6.2 神经网络基本原理

6.2.1 人工神经元(Artificial Neuron)模型

- 生物的大脑是由许多神经细胞组成，同样，模拟大脑的人工神经网络ANN是由许多叫做人工神经细胞（或人工神经元）的细小结构模块组成。人工神经细胞就像真实神经细胞的一个简化版。

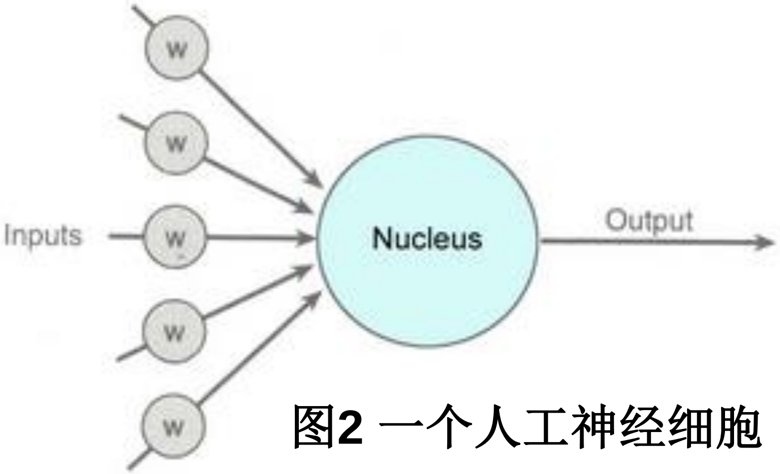
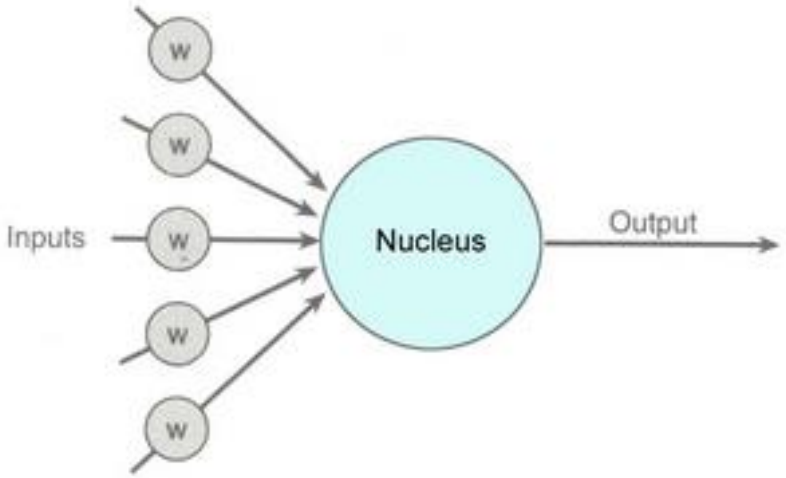


图2 一个人工神经细胞

- 图中，左边几个灰底圆中所标了字母w代表浮点数，称为权重。进入人工神经细胞的每一个input(输入)都与一个权重w相联系，这些权重将决定神经网络的整体活跃性。可以设想这些权重被设置成-1和1之间的一个随机小数。因为权重可正可负，故能对与它关联的输入施加不同的影响，如果权重为正，就会有激发作用，权重为负，则会有抑制作用。

6.2.1 人工神经元(Artificial Neuron)模型

- 当输入信号进入神经细胞时，将与它们对应的权重相乘，作为图中大圆的输入。大圆的‘核’是一个函数，叫激励函数，它把所有这些经过权重调整后的输入全部加起来，形成单个的激励值。
- 激励值也是一浮点数，且同样可正可负。然后，再根据激励值来产生函数的输出也即神经细胞的输出：如果激励值超过某个阀值（假设阀值为1.0），就会产生一个值为1的信号输出；如果激励值小于阀值1.0，则输出一个0。这是人工神经细胞激励函数的一种最简单的类型。在这里，从激励值产生输出值是一个阶跃函数。



