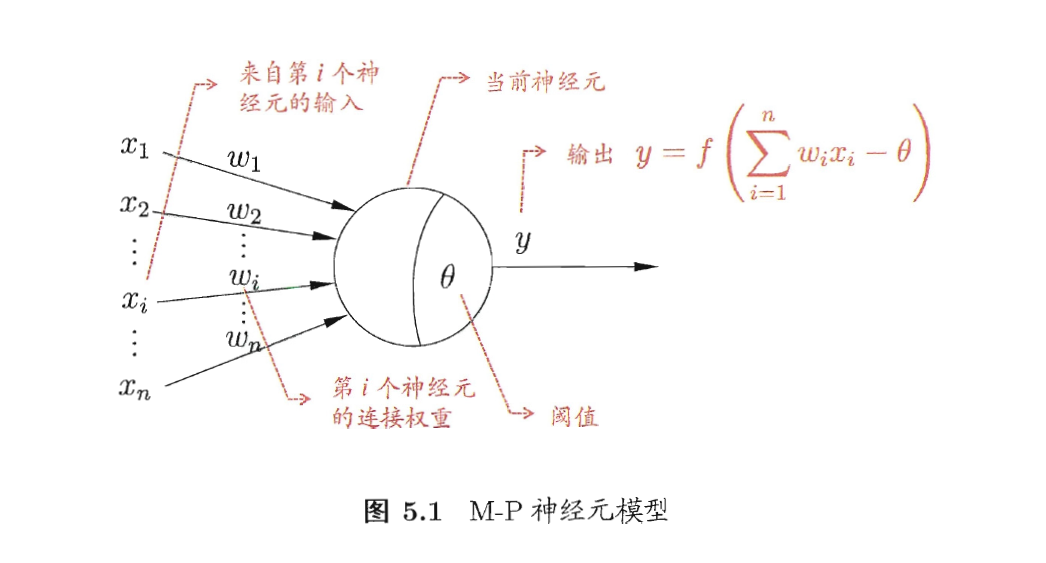
# 5.1神经元模型

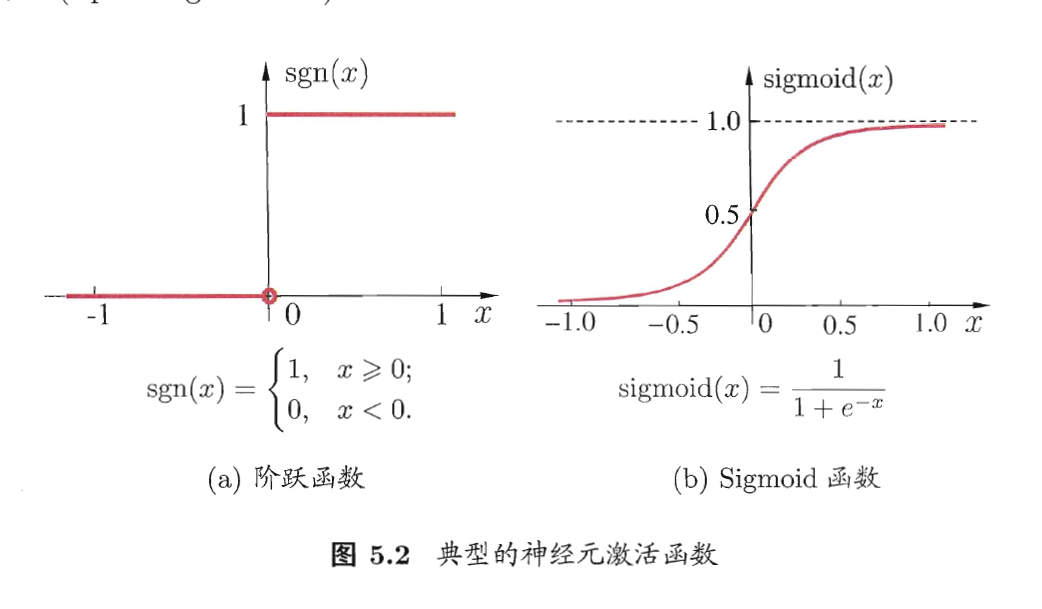
神经网络的定义：由具有适应性的简单单元组成的广泛并行互联的网络，它的组织能够模拟生物神经系统对真实世界物体所作出的交互反应。

**神经元模型**：所谓简单单元，是神经网络中最基本的成分。

**M-P神经元模型**：该模型中神经元接收到n个其他神经元传递的输入信号，这些信号通过**带权重的连接（connection）**进行传递，神经元收到的**总输入值**将与神经元的**阈值**进行比较，然后通过“**激活函数**”处理以产生神经元输出。

