽然我们讨论的是 RBF 网学习时各隐节点参数的变化情况，但不失一般性，我们可以先考虑最后一个隐节点（即隐节点 h）参数，即 w_h， $\boldsymbol{c}_h$， r_h 的变化情况。

令 $e_{h-1,j} = y_j - \sum_{i=1}^{h-1} w_i o_{ij}$，则 $\boldsymbol{e}_{h-1} = \left[e_{h-1,1}, e_{h-1,2}, ..., e_{h-1,N}\right]^T = \boldsymbol{Y} - \sum_{i=1}^{h-1} w_i \boldsymbol{o}_i$ 表示前 $h-1$ 个隐节点对目标输出的逼近残差矢量，于是 $\boldsymbol{e}_h = \boldsymbol{e}_{h-1} - w_h \boldsymbol{o}_h$，且有

$$P\left(\boldsymbol{\theta}_{(h)}\right) = \boldsymbol{e}_{h-1}^T \boldsymbol{e}_{h-1} - 2w_h \boldsymbol{e}_{h-1}^T \boldsymbol{o}_h + w_h^2 \boldsymbol{o}_h^T \boldsymbol{o}_h \tag{4.38}$$

求解 $P\left(\boldsymbol{\theta}_{(h)}\right)$ 对 $\boldsymbol{c}_h$， r_h 和 w_h 的偏导数，得到：

$$\frac{\partial P}{\partial \boldsymbol{c}_h} = -2w_h \frac{\partial \boldsymbol{o}_h^T}{\partial \boldsymbol{c}_h} \boldsymbol{e}_{h-1} + 2w_h^2 \frac{\partial \boldsymbol{o}_h^T}{\partial \boldsymbol{c}_h} \boldsymbol{o}_h \tag{4.39}$$

$$\frac{\partial P}{\partial r_h} = -2w_h \frac{\partial \boldsymbol{o}_h^T}{\partial r_h} \boldsymbol{e}_{h-1} + 2w_h^2 \frac{\partial \boldsymbol{o}_h^T}{\partial r_h} \boldsymbol{o}_h \tag{4.40}$$

$$\frac{\partial P}{\partial w_h} = -2\left(\boldsymbol{e}_{h-1} - w_h \boldsymbol{o}_h\right)^T \boldsymbol{o}_h \tag{4.41}$$

由于式（4.39）和（4.40）中 $\frac{\partial \boldsymbol{o}_h^T}{\partial \boldsymbol{c}_h} = \frac{2}{r_h^2}\left[o_{h1}(\boldsymbol{x}_1 - \boldsymbol{c}_h), o_{h2}(\boldsymbol{x}_2 - \boldsymbol{c}_h), \cdots, o_{hN}(\boldsymbol{x}_N - \boldsymbol{c}_h)\right]$，$\frac{\partial \boldsymbol{o}_h^T}{\partial r_h} = \frac{2}{r_h^3}\left[o_{h1}\left\|\boldsymbol{x}_1 - \boldsymbol{c}_h\right\|^2, o_{h2}\left\|\boldsymbol{x}_2 - \boldsymbol{c}_h\right\|^2, \cdots, o_{hN}\left\|\boldsymbol{x}_N - \boldsymbol{c}_h\right\|^2\right]$，代入后得到

$$\frac{\partial P}{\partial \boldsymbol{c_h}} = -\frac{4w_h}{r_h^2}\left[o_{h1}(\boldsymbol{x_1}-\boldsymbol{c_h}), o_{h2}(\boldsymbol{x_2}-\boldsymbol{c_h}), \cdots, o_{hN}(\boldsymbol{x_N}-\boldsymbol{c_h})\right]\left(\boldsymbol{e_{h-1}}-w_h\boldsymbol{o_h}\right)$$
$$= -\frac{4w_h}{r_h^2}\sum_{i=1}^{N}\left(e_{h-1,i}-w_h o_{hi}\right)o_{hi}(\boldsymbol{x_i}-\boldsymbol{c_h}) = -\frac{4w_h}{r_h^2}\sum_{i=1}^{N}e_{h,i}o_{hi}(\boldsymbol{x_i}-\boldsymbol{c_h}) \tag{4.42}$$

$$\frac{\partial P}{\partial r_h} = -\frac{4w_h}{r_h^3}\left[o_{h1}\left\|\boldsymbol{x_1}-\boldsymbol{c_h}\right\|^2, o_{h2}\left\|\boldsymbol{x_2}-\boldsymbol{c_h}\right\|^2, \cdots, o_{hN}\left\|\boldsymbol{x_N}-\boldsymbol{c_h}\right\|^2\right]\left(\boldsymbol{e_{h-1}}-w_h\boldsymbol{o_h}\right)$$
$$= -\frac{4w_h}{r_h^3}\sum_{i=1}^{N}\left\|\boldsymbol{x_i}-\boldsymbol{c_h}\right\|^2 o_{hi}\left(e_{h-1,i}-w_h o_{hi}\right) = -\frac{4w_h}{r_h^3}\sum_{i=1}^{N}e_{h,i}o_{hi}\left\|\boldsymbol{x_i}-\boldsymbol{c_h}\right\|^2 \tag{4.43}$$

我们知道，负梯度方向是性能指标下降最快的方向。由式（4.42）和式（4.43）可见：由于 $o_{hi}>0$，因此对样本输入 $\boldsymbol{x_i}$，如果神经网络输出小于目标值(即 $e_{h,i}>0$)，则数据中心 $\boldsymbol{c_h}$ 被拉向 $\boldsymbol{x_i}$，同时扩展常数 r_h 增加；反之，如果神经网络输出大于目标值(即 $e_{h,i}<0$)，则数据中心 $\boldsymbol{c_h}$ 将被推离 $\boldsymbol{x_i}$，同时扩展常数 r_h 减小。$\boldsymbol{c_h}$ 每次调整的最终方向，及扩展常数 r_h 是否增加则由所有样本共同作用决定。