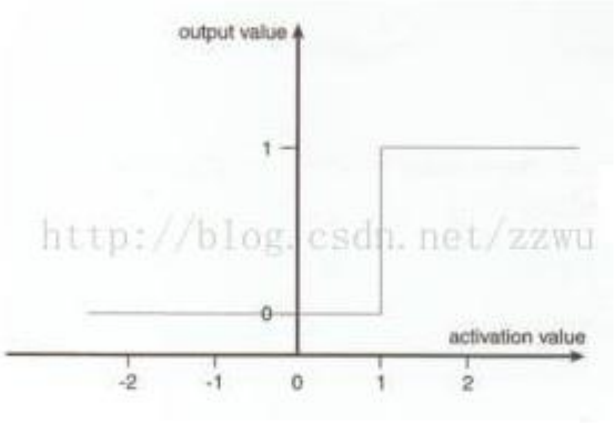
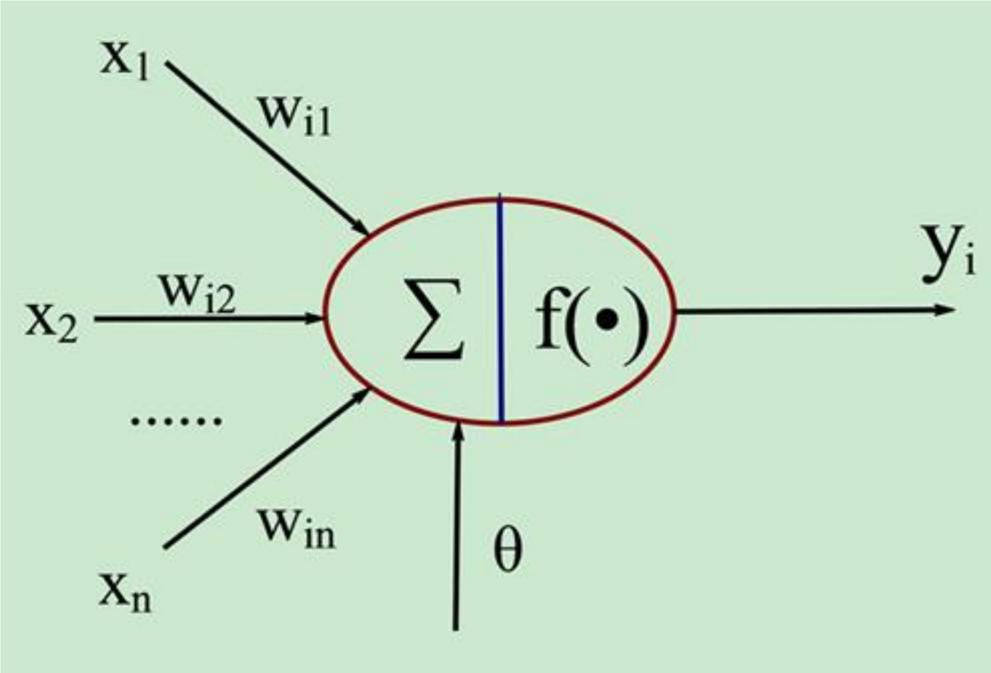


图3 阶跃激励函数

6.2.1 人工神经元(Artificial Neuron)模型

- 一个人工神经细胞可以有任意n个输入，n代表总数。
- 激励值就是所有输入与它们对应权重的之乘积之总和，如果激励值超过了阈值，神经细胞就输出1；如果激活小于阈值，则神经细胞的输出为0。这和一个生物神经细胞的兴奋和抑制是等价的。



