第5章：神经网络学习

蒋良孝 |  中国地质大学（武汉）

机器学习与数据挖掘团队

ljiang@cug.edu.cn

http://grzy.cug.edu.cn/jlx/

本章内容

一、神经网络的定义

二、神经网络的发展历史

三、M-P神经元模型

四、单层感知机

五、多层前馈神经网络

六、深层神经网络

一、神经网络的定义

神经网络的命名：

- ✓ 神经网络（ Neural Networks ）
- ✓ 人工神经网络（ Artificial Neural Networks ）
- ✓ 人工神经系统（ Artificial Neural Systems）
- ✓ 神经计算机（Neural Computers）
- ✓ 自适应系统（Adaptive Systems）
- ✓ 自适应网（Adaptive Networks）
- ✓ 联结主义（Connectionisms）

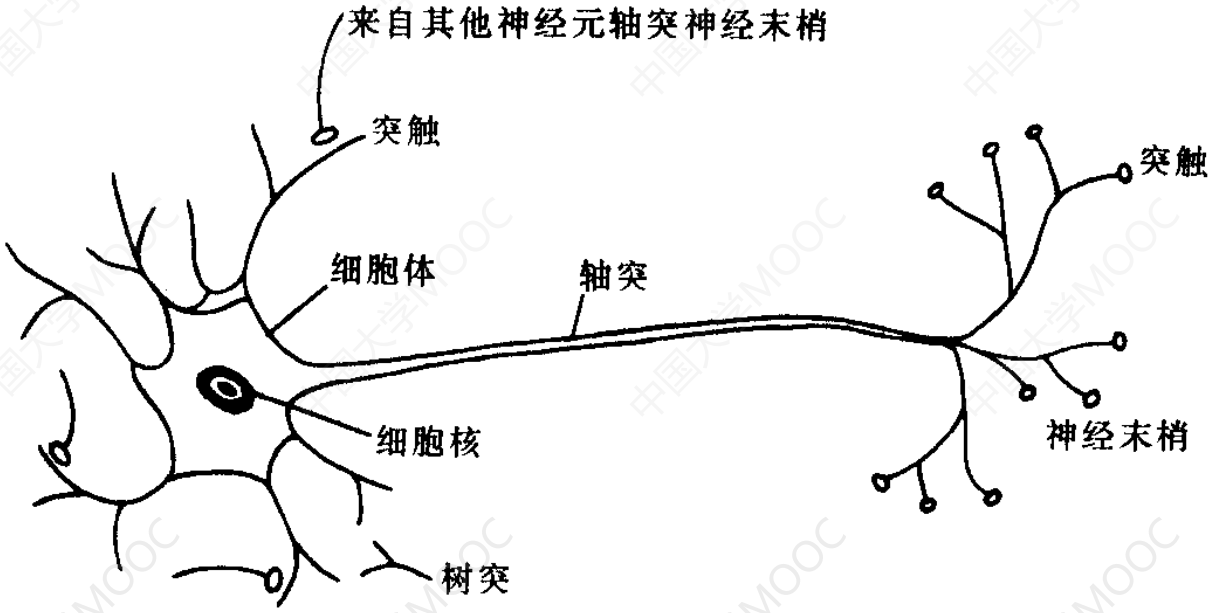
一、神经网络的定义

“神经网络是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反应。” [Kohonen, 1988]

- 机器学习中的神经网络通常是指“神经网络学习”，或者说是机器学习与神经网络两个学科的交叉部分。
- 既然神经网络的研究是由试图模拟生物神经系统受启发而来的。因此，有必要先来看看生物神经系统的工作过程：

一、神经网络的定义

- 生物神经系统：每个神经元通过轴突与其他相邻的神经元相连，当神经元受到刺激而“兴奋”时，就会向相连的神经云传递神经脉冲，从而改变这些神经元内的电位；如果神经元的电位超过一个“阈值”，那么它就会被激活，即“兴奋”起来，再向其它相连的神经元传递神经脉冲。