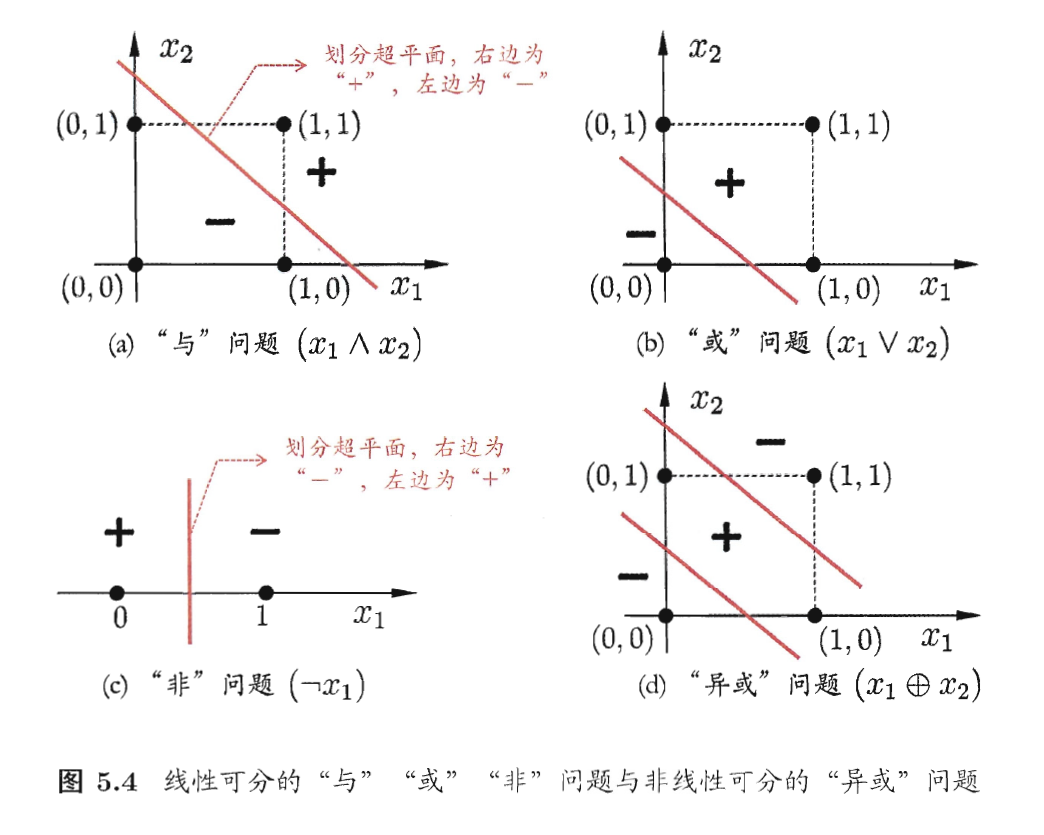


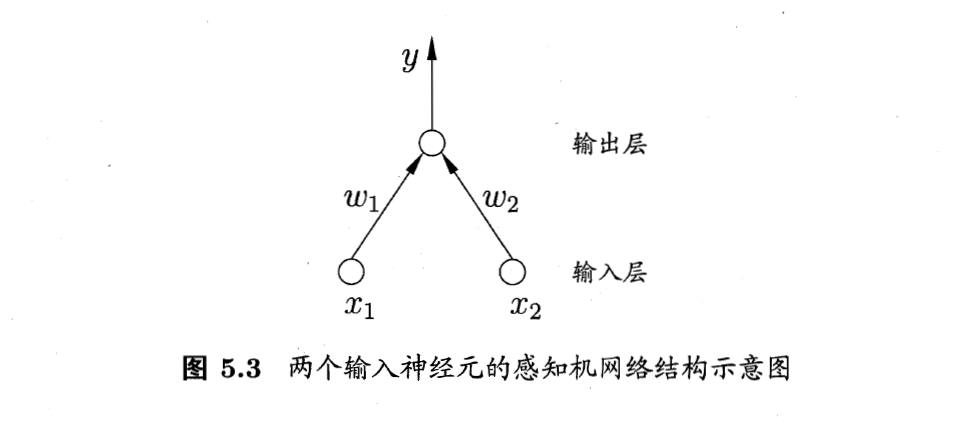
**激活函数**：理想的激活函数如下图左所示，但是由于不连续，不光滑的性质，实际常用Sigmoid函数作为激活函数。



# 5.2感知机与多层网络

**感知机**由**两层神经网络**组成，结构如图所示，下层是输入层，上层是输出层，输出层是M-P神经元，亦称“阈值逻辑单元”。感知机仅有输出层神经元有激活函数处理，即只有一层**功能神经元**，因此学习能力非常有限。感知机能够解决的是**线性可分**的问题，如与、或、非问题，存在一个线性超平面来完成划分，如图5.4前三张所示。面对线性不可分问题，感知机会发生震荡，不能收敛，常见的线性不可分问题如图5.4第四张所示。





感知机工作举例：实现逻辑或运算。

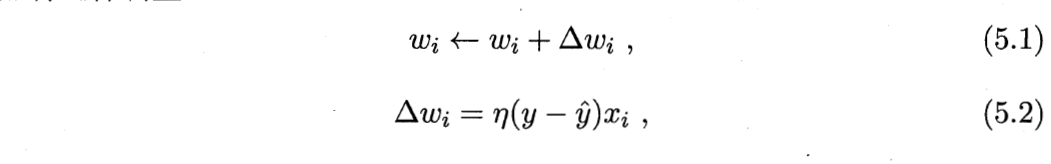


其中0.5是阈值的取值，w为权重取值。

一般情况下，**阈值**和**权重**需要通过**在训练集上的学习**获得。

**Tips：**实施过程中，阈值可以看作是**固定输入为-1的“哑结点”**，对应一定的权重，这样可以将对权重和阈值的学习**统一为对权重的学习**。

感知机在学习中权重如何调整？

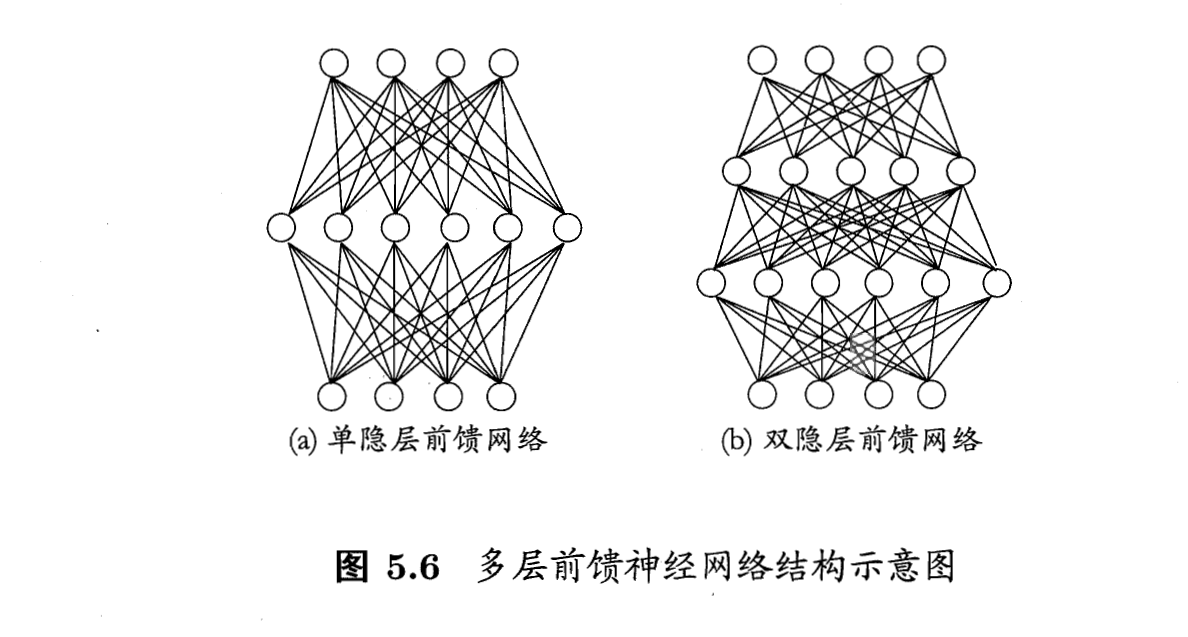


xi是x对应输入第i个输入神经元的分量，η是学习率，y^是感知机的输出。

**Q：解决在线性可分问题上不收敛的问题？**

A：使用**多层**功能神经元。在输出层和输入层之间的神经元被称为**隐层**或**隐含层**。隐含层和输出层神经元都是拥有激活函数的神经元。

**多层前馈神经网络**



由上述问题引入包含隐含层的神经网络，称为多层前馈神经网络，神经元之间不存在同层连接，也不存在跨层连接。输入层神经元仅接收输入，隐含层和输出层包含功能神经元，能够进行函数处理。**\*\*上图5.6a常被称为“两层网络“，实则有三层神经元\*\***

神经网络的学习过程就是根据训练数据来调整神经元之间的连接权和每个功能神经元的阈值。换言之，神经网络学到的东西，蕴含在连接权和阈值中。

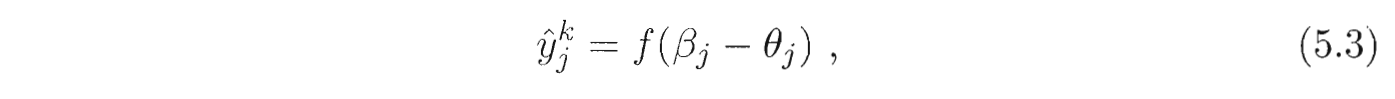
# 5.3误差逆传播算法

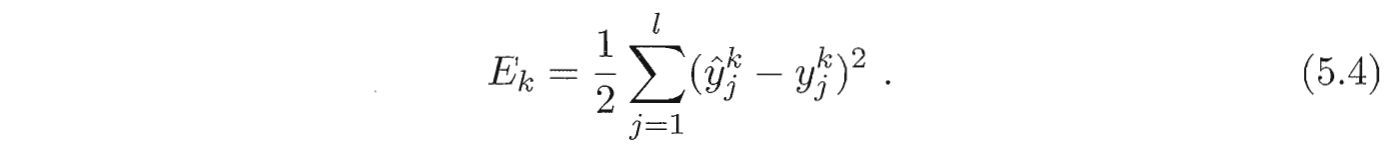
多层网络需要比感知机**更为复杂**的学习算法，**误差逆传播算法（BP）**是其中的最杰出代表。通常说的”**BP网络**“指代的是用BP算法训练的多层前馈神经网络。