那么算法收敛时，w_h，$\boldsymbol{c_h}$，r_h 最终将取什么值呢？由于梯度算法最终将收敛到性能指标函数的局部最小点，因此当算法最终收敛时必然同时满足 $\frac{\partial P}{\partial \boldsymbol{c_h}}=0$，$\frac{\partial P}{\partial r_h}=0$，$\frac{\partial P}{\partial w_h}=0$，此时由式（4.42)、(4.43）和式（4.41）可得：

$$-\frac{4w_h}{r_h^2}\sum_{i=1}^{N}e_{h,i}o_{hi}(\boldsymbol{x_i}-\boldsymbol{c_h})=0 \tag{4.44}$$

$$-\frac{4w_h}{r_h^2}\sum_{i=1}^{N}e_{h,i}o_{hi}\left\|\boldsymbol{x_i}-\boldsymbol{c_h}\right\|^2=0 \tag{4.45}$$

$$2\sum_{i=1}^{N}\left(e_{h-1,i}-w_h o_{hi}\right)o_{hi}=0 \tag{4.46}$$

此时由式（4.44)、(4.45)、(4.41)，

$$\boldsymbol{c_h}=\frac{\sum_{i=1}^{N}e_{h,i}o_{hi}\boldsymbol{x_i}}{\sum_{i=1}^{N}e_{h,i}o_{hi}} \tag{4.47}$$

$$\sum_{i=1}^{N}e_{h,i}o_{hi}\left\|\boldsymbol{x_i}-\boldsymbol{c_h}\right\|^2=0 \tag{4.48}$$

$$w_h=\frac{\boldsymbol{e_{h-1}^T o_h}}{\boldsymbol{o_h^T o_h}} \tag{4.49}$$

我们知道，N 个样本输入的几何中心（聚类中心）为 $C=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x_i}$。如果令

$$k_i = \frac{e_{h,i} o_{hi}}{\sum_{j=1}^{N} e_{h,j} o_{hj}} \tag{4.50}$$

则由式（4.49）可得：

$$\boldsymbol{c_h} = \sum_{i=1}^{N} k_i \boldsymbol{x_i} \tag{4.51}$$

此时 $\boldsymbol{c_h}$ 可看作 N 个样本输入“加权聚类中心”。从加权系数 k_i 的表达式可见：如果某样本输入 $\boldsymbol{x_i}$ 不能激活隐节点 h（o_{hi} 值较小），或者当前神经网络对该样本的误差较小（$e_{h,i}$ 较小），则该 $\boldsymbol{x_i}$ 对 $\boldsymbol{c_h}$ 的影响就较小。隐节点 h 的数据中心 $\boldsymbol{c_h}$ 最终移向那些未被其它隐节点激活且偏差较大的样本输入。这也与我们的直觉是一致的。

式（4.48）是算法收敛时 r_h 必须满足的约束条件。其中 $\boldsymbol{o_h}$ 表示隐节点 h 对所有样本的输出，$\boldsymbol{e_h} = \boldsymbol{e_{h-1}} - w_h \boldsymbol{o_h}$ 也可理解为前 $h-1$ 个隐节点对所有样本的残差被 $w_h \boldsymbol{o_h}$ 进一步作用后的剩余残差矢量。一旦 r_h 发生变化，$\boldsymbol{o_h}$ 将发生变化，最终 $\boldsymbol{e_h}$ 也将发生变化。我们知道，当 r_h 调整至最佳时 $\boldsymbol{o_h}$ 和 $\boldsymbol{e_h}$ 应该是互相垂直的，即 $\boldsymbol{o_h^T e_h} = 0$。但这里，$o'_{h,i} = d_i o_{h,i}$，其中 $d_i = \left\| \boldsymbol{x_i} - \boldsymbol{c_i} \right\|^2$，说明 r_h 的作用是隐节点 h 的加权输出 $o'_h = \left[o'_{h,1}, o'_{h,2}, ..., o'_{h,N} \right]^T$ 与 $\boldsymbol{e_h}$ 互相垂直。因此权系数 d_i 的作用是惩罚 $\left\| \boldsymbol{x} - \boldsymbol{c_i} \right\| = 0$ 附近样本对应的隐节点输出，因为在 $\left\| \boldsymbol{x} - \boldsymbol{c_i} \right\| = 0$ 附近的样本，其相应的隐节点输出受 r_h 变化的影响较小。

式（4.49）的几何意义是很明显的，它是使 $\boldsymbol{e_{h-1}}$ 和 $\boldsymbol{o_h}$ 间最小二乘误差达最小时系数 w_h 的取值。

尽管我们只讨论了隐节点 h 的参数变化规律，但由于 RBF 网的对称性，其它隐节点参数也将遵循相同的参数变化规律。因此，在 RBF 网中没有冗余的隐节点的情况下，当梯度算法收敛时，我们有以下结论：

如果 RBF 网络中没有冗余的隐节点，则在梯度算法作用下，各隐节点将移向所有样本输入的加权聚类中心。

2） 当前网络存在冗余隐节点的情况

假定含 $h-1$ 个隐节点的 RBF 网（称 $Net_{(h-1)}$）对训练样本的学习就能满足精度要求，而此时网络中有 $s \geq h$ 个隐节点，则冗余节点参数将如何调节？这里考虑只有一个冗余节点，即 $s = h$ 的情况。网络中有多个冗余隐节点的情况可以类推得到。

先考虑用 RBF 网实现某一目标函数 $g(\boldsymbol{x})$ 的情形。对此，我们有以下定理：

定理 4.1：假设目标函数 $g(\boldsymbol{x})$ 可由最小结构网络 $Net_{(h-1)}$ 实现，网络 $Net_{(h)}$ 比 $Net_{(h-1)}$ 多一个隐节点。如果 $Net_{(h)}$ 也能实现 $g(\boldsymbol{x})$，则 $Net_{(h)}$ 或者有 $h-1$ 个隐节点与 $Net_{(h-1)}$ 的隐节点完全重合，而余下的那个独立隐节点输出权值为零；或者 $Net_{(h)}$ 中有 $h-2$ 个隐节点与 $Net_{(h-1)}$ 中的隐节点完全重合，而余下的两个重合的隐节点叠加后与 $Net_{(h-1)}$ 中最后一个隐节点相等。

证明： $g(\boldsymbol{x})$可由 $Net_{(h-1)}$和 $Net_{(h)}$同时实现，故有 $g(\boldsymbol{x})=\sum_{i=1}^{h-1} w_i\phi_i(\boldsymbol{x})=\sum_{j=1}^{h} v_j\varphi_j(\boldsymbol{x})$，于是下式成立：

$$\sum_{i=1}^{h-1} w_i\phi_i(\boldsymbol{x})-\sum_{j=1}^{h} v_j\varphi_j(\boldsymbol{x})=0 \qquad (4.51)$$

由于 $Net_{(h-1)}$是最小结构网络，因此上式中所有的 w_i均不为零（否则我们可得到一个结构比 $Net_{(h-1)}$更精简的网络）。且对每个 $\phi_i(\boldsymbol{x})$，至少有一个 $\varphi_j(\boldsymbol{x})$与之线性相关（否则至少有一个 w_i不为零），因此各 $\varphi_j(\boldsymbol{x})$中至少有 $h-1$个是相互独立且与各 $\phi_i(\boldsymbol{x})$一一对应的，即最多只有一个 $\varphi_j(\boldsymbol{x})$与各 $\phi_i(\boldsymbol{x})$是独立的。此时只有下面两种情况：