二、神经网络的发展历史

可以说，神经网络的发展是非常曲折的，从诞生到现在，几经兴衰。大体上，可以将其发展历史分成如下五个时期：

- ✓ 萌芽期（人类研究自己智能的开始~1949）
- ✓ 第一次高潮期（1950~1968）
- ✓ 反思期（1969~1982）
- ✓ 第二次高潮期（1983~1990）
- ✓ 再认识与应用期（1991~）

二、神经网络的发展历史

萌芽期（人类研究自己智能的开始~1949）：

- 人类对神经网络的研究最早可以追溯到人类研究自己智能的开始。
- 1943年，心理学家McCulloch和数学家Pitts建立起了著名的阈值加权和模型，简称为M-P模型。该成果发表在数学生物物理学会会刊上。
- 1949年，心理学家Hebb提出神经元之间的突触联系是可变的假说——Hebb学习规律。Hebb学习规律被认为是神经网络学习算法的里程碑。
- 这两个重大的研究成果，构成了神经网络萌芽期的标志。

二、神经网络的发展历史

第一次高潮期（1950~1968）：

- 1958年，Rosenblatt提出了单层感知机(Perceptron)模型及其学习规则。
- 单层感知机的成功标示着神经网络研究的第一高潮期的到来。
- 第一高潮期的成功让人们乐观地认为几乎已经找到了智能研究的关键。因此，许多部门都开始大规模地投入此项研究，希望尽快占领神经网络研究的制高点。

二、神经网络的发展历史

反思期（1969~1982）：

- 1969年，正当人们兴奋不已的时候，Minsky和Papert发表了《Perceptrons》一书，明确指出：单层感知机不能解决非线性问题，多层网络的训练算法尚无希望。
- 这一成果的发表标志着人类对神经网络的研究进入了反思期。很多献身于神经网络研究的科学家的研究结果很难得到发表，而且是散布在各种杂志之中，使得不少有意义的研究成果即使发表了，也很难被同行看到，著名的BP算法的研究就是一个典型的例子。
- 反思期的到来揭示了人类的认识规律：认识->实践->再认识 。

二、神经网络的发展历史

第二次高潮期（1983~1990）：

- 1982年，Hopfield提出了循环网络：引入李雅普诺夫(Lyapunov)函数作为网络性能判定的能量函数，建立了神经网络稳定性的判别依据；阐明了神经网络与动力学的关系；指出用非线性动力学的方法来研究神经网络的特性、信息是被存放在网络中神经元的联接权上。
- 1984年，Hopfield设计并实现了后来被人们称为Hopfield网络的电路。较好地解决了著名的TSP问题，找到了最佳解的近似解，引起了较大的轰动。
- Hopfield这两项成果分别于1982和1984年发表在美国科学院院刊上，标志着神经网络研究第二高潮期的到来。

二、神经网络的发展历史

第二次高潮期（1983~1990）：

- 1985年，Hinton、Sejnowsky、Rumelhart等人所在的并行分布处理(PDP)小组在Hopfield网络中引入了随机机制，提出Boltzmann机。
- 1986年，PDP小组的Rumelhart等研究者重新独立地提出了多层神经网络的学习算法—BP算法，较好地解决了多层神经网络的学习问题。
- 1987年，IEEE 在美国加州圣地亚哥召开第一届神经网络国际会议，宣告了国际神经网络协会正式成立，掀起了人类向生物学习、研究及应用神经网络的新热潮。
- 1990年，国内首届神经网络大会在北京举行。

二、神经网络的发展历史

再认识与应用期（1991~）：

- 90年代初，伴随统计学习理论和SVM的兴起，神经网络由于理论不够清楚，试错性强，难以训练，再次进入低谷。
- 2006年，Hinton提出了深度信念网络(DBN)，通过“预训练+微调”使得深度模型的最优化变得相对容易。
- 2012年，Hinton组参加ImageNet竞赛，使用CNN模型以超过第二名10个百分点的成绩夺得当年竞赛的冠军。