参数每次迭代更新的计算方法基于**梯度下降策略**。

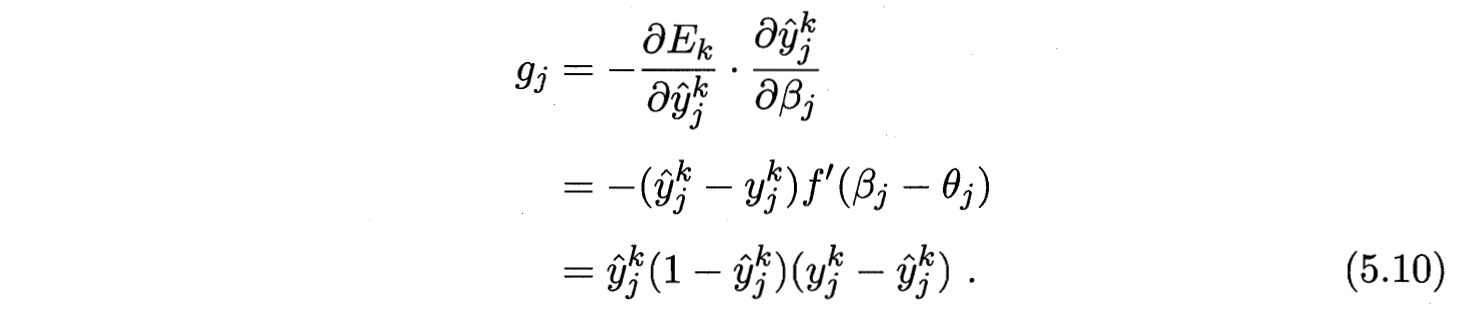
首先定义一个一个BP网络如下图所示：

针对某训练例(xk,yk),神经网络**输出**计算公式以及输出的**均方误差**计算公式如下：

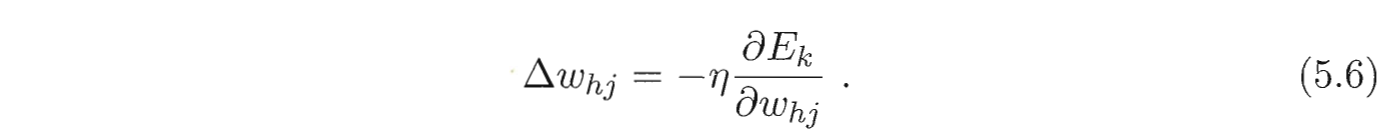




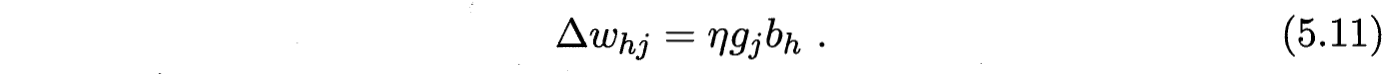
根据以上两式可推得：



以隐层到输出层的**连接权重w**为例，变化量计算公式如下：

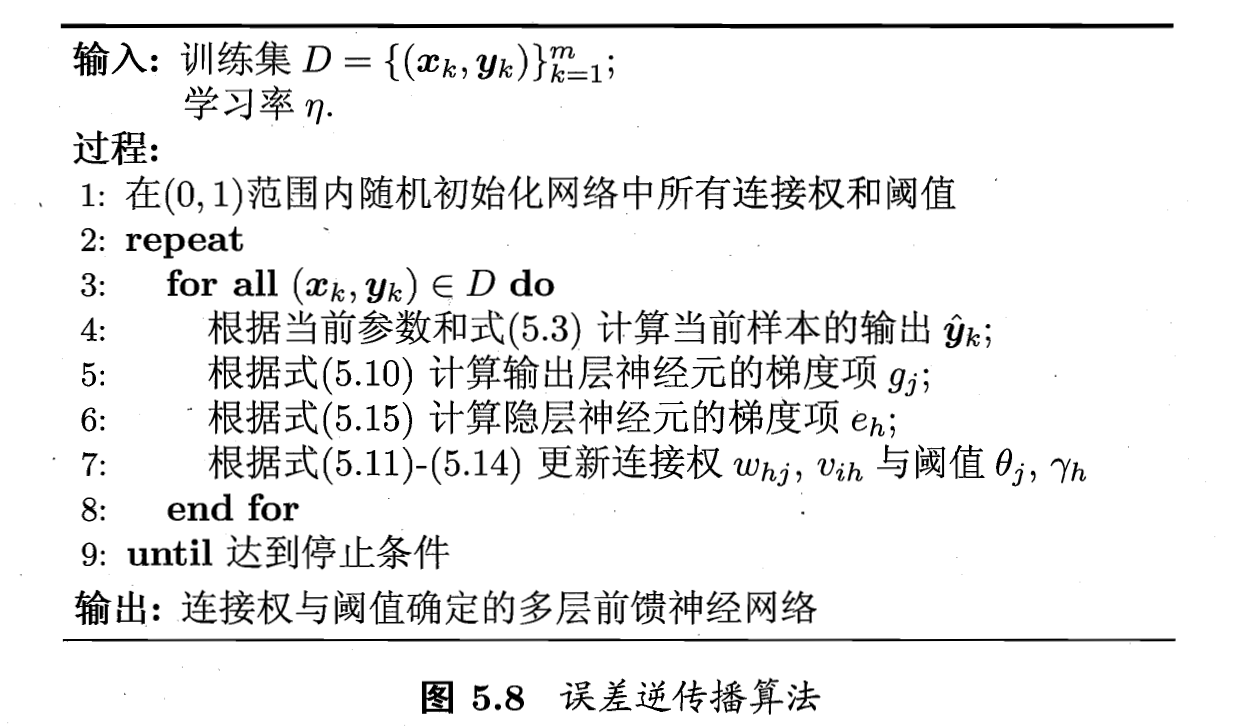


通过变形得到更新公式如下：



阈值和其余权重的变化类似。

BP算法流程伪代码如下：



示例输入**输入神经元**后逐层将信号传递到**输出神经元**产生结果。之后计算**输出的误差**（4，5行），再将误差**逆向传播**到**隐层神经元**（6行），最后根据隐层神经元的误差来对**连接权**和**阈值**进行调整（7行）。上述迭代过程**循环进行**，直到达到停止条件，如误差可以接受。

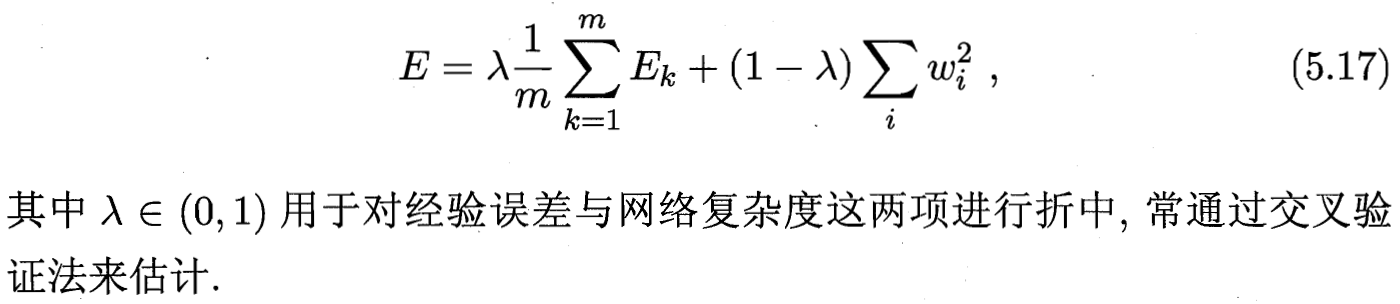
上述算法被称为**标准BP算法**，该算法**每次更新**针对的是**某一个示例**，更新之间可能存在**抵消现象**，并且实现的并**不是**训练集上的**累计误差最小**，由此提出**累积BP算法**。