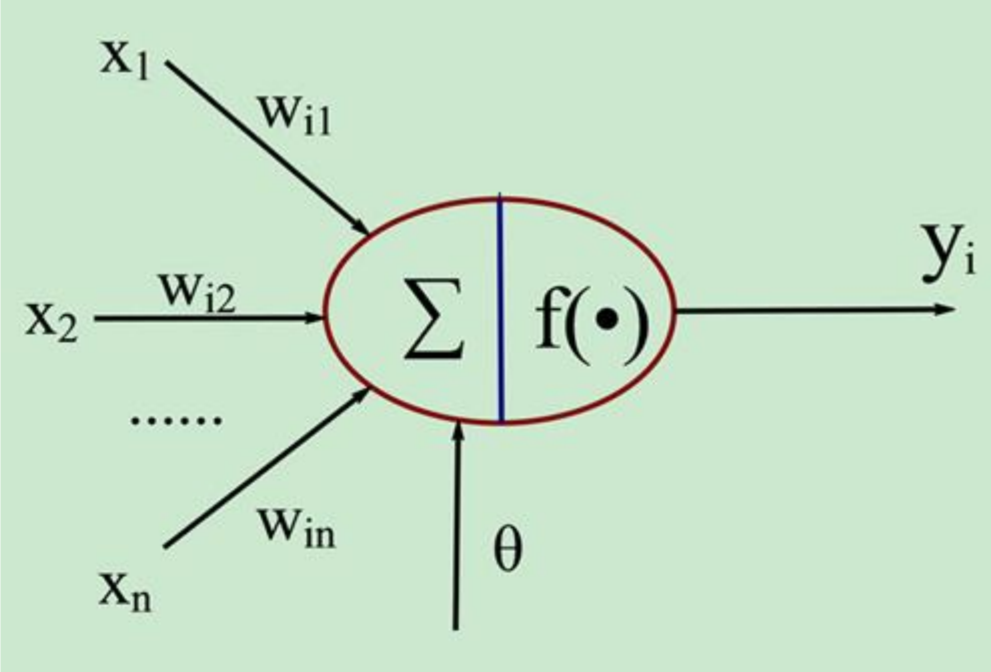
净激活

$$\mathrm{net}_i = \sum_{j=1}^{n} w_{ij} x_j - \theta$$

$$y_i = \mathrm{f}(\mathrm{net}_i)$$

6.2.1 人工神经元(Artificial Neuron)模型

- 图中的这种“阈值加权和”的神经元模型称为M-P模型 (McCulloch-Pitts Model)，也称为神经网络的一个处理单元(PE, Processing Element)。



$$net_i = \sum_{j=1}^{n} w_{ij} x_j - \theta$$

$$y_i = f(net_i)$$

6.2.1 人工神经元(Artificial Neuron)模型

- 假设一个神经细胞有5个输入，他们的权重w都初始化成正负1之间的随机值(-1 < w < 1) 。 表2说明了激励值的求和计算过程。

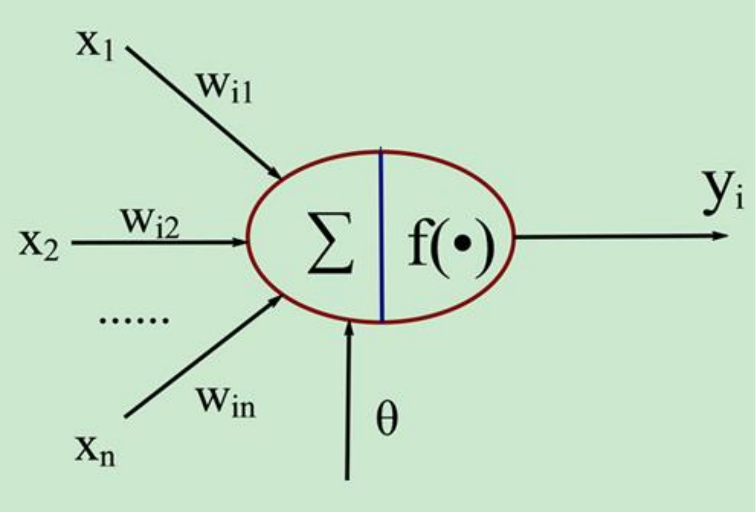
表2 神经细胞激励值的计算

输 入	权 重	输入与权重的乘积	运行后总和
1	0.5	0.5	0.5
0	-0.2	0	0.5
1	-0.3	-0.3	0.2
1	0.9	0.9	1.1
0	0.1	0	1.1

- 如果我们假定激活所需阀值=1，则因激励值1.1 > 激活阀值 1， 所以这个神经细胞将输出1。

6.2.1 人工神经元(Artificial Neuron)模型

- 神经元（Neurodes）
  - 神经网络中的知识，表示为处理单元节点的集合。
- 突触（Synapse）
  - 每个节点与邻近层节点之间的加权连接。
- 权重（weight）值
  - 连接间的权值，相当于神经网络的记忆。
- 激励函数（activation function）
  - 每个节点的输出由一个输出函数计算所得。



6.2.2 激励函数

- 隐层和输出层节点的输入和输出之间具有的函数关系。
- 满足两个要求的多种函数可以作为激励函数
  - 必须输出[0,1]之间的值；
  - 在充分活跃时，将输出一个接近1的值。
- 常见的激励函数
  - 阶跃函数、 Sigmoid函数、准线性函数和双曲正切函数等。
  - 阶跃函数，或称阈值型函数。
  - $$f\left(x\right)=\begin{cases}1, x\geq 0\\0, x<0\end{cases}$$
  - 神经元模型中最简单的一种,经典的M-P模型神经元就属于这一类。