二、神经网络的发展历史

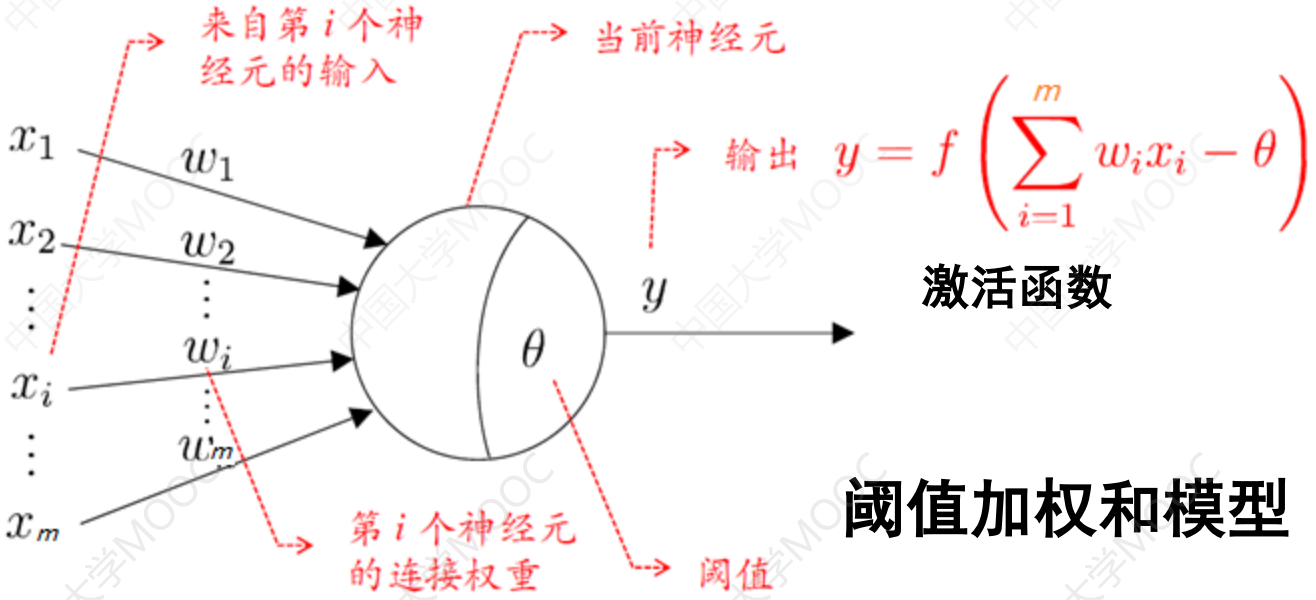
再认识与应用期（1991~）：

- 2016年3月谷歌人工智能AlphaGo在围棋比赛中4：1战胜世界围棋冠军李世石；2017年5月，又在浙江乌镇的围棋峰会上，以3：0完胜我国世界排名第一棋手柯洁。
- 2017年1月6日，最强大脑第四季引入“人机大战”模式，百度研发的人工智能机器人“小度”作为特别选手参赛，并在比赛中战胜了最强大脑的队长——王峰。
- 伴随云计算、大数据时代的到来，计算能力的大幅提升，使得深度学习模型在计算机视觉、自然语言处理、语音识别等众多领域都取得了较大的成功。

三、M-P神经元模型

最基本的信息处理单元：M-P神经元模型

- ✓ **输入：**来自其他m个神经云传递过来的输入信号
- ✓ **处理：**输入信号通过带权重的连接进行传递，神经元接受到总输入值将与神经元的阈值进行比较
- ✓ **输出：**通过激活函数的处理以得到输出