累积BP算法读取**整个训练集**一遍后才开始更新参数，**前期**误差**下降快**，但是**后期**误差**下降缓慢**，与标准BP算法各有优劣。

**可以证明，一个包含足够多隐层神经元的多层前馈网络可以以任意精度逼近任意复杂的连续函数**，由于学习能力强大，BP神经网络经常遭遇**过拟合问题**。解决策略如下：

1）早停：当训练集误差降低而验证集误差升高时停止训练，选择效果最好的参数。

2）正则化：修改误差目标函数，添加项描述网络的复杂程度（如连接权与阈值的平方和）



# 5.4全局最小与局部极小

如何尽可能跳出局部极小找到全局最小？

1）以**多组不同的参数值**初始化多个神经网络。

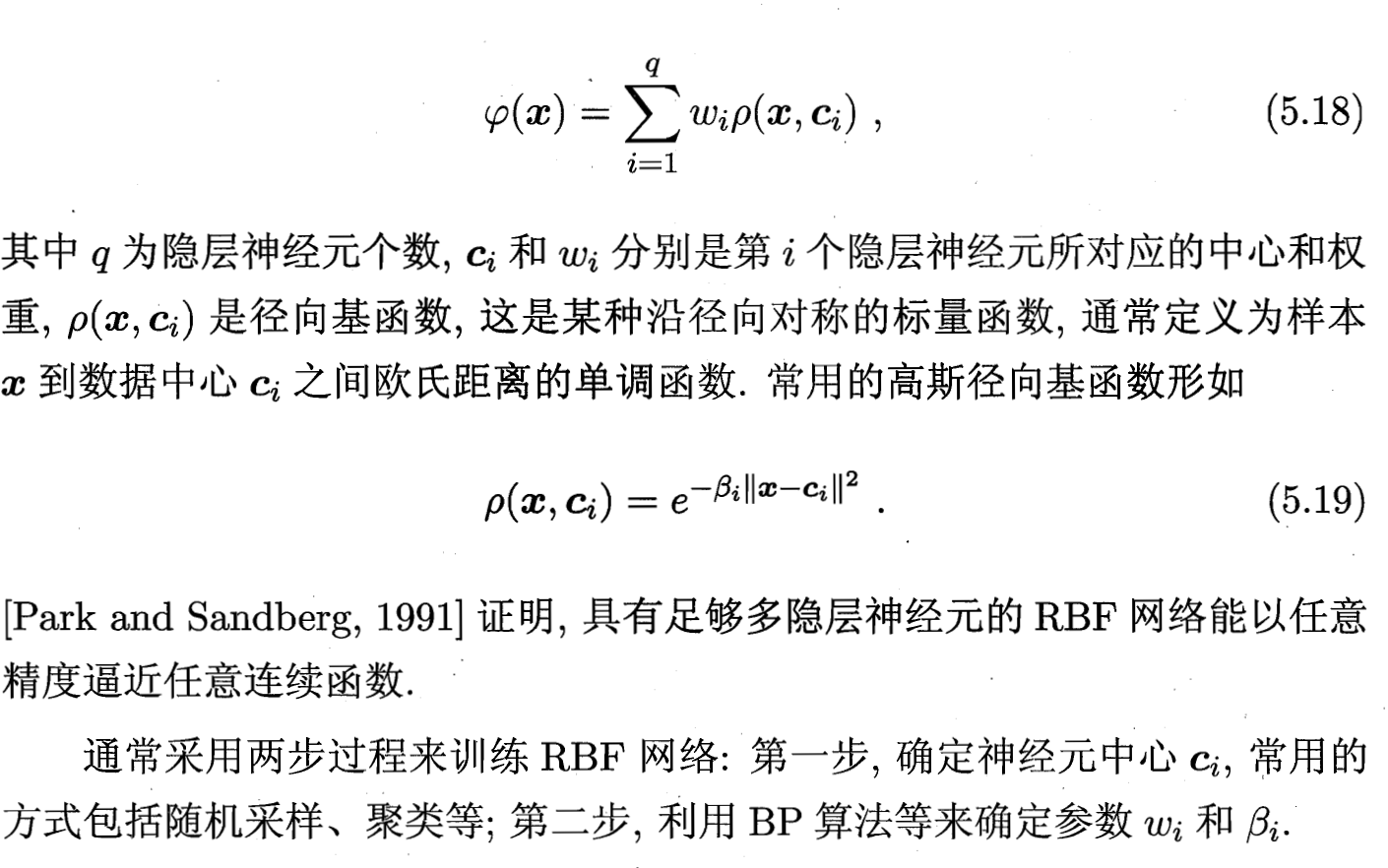
2）**模拟退火技术**。在每一步都有一定概率选择比最优差的结果，接受次优解的概率随时间的推移而降低，保证模型趋于稳定。