6.2.2 激励函数

- 常见的激励函数
  - Sigmoid函数最常用，也称S形函数。

$$f(x)=\frac{1}{1+e^{-\alpha x}},0< f(x)<1$$

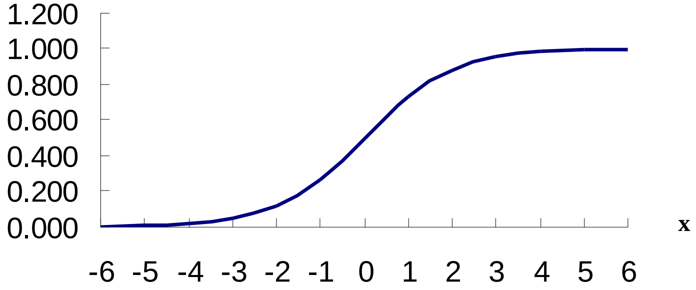
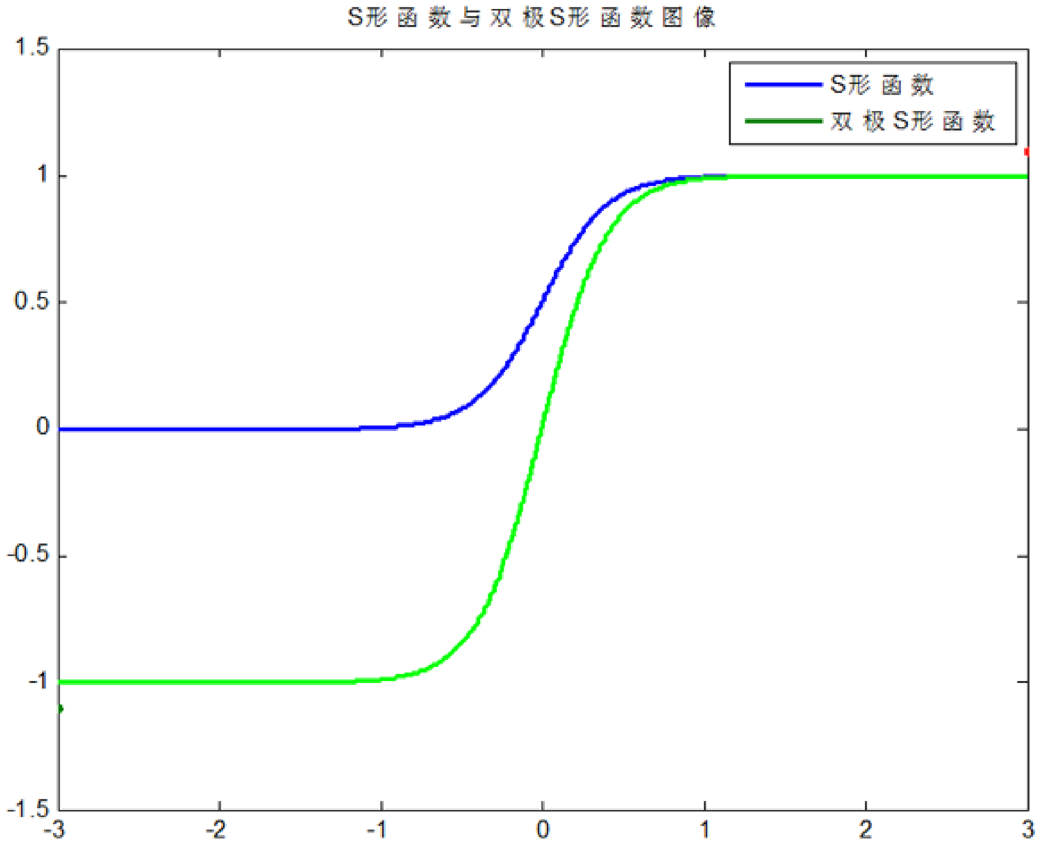


图6.2　S形函数

6.2.2 激励函数

- 常见的激励函数
  - 双极S形函数。

$$f(x)=\frac{2}{1+e^{-\alpha x}}-1, -1< f(x)<1$$



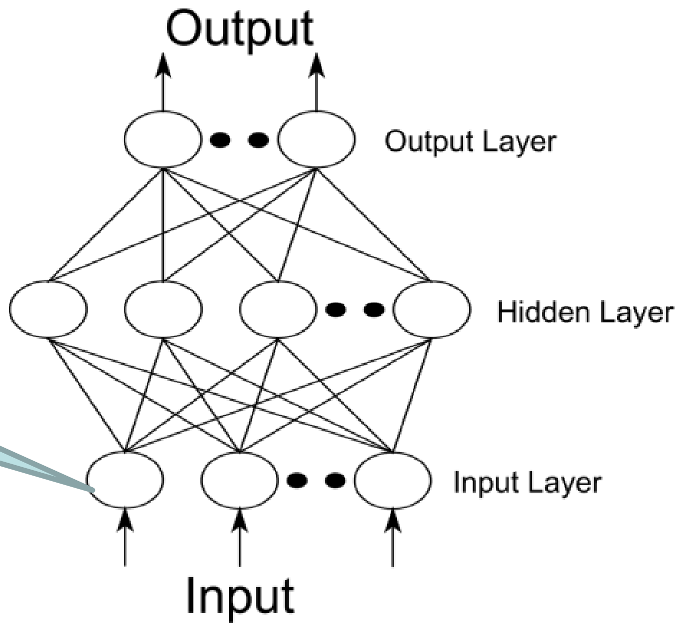
6.2.3 神经网络模型

- 大脑里的生物神经细胞和其他的神经细胞是相互连接在一起的。为了创建一个人工神经网络，人工神经细胞也要以同样方式相互连接在一起。为此可以有许多不同的连接方式。
- 根据网络中神经元的互联方式，常见网络结构主要可以分为 3 类:
- (1) 前馈神经网络　（Feedforward Neural Networks)
- (2) 反馈神经网络　（Feedback Neural Networks)
- (3) 自组织网络 (SOM ,Self-Organizing Neural Networks)
- 其中最容易理解并且也是最广泛地使用的，是前馈神经网络。