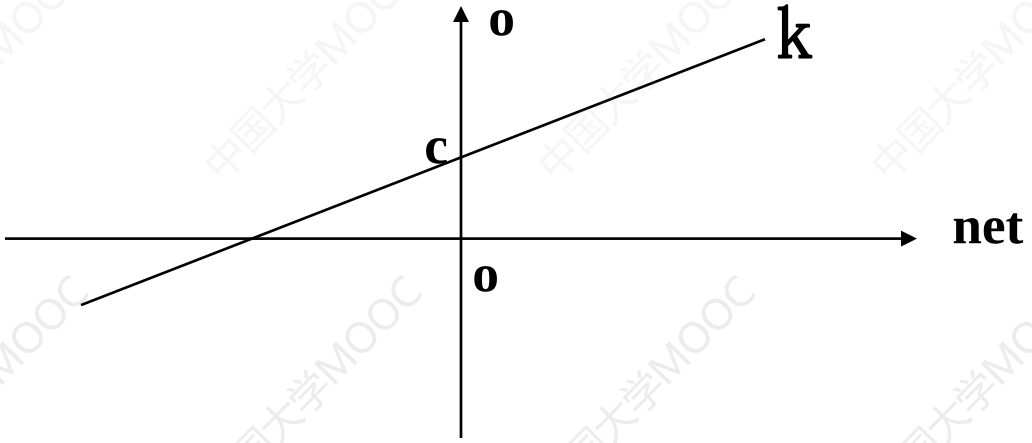
三、M-P神经元模型

- 激活函数(Activation Function)：对神经元所获得的网络输入的进行变换，也称为激励函数、活化函数：o=f（net）。常用的激活函数有：

  1、线性函数（Linear Function）

  f（net）=k*net+c

- ✓ 线性函数只能进行线性变化，不适合用来处理非线性问题。

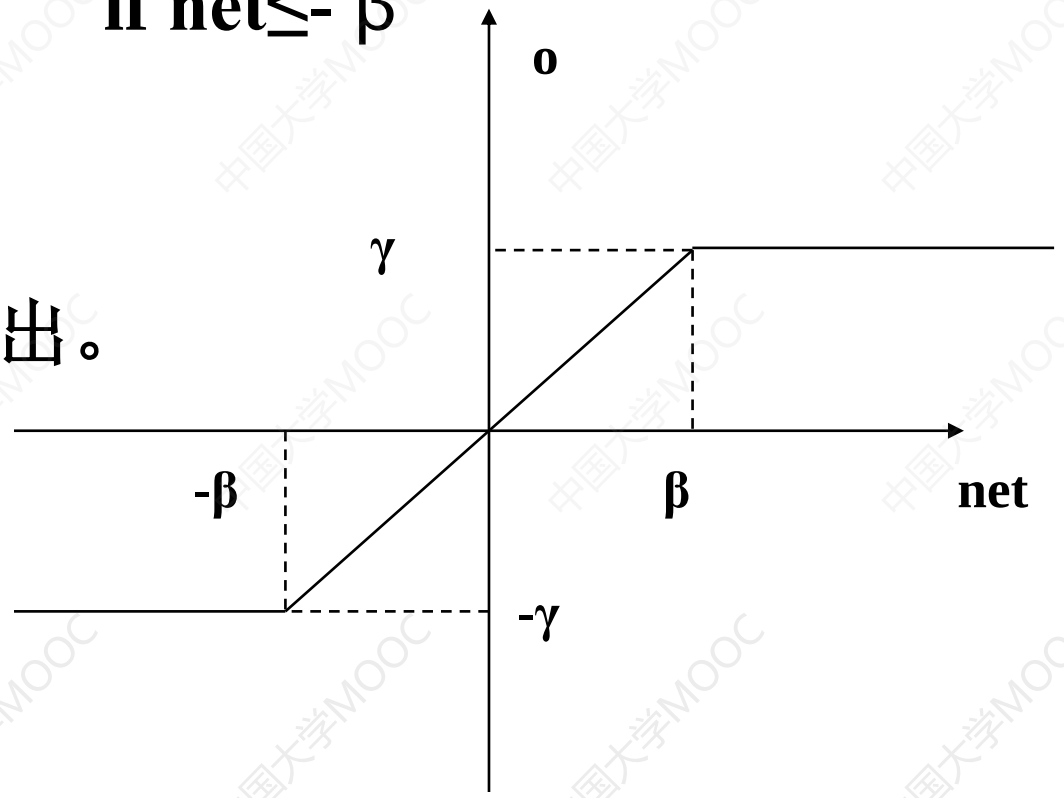


三、M-P神经元模型

2、非线性斜面函数（Ramp Function）

$$f(net)=\begin{cases}\gamma & \text{if } net\geq\beta\\ k*net & \text{if } |net|<\beta\\ -\gamma & \text{if } net\leq-\beta\end{cases}$$