(1) 如果有这样一个 $\varphi_j(\boldsymbol{x})$与各 $\phi_i(\boldsymbol{x})$都相互独立，则其输出权值必为零（否则不能保证式(20)成立），即该 $\varphi_j(\boldsymbol{x})$对应一个输出权值为零的冗余隐节点，此时式(20)变为 $\sum_{i=1}^{h-1} w_i\phi_i(\boldsymbol{x})-\sum_{j=1}^{h} v_j\varphi_j(\boldsymbol{x})=\sum_{i=1}^{h-1}(w_i-v_j)\phi_i(\boldsymbol{x})=0$，由于各 $\phi_i(\boldsymbol{x})$线性无关，因此必有 $w_i=v_i$，$i=1,2,...,h-1$。这说明 $Net_{(h)}$中有 $h-1$个隐节点与 $Net_{(h-1)}$的隐节点完全重合，而余下的那个独立隐节点输出权值为零。

(2) 如果没有一个 $\varphi_j(\boldsymbol{x})$是独立的，则必有两个重合的 $\varphi_j(\boldsymbol{x})$与某个 $\phi_i(\boldsymbol{x})$相关。类似于前面的讨论，我们也可以得到：这两个 $\varphi_j(\boldsymbol{x})$的权值相加后与对应的那个 $\phi_i(\boldsymbol{x})$的权值相等，而 $h-2$个隐节点与 $Net_{(h-1)}$中的隐节点完全重合。证毕。

在实际应用中我们只能得到目标函数 $g(\boldsymbol{x})$的一组样本。我们知道，只有当样本数充分多时目标函数才可能由其样本表示（一种极端的情况是，对任意 $\boldsymbol{x}\in R^m$，如果我们都知道 $g(\boldsymbol{x})$的值，则 $g(\boldsymbol{x})$就是唯一确定的）。因此定理 4.1 说明，训练样本充分多时隐节点的衰减和重合是必然的。而在有限样本时，冗余隐节点的学习动态要复杂得多。下面我们假定 $\boldsymbol{\theta}^*_{(h-1)}=\{\boldsymbol{\theta}_i\,|\,\boldsymbol{\theta}_i=\boldsymbol{\theta}^*_i, i=1,2,...,h-1\}$是使网络 $Net_{(h-1)}$对训练样本达到给定精度学习的一组隐节点参数，并讨论 $Net_{(h)}$中冗余隐节点的学习动态。

首先，我们有以下定理：

定理 4.2：假设 $\boldsymbol{\theta}^*_{(h-1)}=\{\boldsymbol{\theta}_i\,|\,\boldsymbol{\theta}_i=\boldsymbol{\theta}^*_i, i=1,2,...,h-1\}$时最小结构网络 $Net_{(h-1)}$对训练样本的学习能满足精度要求，网络 $Net_{(h)}$比 $Net_{(h-1)}$增加一个隐节点，则当学习过程中 $Net_{(h)}$的前 $h-1$个隐节点参数一直在 $\boldsymbol{\theta}^*_{(h-1)}$的领域内时，隐节点 h必然出现萎缩和衰减现象。

证明：考虑网络中 i个隐节点的的扩展常数和输出权值在梯度算法作用下扩展常数和输出权值的参数调整规律。神经网络函数 $f(\boldsymbol{x},\boldsymbol{\theta})$对扩展常数 r_i和输出权值 w_i的梯度分别为 $\nabla_{r_i} f(\boldsymbol{x},\boldsymbol{\theta})=\frac{2w_i}{r_i^3}\phi_i(\boldsymbol{x})\|\boldsymbol{x}-\boldsymbol{c}_i\|^2$ 和 $\nabla_{w_i} f(\boldsymbol{x},\boldsymbol{\theta})=\phi_i(\boldsymbol{x})$。在所有训练样本

的作用下，r_i 和 w_i 的调节量分别为 $\Delta r_i = \eta \sum_{j=1}^{N} e_{i,j} \frac{2w_i}{r_i^3} \phi_i(\boldsymbol{x}_j) \|\boldsymbol{x}_j - \boldsymbol{c}_i\|^2$，及 $\Delta w_i = \eta \sum_{j=1}^{N} e_{i,j} \phi_i(\boldsymbol{x}_j)$，其中学习率 η 为较小的正数。由于前 $h-1$ 个隐节点的参数已经使神经网络的训练误差较小，因此 $e_{h,j} \approx -w_h \phi_h(\boldsymbol{x}_j)$，于是对隐节点 h 有 $\Delta r_h \approx -\eta \sum_{j=1}^{N} \frac{2w_h^2}{r_h^3} \phi_h^2(\boldsymbol{x}_j) \|\boldsymbol{x}_j - \boldsymbol{c}_h\|^2$，$\Delta w_h \approx -\eta \sum_{j=1}^{N} w_h \phi_h^2(\boldsymbol{x}_j)$，显然 $\Delta r_h < 0$，而 Δw_h 的符号总是与 w_h 相反的。于是在后一参数修正时刻 $t+1$ 必然有：$r_h(t+1) < r_h(t)$，$|w_h(t+1)| < |w_h(t)|$，即隐节点 h 将出现萎缩和衰减现象。证毕。

定理 4.2 说明，当网络中存在冗余隐节点时，如果其它隐节点对样本的学习就能达到较小的误差，则冗余隐节点将出现萎缩和衰减。而在该隐节点的萎缩和衰减过程中，如果其输出权值最终变为为零，或者其扩展常数小到一定程度，最终都将使该隐节点对所有样本的的加权输出矢量 $w_h \boldsymbol{o}_h = 0$，于是该隐节点必然停止萎缩和衰减过程。那么冗余隐节点在衰减和萎缩过程中还会出现其它动态吗？下面的定理说明：在隐节点衰减和萎缩过程中，冗余隐节点还可能出现隐节点的重合和外移现象。

定理 4.3 :假设 $\theta_{(h-1)}^* = \{\theta_i | \theta_i = \theta_i^*, i = 1,2,...,h-1\}$ 时最小结构网络 $Net_{(h-1)}$ 对训练样本实现完全学习，网络 $Net_{(h)}$ 比 $Net_{(h-1)}$ 增加一个隐节点，则当学习过程中 $Net_{(h)}$ 的隐节点参数一直满足：当 k 为隐节点 h 或某个隐节点 $j < h$ 时，其隐节点参数满足 $\|\boldsymbol{c}_k - \boldsymbol{c}_j^*\| < \delta$，$\|r_k - r_j^*\| < \delta$，且 $\|w_j + w_h - w_j^*\| < \delta$，而当 k 为其它隐节点时则满足 $\|\boldsymbol{c}_k - \boldsymbol{c}_k^*\| < \delta$，$\|r_k - r_k^*\| < \delta$，$\|w_k - w_k^*\| < \delta$，其中 δ 为小的正数，则隐节点 h 和隐节点 j 必然出现重合现象，且算法收敛时 $w_j + w_h = w_j^*$。