6.2.3 神经网络模型

- (1) 前馈神经网络　（Feedforward Neural Networks ）
  - 网络的每一层神经细胞的输出都向前馈送（feed）到了它们的下一层（在图中是画在它的上面的那一层)，直到获得整个网络的输出为止，层间没有向后的反馈信号，因此被称为前馈网络。如BP神经网络。

输入单元不是神经元，因此图中有2层神经元



  - 图 中是一个3层的前馈神经网络，第一层是输入单元，第二层称为隐含层，第三层称为输出层。

6.2.3 神经网络模型

- 单层神经网络由一个输入层（Input layer）和一个输出层（Output layer）组成。

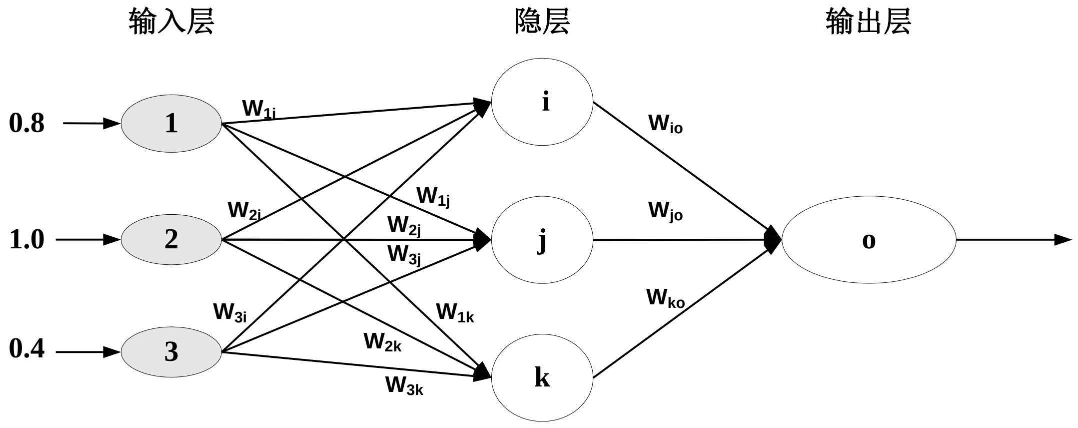
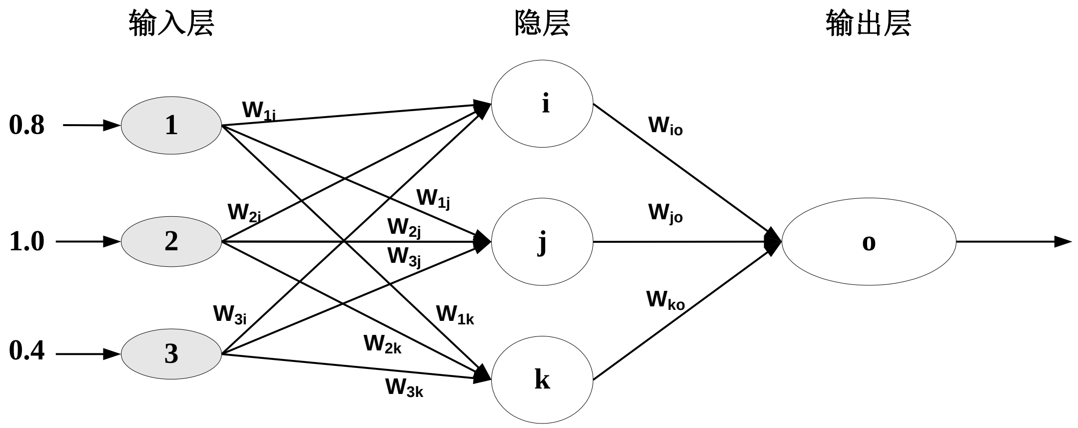