3）**随机梯度下降**。计算梯度时加入随机因素，导致局部最小点梯度可能不为零，帮助跳出局部最小。

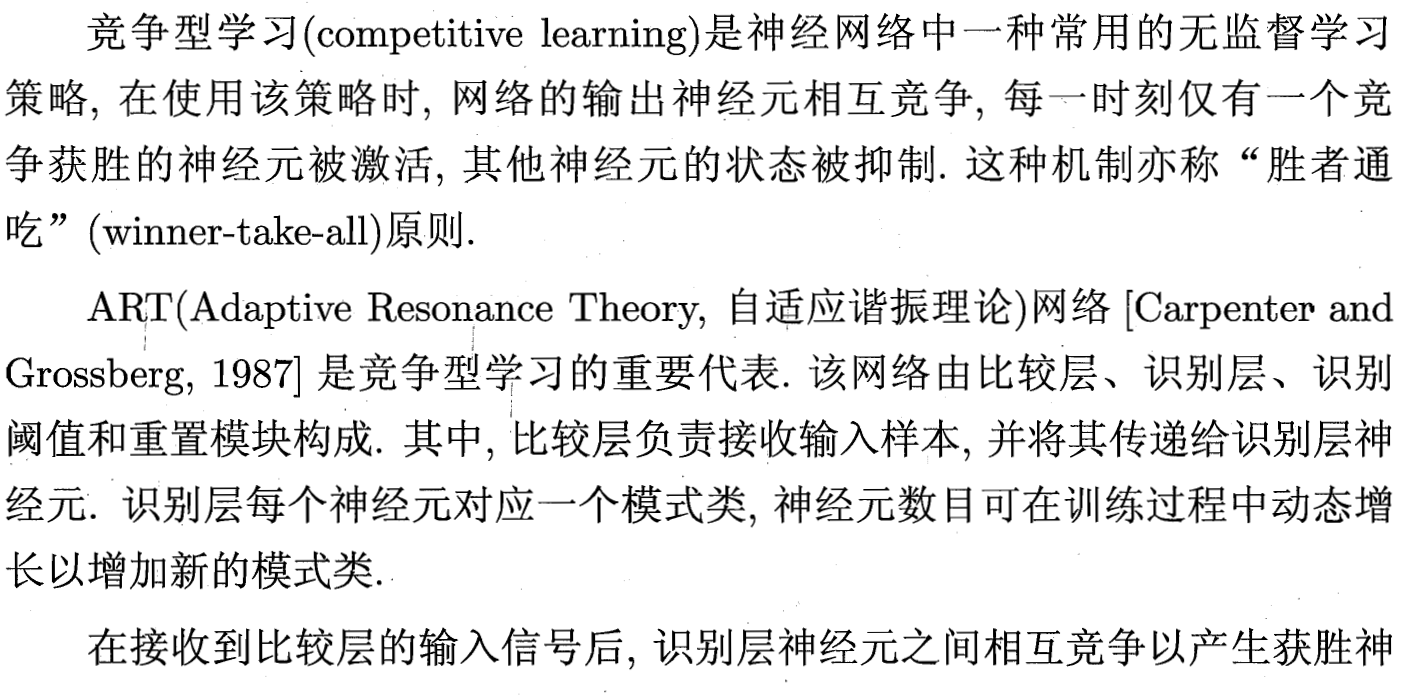
# 5.5其他常见的神经网络

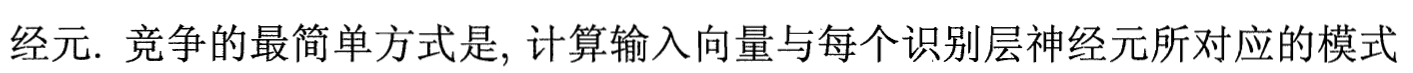
## 5.5.1 RBF网络

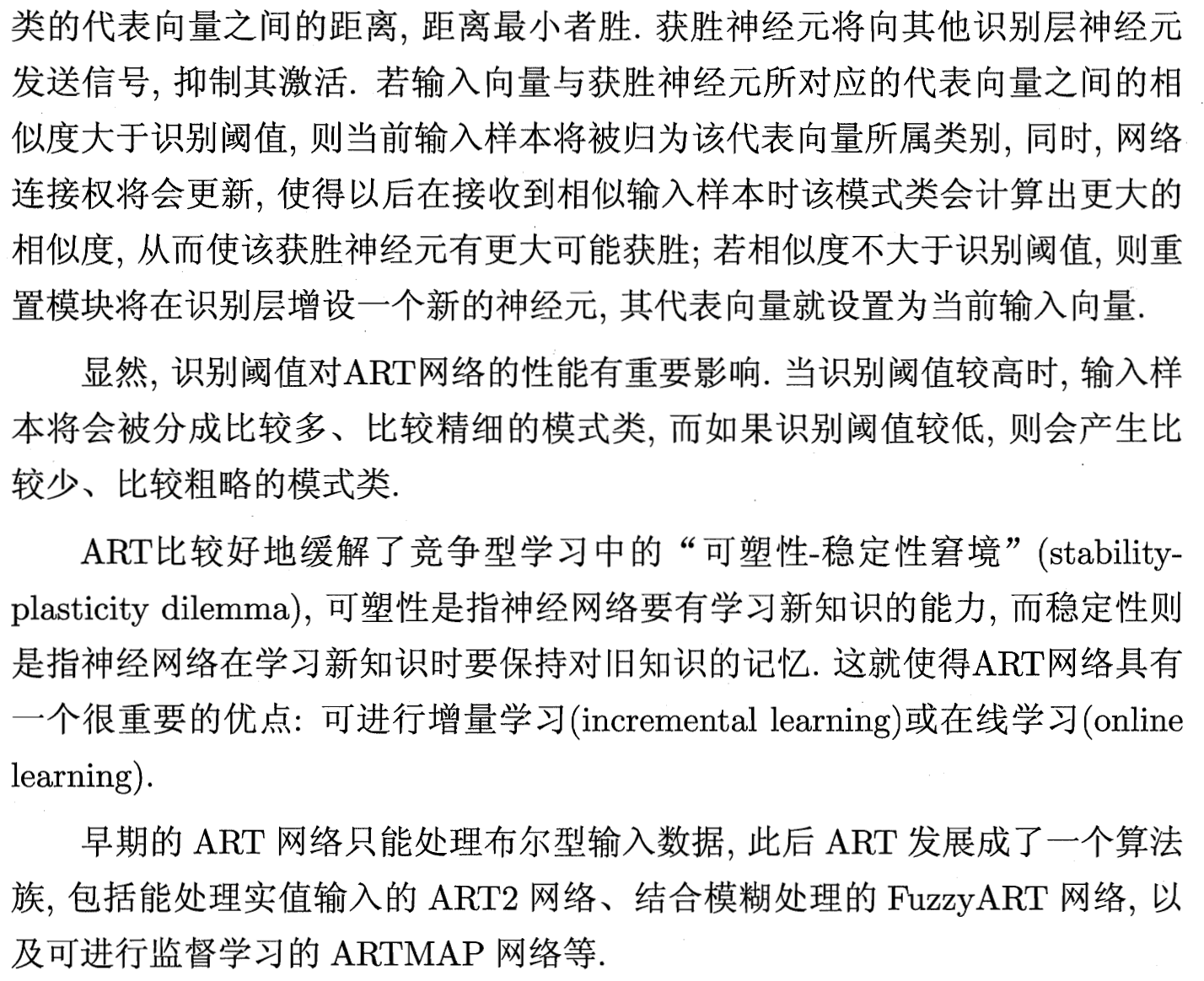
一种单隐层前馈神经网络，使用**径向基函数**作为**隐层神经元激活函数**，而**输出层**则是对**隐层神经元**输出的**线性组合**。RBF网络可以表示为：