证明：在训练样本确定的情况下，式（4）的性能指标函数在其定义域为 $D \subset R^m$ 内是无穷阶连续可微的。由于当隐节点参数 $\boldsymbol{\theta}_{(h-1)}^* = \{\boldsymbol{\theta}_i | \boldsymbol{\theta}_i = \boldsymbol{\theta}_i^*, i = 1,2,...,h-1\}$ 时网络 $Net_{(h-1)}$ 对样本实现完全学习，因此如果从 $Net_{(h-1)}$ 中的某个隐节点 j 分离出一个与 j 重叠的隐节点 h，则 $Net_{(h-1)}$ 成为 $Net_{(h)}$ 使并使分离后的 j 和 h 满足：$\boldsymbol{c}_j = \boldsymbol{c}_h = \boldsymbol{c}_j^*$，$r_j = r_h = r_j^*$，$w_j + w_h = w_j^*$，则网络 $Net_{(h)}$ 与 $Net_{(h-1)}$ 是等价的，即 $\boldsymbol{\theta}_{(h)}^*$ 也是 $P_h(\boldsymbol{\theta}_{(h)})$ 的一个全局最小点。由于 $P_h(\boldsymbol{\theta}_{(h)})$ 无穷阶连续可微的，因此当 $Net_{(h)}$ 的隐节点初始参数在 $\boldsymbol{\theta}_{(h)}^*$ 附近时，梯度法必然能使参数收敛到 $\boldsymbol{\theta}_{(h)}^*$，即必出现隐节点重合现象。证毕。

定理 4.3 也说明了 RBF 网学习过程中的一个重要动态：在学习过程中，当算法接近完全学习，且冗余的隐节点 h 与某个隐节点 j 的数据中心和宽度均比较接近，且它们的输出权值之和也接近于 w_j^* 时，h 与 j 必然出现重合现象，从而可以将它们合并。

定理 4.4 :假设 $\boldsymbol{\theta}_{(h-1)}^* = \{\boldsymbol{\theta}_i | \boldsymbol{\theta}_i = \boldsymbol{\theta}_i^*, i = 1,2,...,h-1\}$ 时最小结构网络 $Net_{(h-1)}$ 对 $D \subset R^m$ 内训练样本的学习能满足精度要求，网络 $Net_{(h)}$ 比 $Net_{(h-1)}$ 增加一个隐节点，则当学

习过程中 $Net_{(h)}$ 的前 $h-1$ 个隐节点参数一直在 $\theta^{*}_{(h-1)}$ 的领域内，而对隐节点 h 有 $c_{h+1} \notin D$，则当算法收敛后，隐节点 h 必然出现隐节点外移现象。

证明：由于当前神经网络函数 $f(x,\theta_{(h)})$ 对数据中心 c_i 的梯度为 $\nabla_{c_i} f(x,\theta) = \frac{2w_i}{r_i^2}\phi_i(x)(x-c_i)$，考虑所有训练样本的影响，$c_i$ 的调节量为 $\Delta c_i = \eta\sum_{j=1}^{N} e_{i,j}\frac{2w_i}{r_i^2}\phi_i(x_j)(x_j-c_i)$。当 $Net_{(h)}$ 满足上述条件时，对隐节点 h 我们有 $e_{h,j} \approx -w_h\phi_h(x_j)$。因此 $\Delta c_h \approx -\eta\sum_{j=1}^{N}\frac{2w_h^2}{r_h^2}\phi_h^2(x_j)(x_j-c_h)$，这说明 Δc_h 的调节方向与所有的 $(x_j-c_h), j=1,2,...,N$ 是相反的。由于所有样本输入均位于 D 的内部，而 c_h 位于 D 的外部，因此在所有训练样本的作用下，Δc_h 最终将指向 D 的外部，即冗余隐节点 h 将出现外移现象。证毕。

定理 4.4 阐述了 RBF 网学习过程中的又一个重要动态：在 RBF 网学习过程中，当算法接近收敛，且某个冗余的隐节点 h 的数据中心移至定义域 D 的“边界外部”时，该隐节点必然离定义域 D 越来越远，直至该隐节点对 D 内的样本的影响可以忽略为止，即对所有训练样本满足 $o_h = 0$。

事实上，由定理 4.4 的证明过程可见，出现冗余隐节点外移的条件可以放得更宽：如果冗余隐节点 h 的数据中心 c_h 移至 D 的“边界附近”，即使 c_h 位于 D 的内部，即 $c_h \in D$，只要 Δc_h 的总的调节方向是向 D 的边界外部的，即所有样本产生外推的“合力”大于内拉的“合力”，隐节点 h 就会出现外移现象。

在训练一个非最小结构 RBF 网时，如果出现隐节点的衰减、萎缩、合并和外移现象，我们就可以在算法中判断这些现象，并删除这些冗余的隐节点，以保证神经网络的泛化能力[魏-徐 2001]，但在学习过程中还可能出现非最小结构 RBF 网对训练样本达到给定精度学习的情况。考虑下面的例子：(x_j, y_j)，$j=1,2,...,N$ 为取自目标函数 $g(x)=\sum_{i=1}^{N-1} w_i e^{-\frac{\|x-c_i\|^2}{r_i^2}}$ 的任意 N 个训练样本，显然这组训练样本可由 $N-1$ 个隐节点组成的 RBF 网（记为 $Net_{(N-1)}$）实现，只要该 RBF 网取目标函数中的隐节点参数。但是，对这 N 个样本，我们也可以由一个 N 个隐节点组成的 RBF 网（记为 $Net_{(N)}$）对这组样本实现完全学习，该网络的数据中心取样本输入 x_j，$j=1,2,...,N$，N 个扩展常数可以取相同的正数（任意），则根据 Micchelli 定理[Micc1986]，N 个隐节点对 N 个样本的输出矢量组成的对称矩阵 $O=[o_1,o_2,...,o_N]$ 是可逆的，则如果我们取 $Net_{(N)}$ 的输出权矢量为 $W=O^{-1}Y$，其中 $Y=[y_1,y_2,...,y_N]^T$，则 $Net_{(N)}$ 也能对这 N 个样本进行完全学习。

之所以出现这种非最小结构 RBF 网对训练样本达到给定精度学习的现象，是因为从一组有限的样本中恢复出目标函数 $g(x)$ 是一个病态问题[Tikh1977]。事实上，在有限样本的情况下，能对这些样本实现完全学习的曲线可以有很多，而一般情况下最

小结构 RBF 网只是其中最光滑的一个。

4.5.3 算例

上面讨论的是 RBF 网训练时各隐节点的学习动态。这一节给出了若干算例，只要满足算例的条件，隐节点将出现相应的动态。在所有算例中，训练样本由式（4.52）所示的目标函数产生，样本输出叠加噪声。

$$g(x_1,x_2)=2.0\exp\left[-\frac{(x_1-0.3)^2+(x_2-0.7)^2}{0.04}\right]+1.8\exp\left[-\frac{(x_1-0.6)^2+(x_2-0.4)^2}{0.04}\right] \tag{4.52}$$