图6.1 全连接前馈神经网络结构

- 输入向量——输入数据，输入层节点数由训练实例的输入属性个数决定。
- 输出向量——输出数据，输出层节点数由问题和应用决定。
- 隐层个数和每个隐层内的节点数可由用户指定，作为前馈网络，可以有任意多个隐层（通常取一层），隐层节点数为输入节点的1.2倍到1.5倍。

6.2.3 神经网络模型

- 神经网络模型的输入、输出关系可描述为

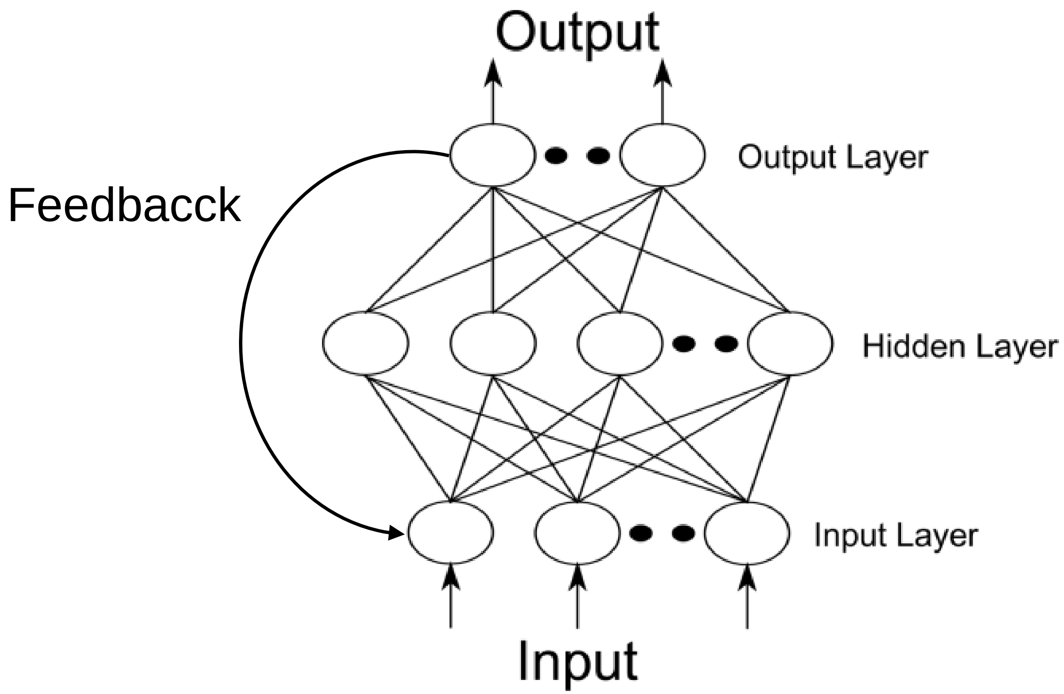


$$
\begin{cases}
u_j = f\left(\sum_{i=1}^{n} w_{ij} x_i - \theta_j\right) \\
y = f\left(u_j\right)
\end{cases}
$$

- 式中 $x_i\,(i=1,2,\cdots,n)$ 是从其它神经元传来的输入信号；θ_j 是该神经元的阈值；w_{ij} 表示从神经元i到神经元j的连接权值；$f(\square)$ 为激活函数。