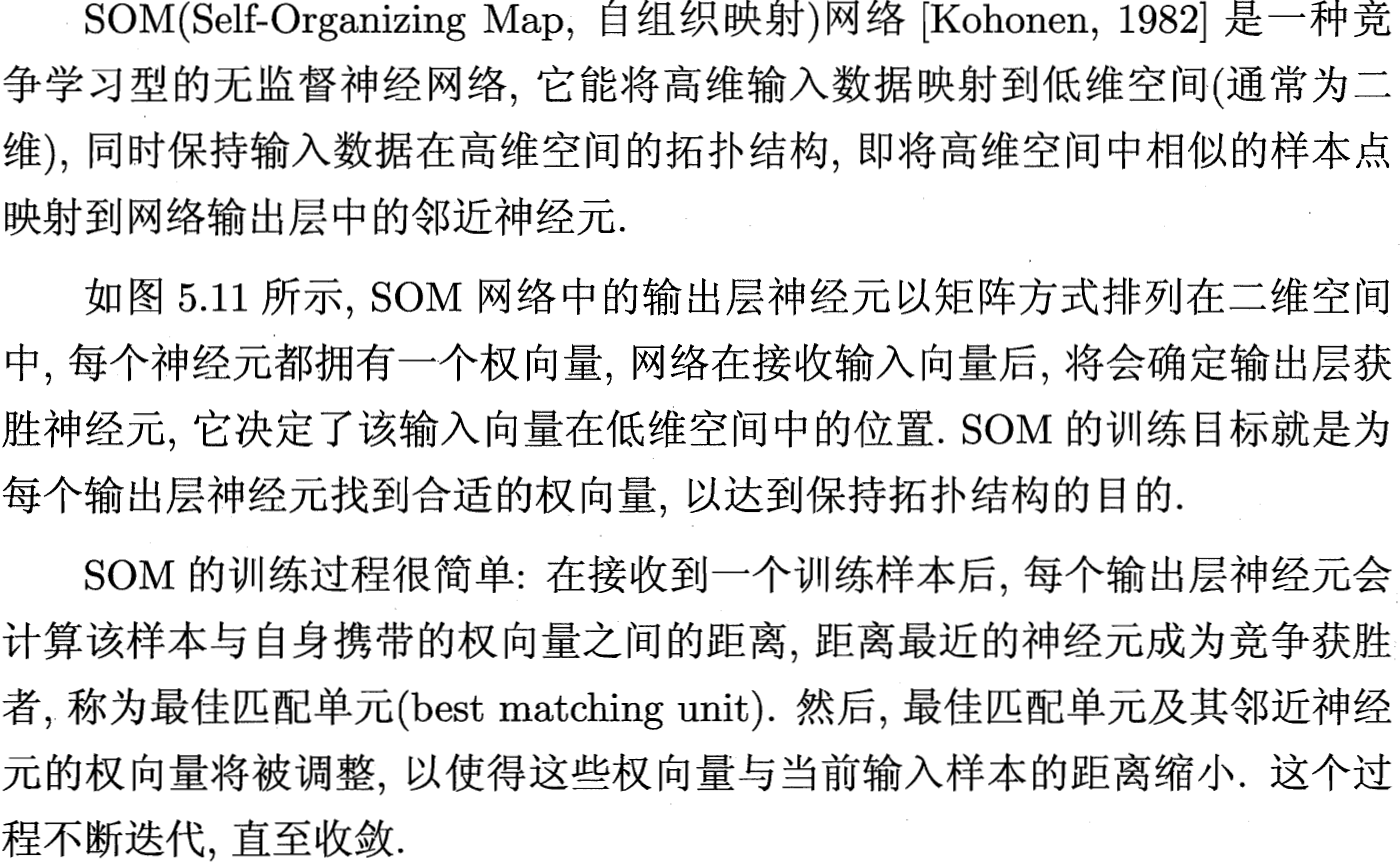
## 5.5.2 ART网络

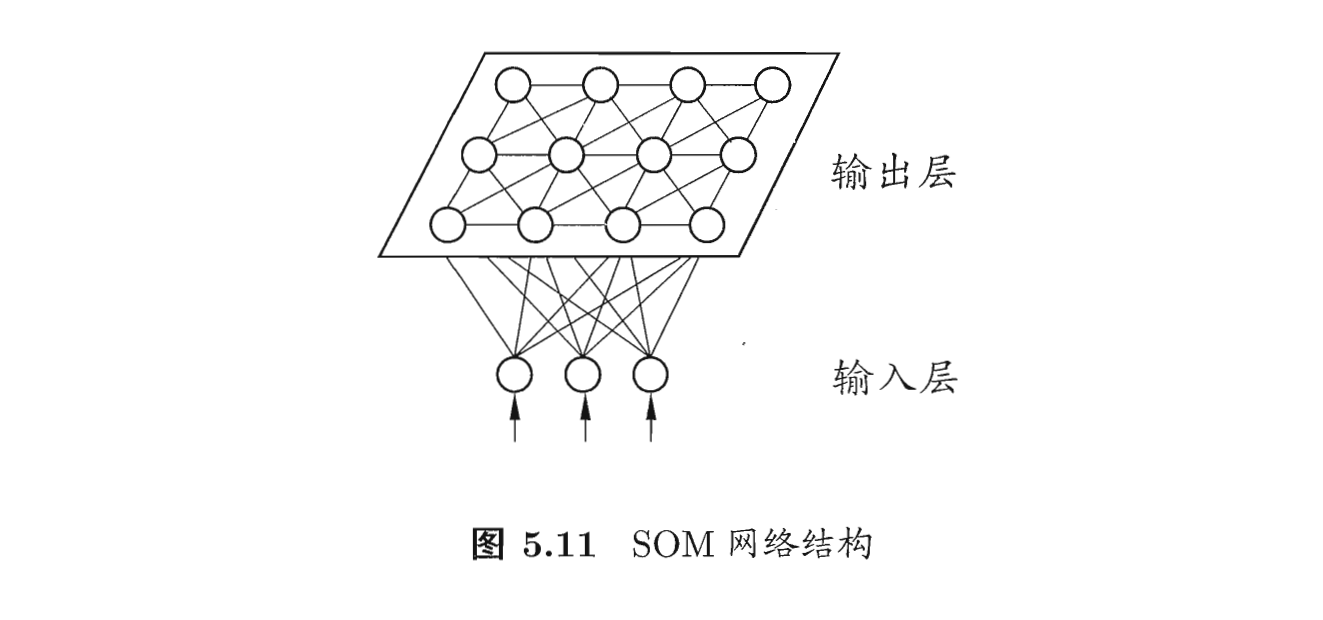




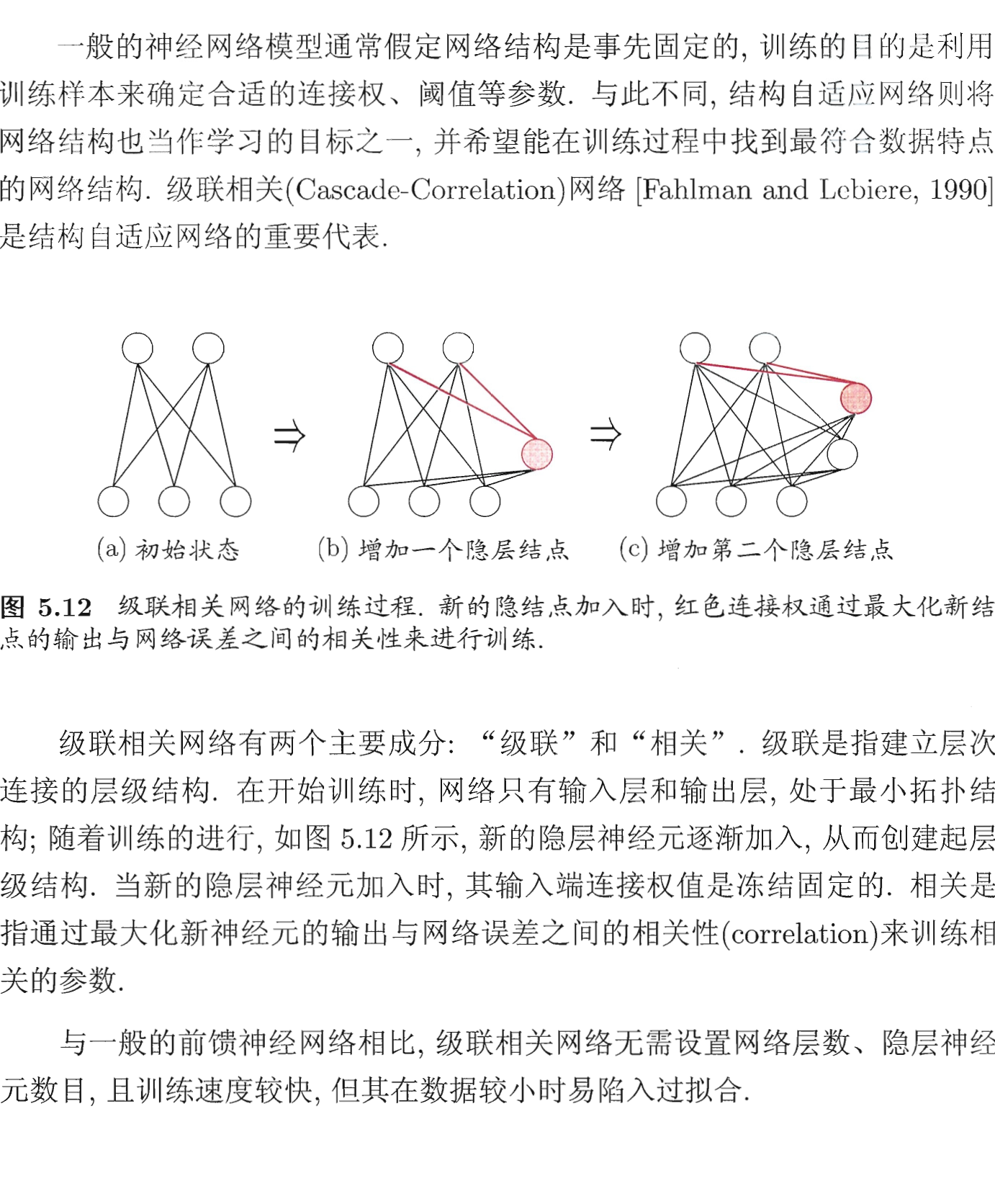


## 5.5.3 SOM网络

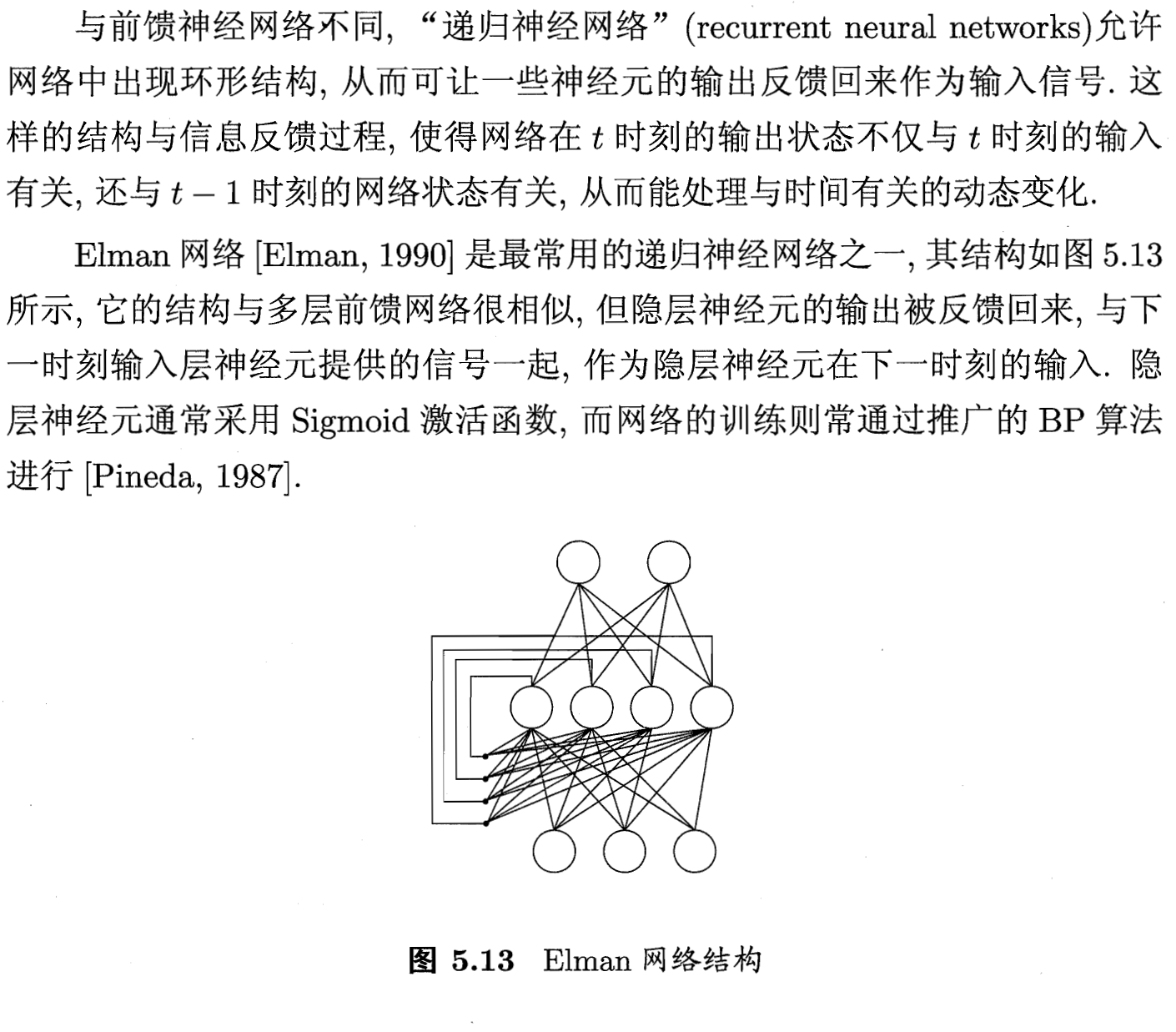




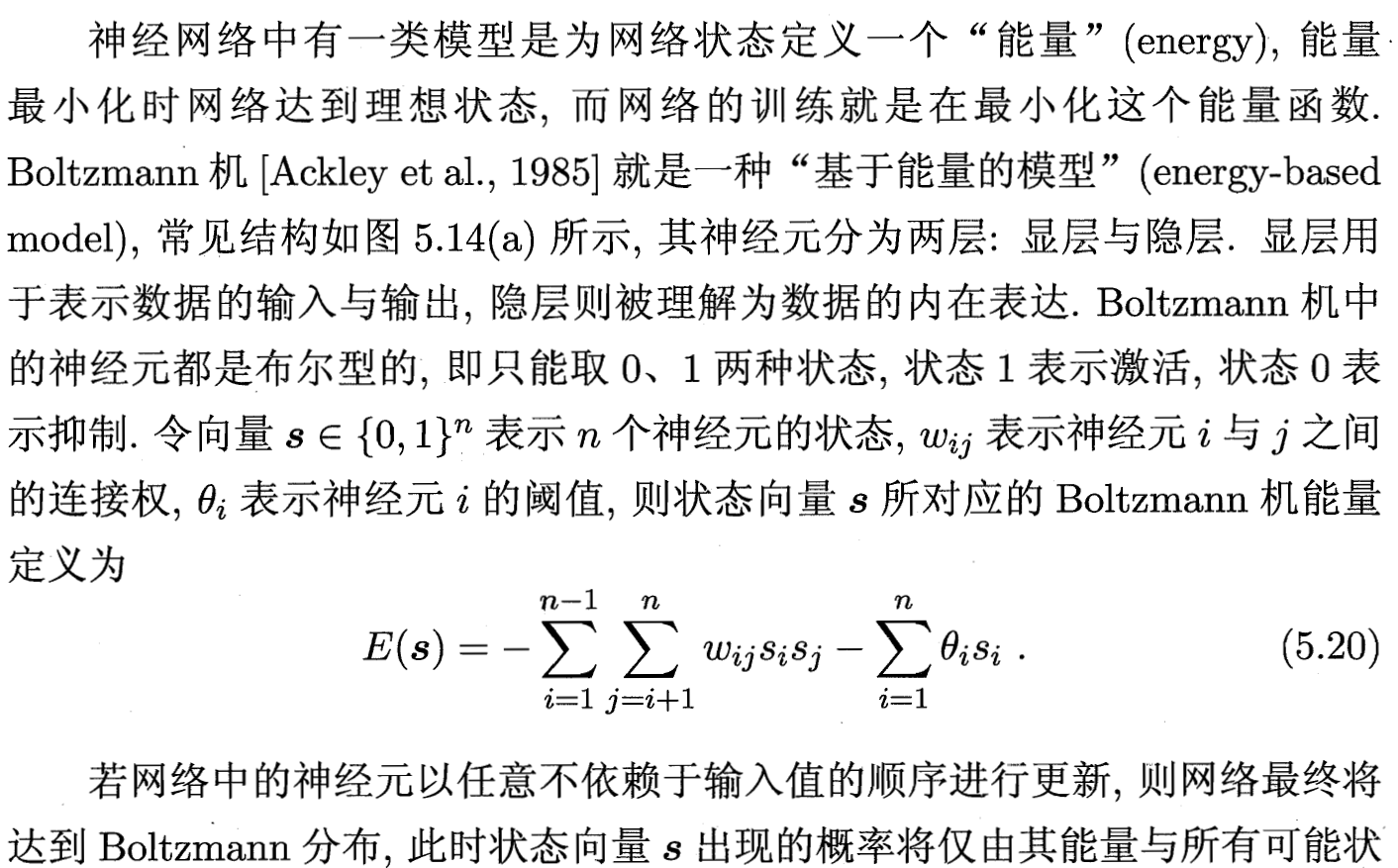