- γ为大于0的常数，被称为饱和值，是神经元的最大输出。



三、M-P神经元模型

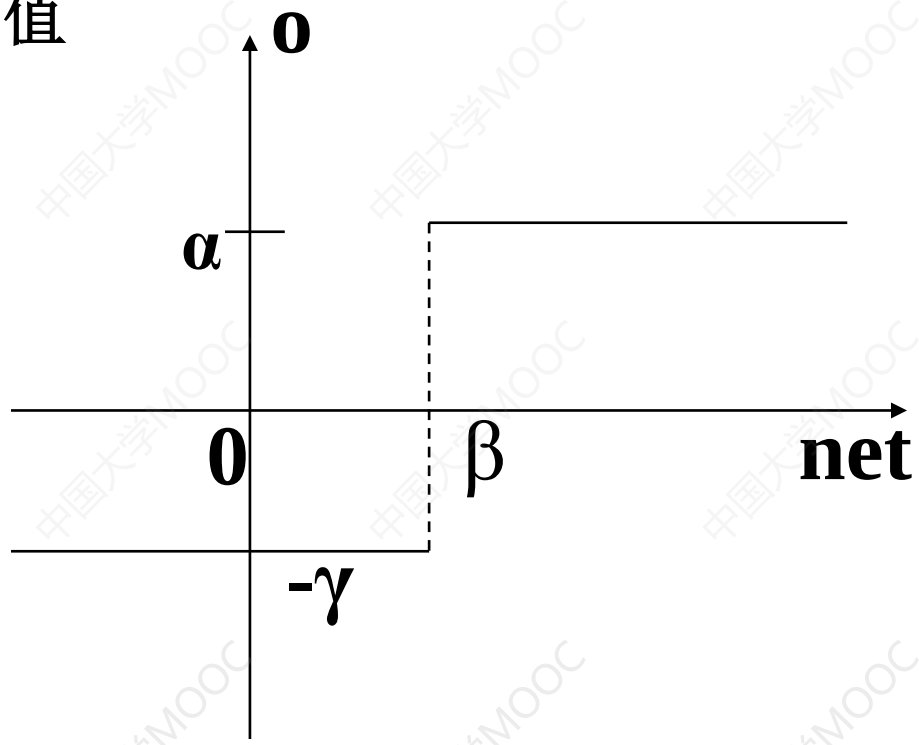
3、阈值函数（Threshold Function）

$$f(net)=\begin{cases}\alpha & \text{if } net>\beta \\ -\gamma & \text{if } net\le\beta\end{cases}$$

α、β、γ均为非负数，θ为阈值

二值形式：

$$f(net)=\begin{cases}1 & \text{if } net>\beta \\ 0 & \text{if } net\le\beta\end{cases}$$