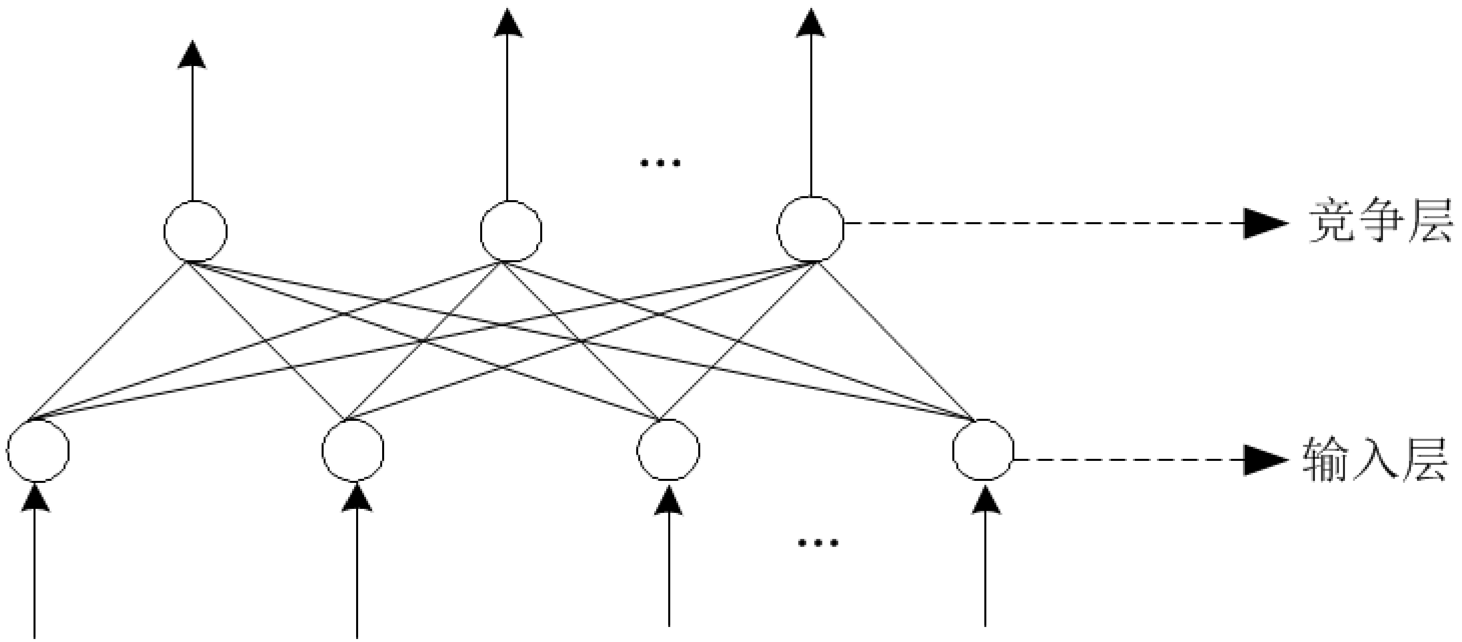
6.2.3 神经网络模型

- (2)反馈神经网络　（Feedback Neural Networks）
  - 反馈型神经网络是一种从输出到输入具有反馈连接的神经网络，其结构比前馈网络要复杂得多。
  - 典型的反馈型神经网络有：Elman网络和Hopfield网络。



6.2.3 神经网络模型

- (3) 自组织网络 (SOM ,Self-Organizing Neural Networks)
  - 自组织神经网络是一种无导师学习网络。它通过自动寻找样本中的内在规律和本质属性，自组织、自适应地改变网络参数与结构。

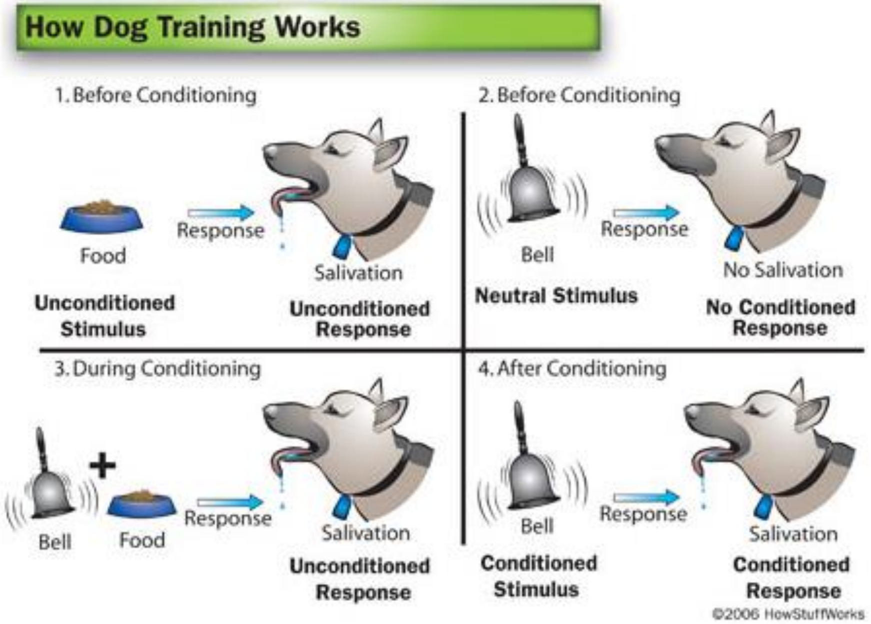


6.2.4 神经网络工作方式

- 神经网络运作过程分为学习和工作两种状态。
- (1)神经网络的学习状态
  - 网络的学习主要是指使用学习算法来调整神经元间的联接权，使得网络输出更符合实际。
  - 学习算法分为有导师学习(Supervised Learning)与无导师学习(Unsupervised Learning)两类。
  - 有导师学习算法将一组训练集 (training set)送入网络，根据网络的实际输出与期望输出间的差别来调整连接权。如BP算法。
  - 无导师学习抽取样本集合中蕴含的统计特性，并以神经元之间的联接权的形式存于网络中。Hebb学习律是一种经典的无导师学习算法。
- (2) 神经网络的工作状态
  - 神经元间的连接权不变，神经网络作为分类器、预测器等使用。