显然，该目标函数可由 2 个隐节点的 RBF 网实现。训练前，产生 121 个训练样本 $((x_1,x_2),y)$，其中 (x_1,x_2) 为样本输入，$y=g(x_1,x_2)+N(0,0.1)$ 为样本输出。$N(0,0.1)$ 为均值为 0，方差为 0.1 的服从正态分布的噪声，$x_i\in\{0,0.1,...,1\}$，$i\in\{1,2\}$。我们采用梯度下降算法学习网络的三个参数，训练一定次数后结束。

1） 隐结点移向聚类中心的例子

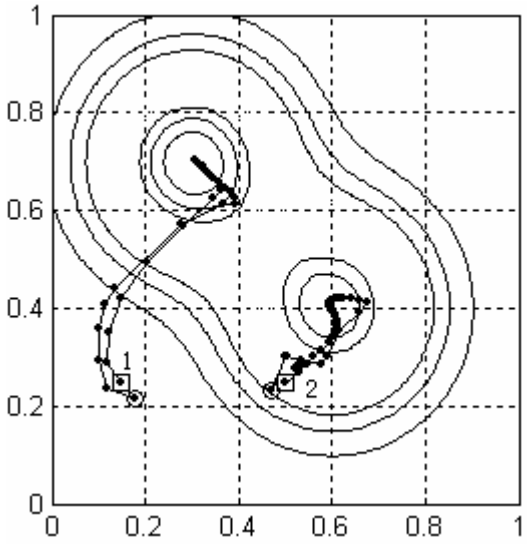

图 4.6 隐节点移向聚类中心

如果径向基网使用两个隐节点，它们的数据中心初始值分别为 $(0.15,0.25)$，$(0.50,0.25)$，宽度初值为 0.45，0.10，输出权值初值为 2.5，1.3，则经过 2000 次学习后（学习率为 0.002），数据中心分别变为 $(0.3031,0.7027)$，$(0.6061,0.3956)$，宽度值为 0.2048，0.1930，输出权值为 1.9453，1.8604。图 4.6 是学习过程中两个隐节点数据中心的移动轨迹，图中的环形线为目标函数的等高线。图中以“□”开始的曲线为两个数据中心的移动轨迹，以“○”开始的曲线为每次参数调节时理论计算得到的两个加权聚类中心的轨迹。从图中可以看到，两个隐节点的最终的加权聚类中心与目标曲线中的数据中心值是一致的，而且两个隐节点最终都移到了相应的加权聚类中心。

2） 隐结点外移的例子

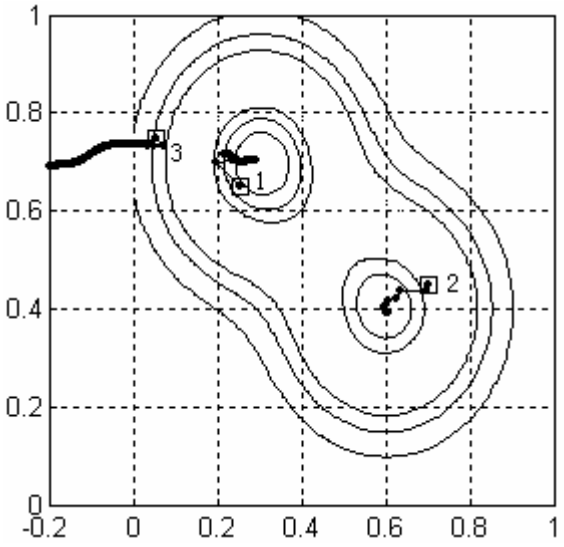

图 4.7　隐节点外移现象

如果径向基网使用三个隐节点，它们的数据中心初始值分别为$(0.25,0.65)$，$(0.70,0.45)$，$(0.05,0.75)$，宽度初值为 0.35，0.38，0.40，输出权值初值为 1.9，1.6，-1.0，则经过 1000 次学习后（学习率为 0.002），三个数据中心分别为$(0.2921,0.7025)$，$(0.5987,0.3945)$，$(-0.3188,0.6420)$，宽度值为 0.2091，0.1991，0.2377，输出权值为 1.9483，1.8407，-0.8085。图 4.7 是三个隐节点的移动轨迹，图中的环形线为目标函数的等高线，“□”为数据中心初始值。从图中可以明显看到，两个隐节点移到了相应的正确位置，第三个隐节点则远远移出了样本输入区域，出现了隐节点外移现象。

3） 隐节点重合的例子

这里的径向基网使用三个隐节点，它们的数据中心初始值分别为$(0.10,0.55)$，$(0.50,0.45)$，$(0.25,0.85)$，宽度初值为 0.08，0.28，0.46，输出权值初值为 0.3，1.6，0.8，则经过 1000 次学习后（学习率为 0.001），三个数据中心分别为$(0.3028,0.7056)$，$(0.6074,0.3865)$，$(0.3027,0.7048)$，宽度值为 0.2048，0.2040，0.2024，输出权值为 0.6793，1.7740，1.3415。图 4.8（a）是三个隐节点数据中心的移动轨迹，图 4.8（b）为两个隐节点宽度的变化曲线。从图中可以明显看到，第一、三隐节点的数据中心趋向重合。同时它们的宽度也趋向于相等，出现了隐节点重合现象。另外，它们的输出权值之和为 2.0208，与目标函数第一个隐节点的输出权值非常接近。