三、M-P神经元模型

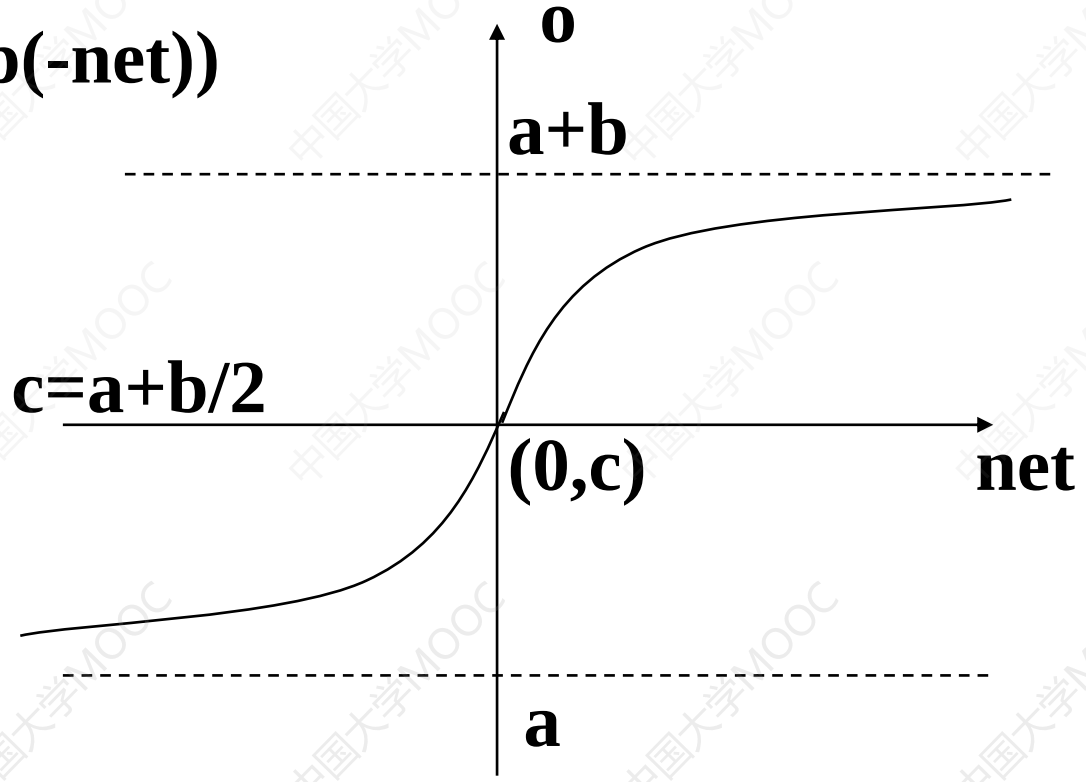
4、逻辑斯蒂函数（Logistic Function）

f（net）=a+b/(1+exp(-d*net))

a，b，d为常数。它的饱和值为a和a+b。

a=0，b=1，d=1时，简化为最简单的形式：

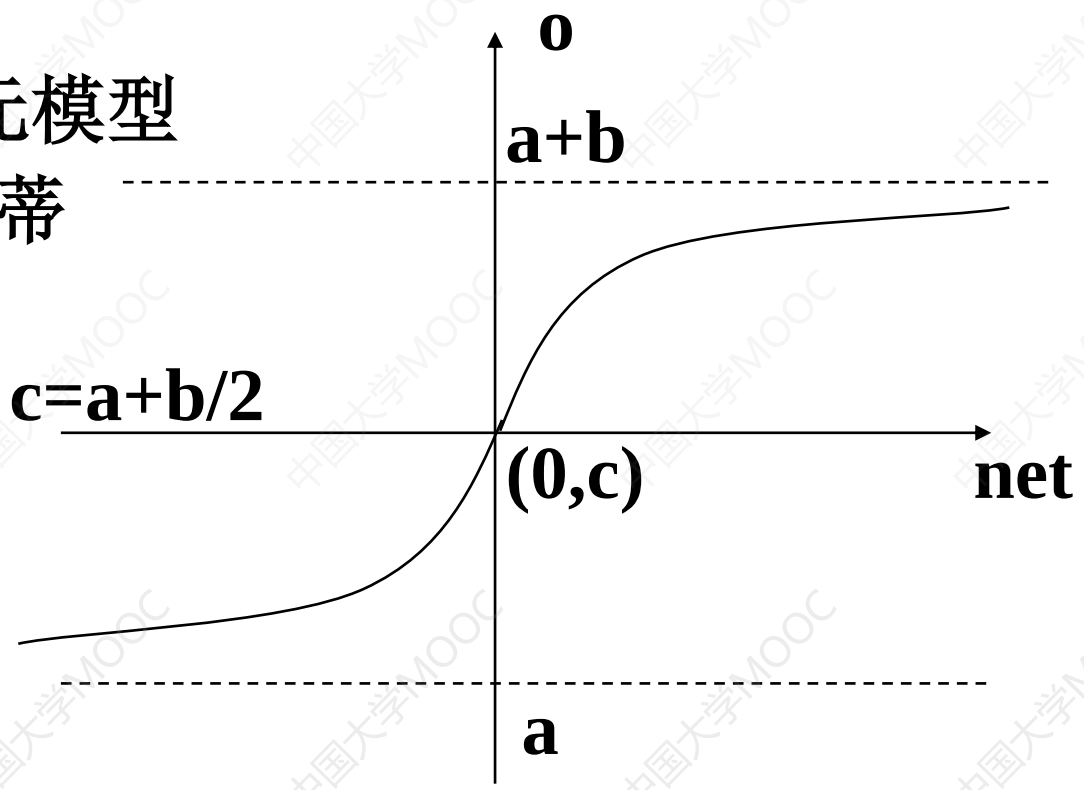
f（net）= 1/(1+exp(-net))