6.2.4 神经网络工作方式

- 无导师学习算法：Hebb学习率
  - Hebb算法核心思想是，当两个神经元同时处于激发状态时两者间的连接权会被加强，否则被减弱。
  - 要理解Hebb算法，有必要介绍一下条件反射实验。巴甫洛夫的条件反射实验：每次给狗喂食前都先响铃，时间一长，狗就会将铃声和食物联系起来。以后如果响铃但是不给食物，狗也会流口水。



6.2.4 神经网络工作方式

- 受该实验的启发，Hebb的理论认为在同一时间被激发的神经元间的联系会被强化。比如，铃声响时一个神经元被激发，在同一时间食物的出现会激发附近的另一个神经元，那么这两个神经元间的联系就会强化，从而记住这两个事物之间存在着联系。相反，如果两个神经元总是不能同步激发，那么它们间的联系将会越来越弱。
- Hebb学习律可表示为:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha y_j(t) y_i(t)$$

- 其中wij表示神经元j到神经元i的连接权，yi与yj为两个神经元的输出，a是表示学习速度的常数。若yi与yj同时被激活，即yi与yj同时为正，那么Wij将增大。若yi被激活，而yj处于抑制状态，即yi为正yj为负，那么Wij将变小。

6.2.4 神经网络工作方式

- 有导师学习算法：Delta学习规则
- Delta学习规则是一种简单的有导师学习算法，该算法根据神经元的实际输出与期望输出差别来调整连接权，其数学表示如下：

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(\, d_i - \, y_i \,)\, x_j(t)$$

- 其中Wij表示神经元j到神经元i的连接权，di是神经元i的期望输出，yi是神经元i的实际输出，xj表示神经元j状态，若神经元j处于激活态则xj为1，若处于抑制状态则xj为0或－1（由激励函数定定）。a是表示学习速度的常数。假设xi为1，若di比yi大，那么Wij将增大，若di比yi小，那么Wij将变小。
- Delta规则简单讲来就是：若神经元实际输出比期望输出大，则减小所有输入为正的连接的权重，增大所有输入为负的连接的权重。反之，若神经元实际输出比期望输出小，则增大所有输入为正的连接的权重，减小所有输入为负的连接的权重。这个增大或减小的幅度就根据上面的式子来计算。

【例6.1】字符识别

神经网络的构建

- 设想有一个由8x8个格子组成的一块面板。每一个格子里放了一个小灯，每个小灯都可独立地被打开（格子变亮）或关闭（格子变黑），这样面板就可以用来显示十个数字符号。下图显示了数字“4”。

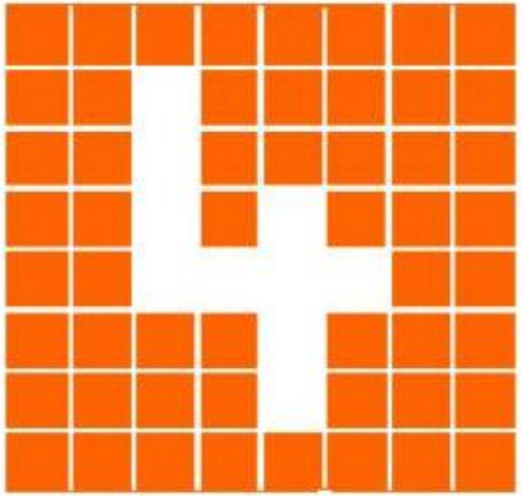


图6 用于字符显示的矩阵格点

神经网络的构建

- 4的显示方式有许多种，如何实现对“4”的识别?
- 方法1：穷举“4”的所有显示方式，困难。
- 方法2：设计一个神经网络，它接收面板的状态作为输入，然后输出一个1或0；输出1代表ANN确认已显示了数字“4”，而输出0表示没有显示“4”。
  - 因此，神经网络的输入层需要64个节点(每一个输入代表面板的一个具体格点)
  - 隐藏层包括96个神经细胞
  - 输出层包括一个神经细胞。

神经网络的构建

- 上述神经网络能识别“4”，如何使网络能识别0到9 的所有数字?
  - 把输出增加到10。
- 如何使网络能识别字母表中的全部字符?
- 这本质上就是手写体识别的工作原理。对每个字符，网络都需要接受许多训练，使它认识此文字的各种不同的版本。到最后，网络不单能认识已经训练的笔迹，通过它的显著的归纳和推广能力，还能识别不同于训练集中的笔迹。正是这种归纳推广能力，使得神经网络已经成为能够用于无数应用的一种无价的工具，从人脸识别、医学诊断，直到跑马赛的预测等。