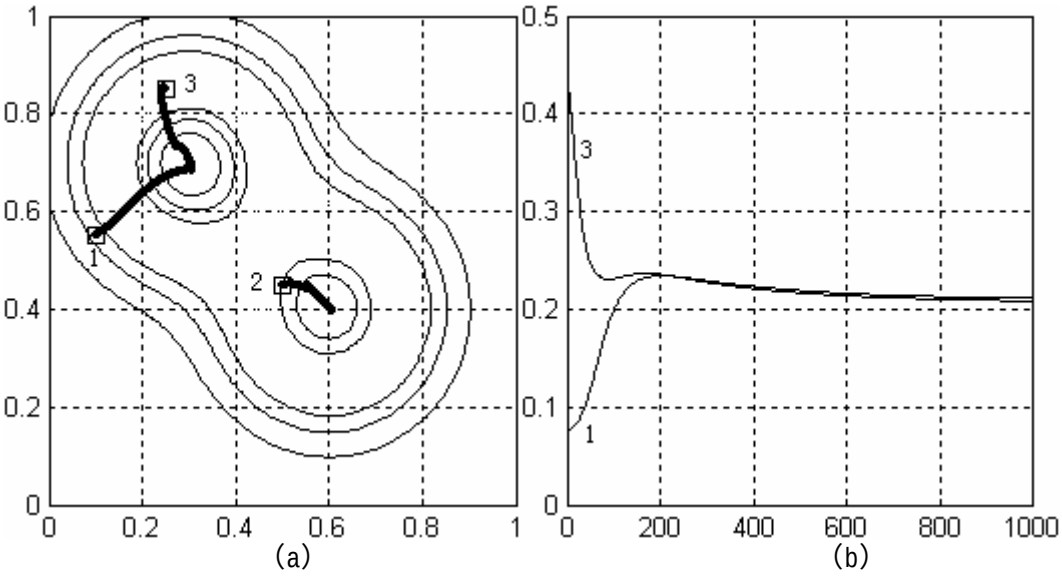

图 4.8 （a）隐节点重合现象；(b）重合隐节点的扩展常数变化

4） 隐结点衰减的例子

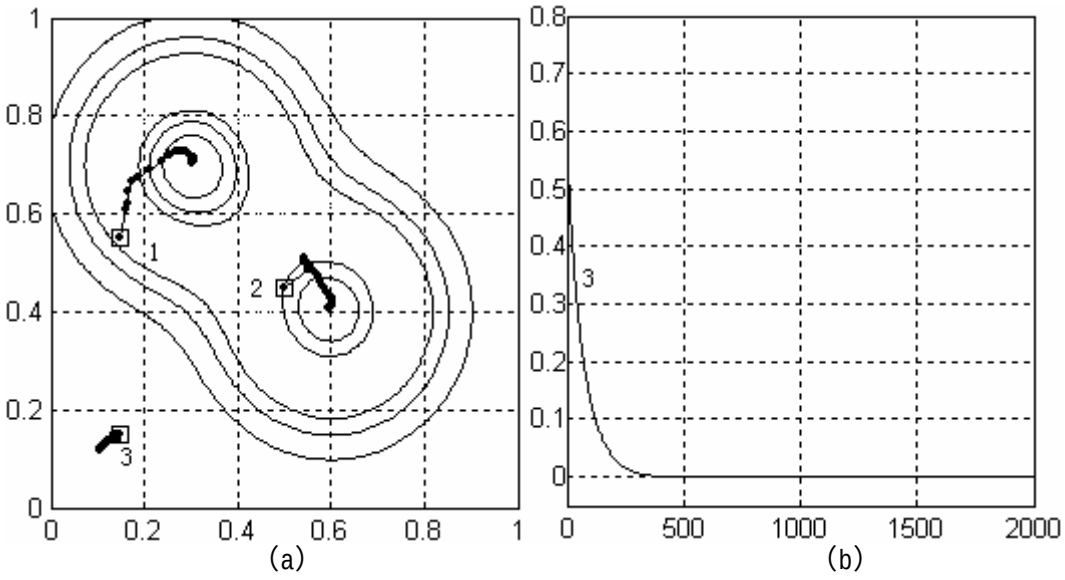

图 4.9 （a）数据中心移动轨迹；(b）衰减隐节点的输出权值变化

这里的径向基网使用三个隐节点，它们的数据中心初始值分别为$(0.15,0.55)$，$(0.50,0.45)$，$(0.15,0.15)$，宽度初值为0.15，0.28，0.50，输出权值初值为1.9，1.6，0.6，则经过 2000 次学习后（学习率为 0.002），三个数据中心分别为$(0.3008,0.7018)$，$(0.5955,0.4071)$，$(0.1033,0.1158)$，宽度值为0.1963，0.1986，0.4097，但输出权值变为1.9585，1.8868，-0.0002。图 4.9（a）是三个隐节点数据中心的移动轨迹，图 4.9（b）是第三个隐节点输出权值的变化情况。可以看到，尽管第三隐节点的数据中心位于样本输入区域内，且其扩展常数也较大，但是其输出权值衰减为接近于零，出现了隐节点衰减现象。

5） 隐结点萎缩的例子

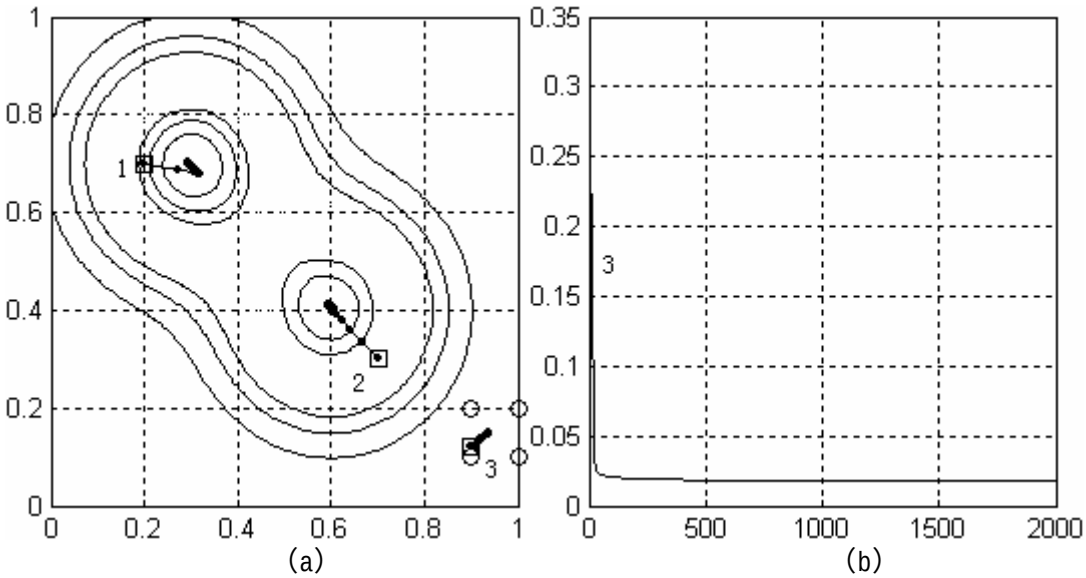

图 4.10 （a）隐节点萎缩现象；(b）萎缩隐节点的扩展常数值变化

这里的径向基网依旧使用三个隐节点，它们的数据中心初始值分别为 $(0.20,0.70)$，$(0.70,0.30)$，$(0.90,0.12)$，宽度初值为 0.25，0.15，0.30，输出权值初值为 2.5，1.2，1.0，则经过 2000 次学习后（学习率为 0.002），三个隐节点数据中心分别为 $(0.2955,0.6994)$，$(0.6069,0.4009)$，$(0.9382,0.1495)$，宽度值为 0.2013，0.1986，0.0172，输出权值变为 2.0047，1.7851，0.8942。图 4.10（a）是三个隐节点数据中心的移动轨迹，图中“○”处为样本输入，图 4.10（b）是第三个隐节点扩展常数的变化情况。可见第三个隐节点的最终宽度虽然不为零，但值较小（0.0172），致使所有训练样本都无法影响该隐节点，训练即便是离其最近的样本输入点 $(0.9,0.1)$，隐节点输出也已经很小（1.6304e-006），其影响已经可以忽略，出现了隐节点萎缩现象。另外，可以看到，在梯度算法作用下，第三个隐节点的移到了附近四个样本的中心 $(0.95,0.15)$ 附近，因为在这一点扩展常数对附近样本的影响最小，说明隐节点萎缩时数据中心确实移到了样本的“间隙”处。