## 5.5.4级联相关网络

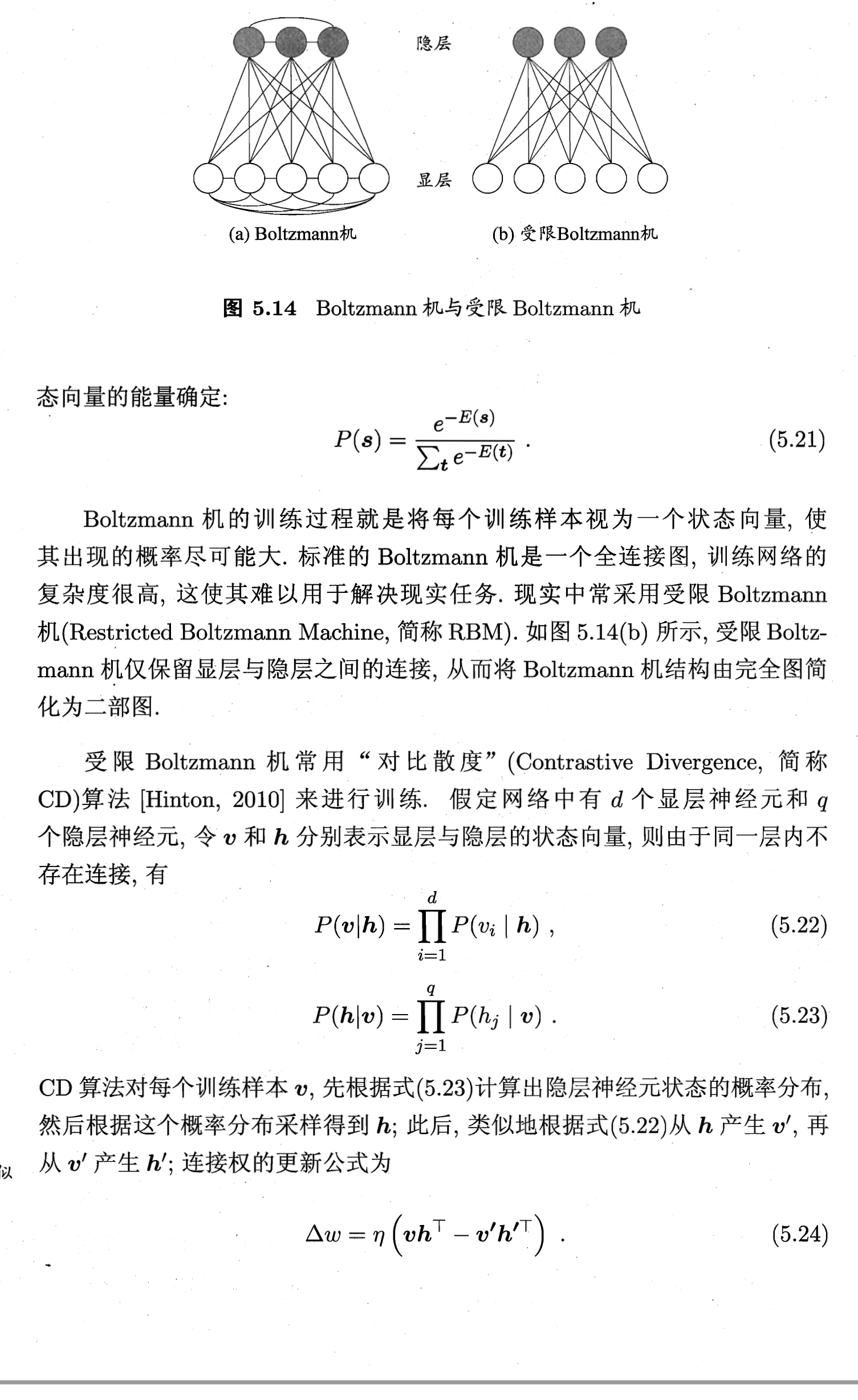


## 5.5.5 Elman网络



## 5.5.6 Boltzmann机





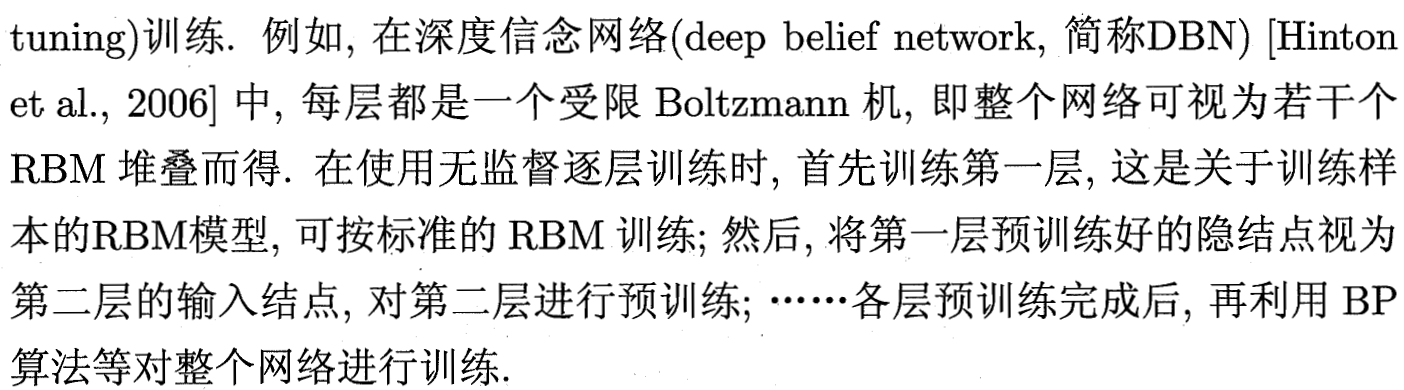
**可以看出也是一种递归神经网络。**

# 5.6深度学习

对着计算能力增强，训练数据量大幅增加缓解了过拟合风险，人们更加青睐结构复杂的多隐层神经网络。然而，前述的**BP算法等经典算法**在多隐层中逆向传播中往往会**发散**而不是收敛到一个稳定的结果，因此不能直接用来训练。

**无监督逐层训练**是多隐层网络训练的有效手段，其基本思想为**每次训练一层隐结点**。

训练时将上一层隐结点的输出作为输入，而本层隐结点的输出作为下一层隐结点的输入，此过程为**“预训练“**过程。预训练完成后再对整个网络进行”微调训练“。相当于先找到**局部较优的参数**，再在此基础上寻找**全局最优的参数**。



**预训练+微调**的做法可以视为一种节省开销的方案，另一种方案是**“权共享“**。即每一组神经元使用相同的连接权，从而大幅减少需要训练的参数数目。（不是同层的神经元权值一样，而是不同层使用同一套权值分布）