4.5.3 进一步的讨论

1） RBF 网学习动态与 RAN 和 MRBF 的关系

资源分配网络（RAN）由 Platt 提出[Platt1991]，是最著名的 RBF 网络在线学习算法。该方法不断地检查新的样本输入输出对，当新样本满足“新性”(Novelty)条件时，则分配一个新节点。“新性”条件为：①当前样本输入距离最近的数据中心超过某一定值 $\delta(t)$ (距离准则)；②网络的输出与样本输出的偏差大于某一定值 $e_{\min}$ (误差准则)。两个条件同时满足则分配新的隐节点。新隐节点的数据中心取当前样本的输入，输出权值取当前神经网络输出与当前样本输出之间的偏差，而扩展常数则取与该样本输入最近的数据中心之间的距离。如果当前网络输出与新样本输出的偏差较小，或样本输入离已有的数据中心都较近时，则不分配新隐节点，此时用梯度法调整各数据中心和权值。该方法通过距离准则和误差准则在线分配必要的资源(隐节点)，

以使神经网络适应最新的动态。

考虑 RAN 的学习机制。假定网络已经学习了 $N-1$ 个样本，且当前网络中已经有 $h-1$ 个隐节点，并假定此时得到的最新的一个样本为 $(\boldsymbol{x}_N, y_N)$。如果该样本满足“ 新性 ”条件，则需要增加一个隐节点 h。由于添加新隐节点之前网络对前 $N-1$ 个样本已经学习得较好，即有 $e_{h-1,j}=0$， $j\in\{1,2,...,N-1\}$，因此新增的隐节点 h 不应该影响已有的 $h-1$ 个隐节点的作用效果，故有 $e_{h,j}=0$， $j\in\{1,2,...,N-1\}$。为了使隐节点 h 产生有效作用，其数据中心只能在 $\boldsymbol{x}_N$ 附近，且其扩展常数取值应使隐节点 h 满足：$o_{hj}=0$， $j\in\{1,2,...,N-1\}$， $o_{hN}\neq 0$。于是根据式（4.50），有

$$k_i=\begin{cases}0, & i=1,2,...,N-1\\ 1, & i=N\end{cases} \tag{4.53}$$

显然，根据式（4.51）和式（4.53），有

$$\boldsymbol{c}_h=\boldsymbol{x}_j \tag{4.54}$$

此时式（4.48）也必然是满足的。又由于此时 $o_{hN}=1$， $\boldsymbol{e}_{h-1}^T\boldsymbol{o}_h=e_{h-1,N}o_{hN}=e_{h-1,N}$，因此由式（4.49）可得

$$w_h=e_{h-1,N} \tag{4.55}$$

式（4.54）、（4.55）正是新增隐节点 h 的数据中心和输出权值，而该隐节点的扩展常数只要满足 $o_{hj}=0$， $j\in\{1,2,...,N-1\}$ 即可。文献[Platt1991]中取 $r_h=\kappa\delta_{\max}$ 显然也满足这一约束条件。因此此处的分析说明 RAN 的新隐节点分配机制是合理的。

最小 RBF 网（MRBF）是 RAN 的一种改进学习算法，该算法在 RAN 基础上增加了隐节点的删除操作，即当一个隐节点长期未被激活时，则删除该隐节点。事实上，由前面的讨论可知，当网络中存在冗余的隐节点时，冗余的隐节点可能出现外移和萎缩现象。冗余隐节点一旦外移将再也不可能被激活，而如果扩展常数足够小，则萎缩的冗余隐节点也可能出现长时间未被激活的情况，因而这些隐节点最终都可以被删除。因此 MRBF 在 RBF 网学习过程中进行隐节点的删除操作不仅是必要的，也是合理的。

2） 提高 RBF 网性能的一些建议

RBF 网设计的核心问题是设计满足精度要求的最小结构神经网络，以保证网络的泛化能力[魏-徐 2001]。但是在一般情况下，设计网络之前我们并不知道什么样的网络才是最小结构的，所以通常我们会选择一个“ 足够大 ”的网络。根据前面的讨论，我们知道，在用梯度法训练这样的 RBF 网时，算法收敛时网络中可能会出现冗余的隐节点，也可能出现非最小结构网络对训练样本达到给定精度学习的现象。因此比较合理的 RBF 网设计方法是在训练过程中尽量使非最小结构网络出现冗余隐节点，然后判断并删除（或合并）这些冗余的隐节点。根据 RBF 网的学习动态，可以考虑在用梯度法调节网络的数据中心、输出权值和偏移的基础上，综合采取以下方法：

(1) 隐节点的动态合并策略。

既然 RBF 网在学习过程中可能出现隐节点重合现象，即两个隐节点不仅数据中心比较接近，扩展常数值也基本相等，那么就应该合并这些重合的隐节点。可采用以下合并策略：假定 i， j 是两个隐节点，如果它们基本重合，则我们有：

$$\|\boldsymbol{c}_i-\boldsymbol{c}_j\|<\Delta c_{\min} \tag{4.56}$$

$$\left|r_i - r_j\right| < \Delta r_{\min} \tag{4.57}$$

其中 $\Delta c_{\min}$ 和 $\Delta r_{\min}$ 是合并阈值。于是对任意输入 x，隐节点 i， j 的输出满足

$$\phi_i(\boldsymbol{x}) \approx \phi_j(\boldsymbol{x}) \tag{4.58}$$

从而可以把这隐节点 j 合并到隐节点 i。显然，合并后隐节点的输出权值为

$$w_i = w_i + w_j \tag{4.59}$$

合并后隐节点的数据中心为

$$\boldsymbol{c}_i = \frac{1}{2}(\boldsymbol{c}_i + \boldsymbol{c}_j) \tag{4.60}$$

(2) 隐节点的动态删除策略。

我们可以通过以下面各式来判断某隐节点 i 是否出现了衰减、萎缩或外移现象，并决定是否应该删除：

$$\left|w_i\right| < w_{\min} \tag{4.61}$$

$$\left|r_i\right| < r_{\min} \tag{4.62}$$

$$\left\|\boldsymbol{o}_i\right\| < o_{\min} \tag{4.63}$$

其中 $\boldsymbol{o}_i$ 为隐节点 i 对所有样本的输出矢量， $w_{\min}$、 $r_{\min}$ 和 $o_{\min}$ 是冗余隐节点的衰减、萎缩或外移删除阈值。

在线学习算法中，式（4.63）应略作修改。此时如果隐节点 i 出现了外移现象，则连续多个样本都无法激活某隐节点，即式（4.63）改为

$$count > C_{\max} \tag{4.64}$$

此式满足时则删除该隐节点，其中 $count$ 为上次被激活到现在这段时间内隐节点未激活的次数累计值。为防止误删有用的隐节点， $C_{\max}$ 可以取较大的值.

(3) 限制各隐节点扩展常数的最小值。

隐节点出现萎缩时，该隐节点虽然不会对训练样本产生影响，但由于它位于样本输入区域内，因此可能对测试样本或工作样本产生影响，即会影响网络的泛化能力。虽然我们可以通过式（4.62）进行判断，但更好的方法是限制各隐节点扩展常数不能小于某最小值。这样一旦该隐节点出现萎缩的趋势，由于扩展常数最小值的限制，梯度算法将使该隐节点出现衰减（也可能出现外移现象），而衰减或外移的冗余隐节点对 RBF 网泛化能力的影响则要比萎缩隐节点小得多。

(4) 采用全局优化策略。

如果对训练样本达到给定精度学习的是非最小结构网络，则神经网络的泛化能力会受到影响。可以考虑在梯度算法中采用多点并行搜索策略，比如说使用进化算法结合梯度算法实现多点并行搜索，只要算法收敛时有一点出现了冗余隐节点，则就可以找到一个结构更精简的网络。

基于 RBF 网的学习动态和上述讨论，我们将在第 12 章将给出 RBF 网在线学习的资源优化网络（RON）设计方法。

4.6 仿真例子

这是上一章 3.3.3 节的 Hermit 多项式逼近的例子，这里改用 RBF 网实现逼近，其中

噪声水平和 100 个训练样本的产生方法不变。

我们用聚类方法、梯度算法及正交最小二乘算法训练 RBF 网。RBF 网隐层采用标准 Gaussian 径向基函数，输出层采用线性激活函数，即 $f(u)=u$。其它参数设定如下：

1、 聚类方法：数据中心和扩展常数用聚类方法得到，输出权值和偏移采用广义逆方法求解。隐节点数（即聚类数）取 10，隐节点重叠系数为 1，初始聚类中心取前 10 个训练样本。

2、 梯度法：数据中心、扩展常数和输出权值均用梯度法求解，它们的学习率均为 0.001。其中隐节点数取 10，初始输出权值取 $[-0.1,0.1]$ 内随机值，初始数据中心取 $[-4.0,4.0]$ 内随机值，初始扩展常数取 $[0.1,0.3]$ 内随机值，目标误差 0.9，最大训练次数 5000。

3、 正交最小二乘算法：扩展常数取固定值 0.6，目标误差 0.9，隐节点数由算法自动决定，输出权值用梯度法求解。

上面三种方法用于 Hermit 多项式逼近的 Matlab 程序见附录 C、D、E。

下图 4.11 为对某批训练样本，三种方法对目标函数的逼近曲线，其中三条实线（未作区分）分别为三种方法得到的 RBF 网的输入输出曲线，“ ＋ ”为样本。由图可见，三条曲线是非常接近的，说明三种方法都能较好地完成对目标函数的逼近。但事实上，聚类方法和 OLS 方法学习速度要快得多，梯度方法速度较慢，与 BP 算法类似。

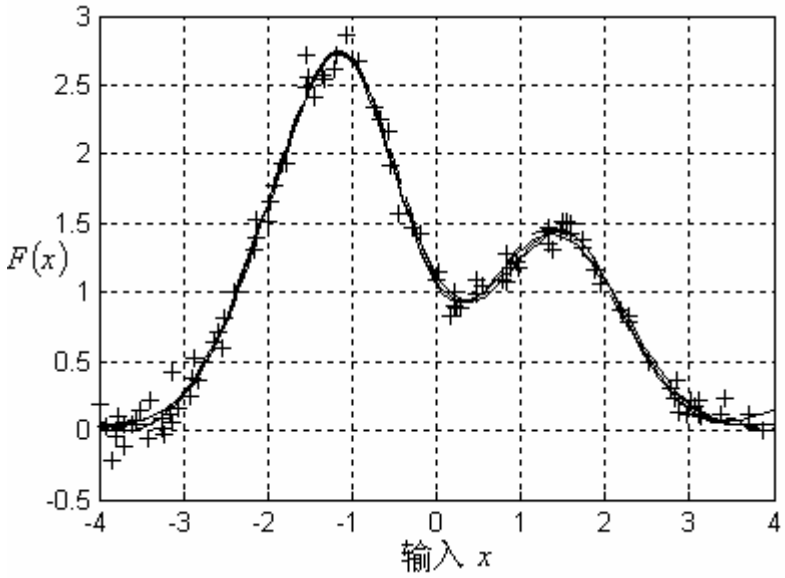

图 4.11　三种 RBF 网学习算法对 Hermit 多项式的逼近

4.7 讨论

1) 径向基函数神经网络的特点

(1) 单隐层。而 MLP 或 BP 网可能多隐层，且输入层到隐层的连接权值固定为 1。
(2) 用于函数逼近时，隐节点为非线性激活函数，输出节点线性函数。隐节点确定后，输出权值可通过解线性方程组得到。
(3) 具有“局部映射”特性。是一种有局部响应特性的神经网络，如果神经网络有输出，必定激活了一个或多个隐节点。局部映射特性在某些情况下可能特别有用。因为，在某些实际应用中，我们宁愿得不到输出，也不愿得到一个可能有较大误差的输出。
(4) RBF 网隐节点的非线性变换作用与 BP 网类似，都是把线性不可分问题转化为线性可分问题。

2) 其它问题

(1) RBF 网的逼近能力
有万能逼近定理：与 BP 网络类似，只要隐节点数足够多，RBF 网能以任意精度逼近紧集上的连续函数[HaKe1990，PaSa1991，GoPo1990]。
(2) 如何确定 RBF 网的隐层节点数
通常情况下，应该设计满足精度要求的最小结构的神经网络[NiGi1996，Moody1992]，以保证神经网络的泛化能力。
(3) OLS 算法并不一定能设计出具有最小结构的网络
尽管 Chen 等声称 OLS 算法能设计最小结构的 RBF 网，但 Sherstinsky 和 Picard 给出了反例[ShPi1996]。OLS 方法设计的 RBF 网确实不一定是最小结构的，但在大多数情况下，OLS 方法确实能给出一个比较精简的神经网络结构。如果神经网络的规模很大（目标函数或概念比较复杂），那么 OLS 方法设计的 RBF 网可能会影响泛化能力。
(4) 如何确定学习精度
与待解决的问题有关。通常的做法是交叉测试（Cross-Validation），以测试误差最小时的训练误差为学习精度。
(5) 如何确定基函数的扩展常数
一种方法是取固定值，但值的大小要通过凑试方法（Try and Error）确定；另一种方法是聚类。聚类方法能根据各聚类中心之间的距离确定各隐节点的扩展常数，缺点是确定数据中心时只用到了样本输入信息，而没有用到样本输出信息或误差信息；另外聚类方法也无法确定聚类的数目。