三、M-P神经元模型

4、逻辑斯蒂函数（Logistic Function）

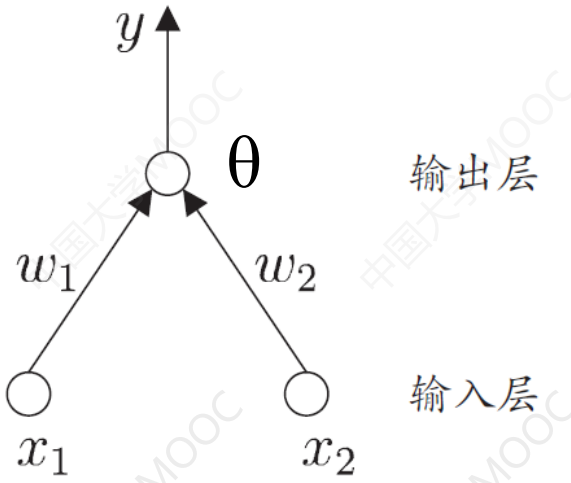
- 逻辑斯蒂函数形如S，也叫S形函数；将较大范围变化的输入值挤压到（0,1）的输出值范围内，因此也叫挤压函数；有较好的增益控制能力，也叫增益控制函数。
- 此时的M-P神经元模型就是一个逻辑斯蒂回归模型。



四、单层感知机

单层感知机的定义

- 单层感知机只拥有一层M-P神经元，即只包含输入层和输出层，输入层接受外界输入信号后传递给输出层，输出层是M-P神经元，进行激活处理。
- 比如，下图所示的M-P神经元模型，就是一个最简单的单层感知机，输出层只有一个神经元。



四、单层感知机

单层感知机的学习规则：

- 给定训练数据集，单层感知机的权重$w_i(i=1,2,\cdots,m)$与阈值θ可以很容易通过学习得到。
- 在单层感知机中，神经元的阈值θ可看作一个固定输入为-1的“哑结点”所对应的连接权重w_{m+1}，这样，权重和阈值的学习就统一为权值的学习了。
- 对训练样例$(\boldsymbol{x},y)$，若当前感知机的输出为$\hat{y}$，则感知机权重调整规则为：$w_i \leftarrow w_i + \triangle w_i$

$$\triangle w_i = (l)(y-\hat{y})x_i$$

其中$l \in (0,1)$为学习率。

若感知机对训练样例$(\boldsymbol{x},y)$预测正确，则感知机不发生变化；否则根据错误程度进行权重的调整。

四、单层感知机

单层感知机的学习能力：

- 尽管单层感知机能够非常容易地实现逻辑与、或、非运算，但其学习能力还是非常有限。
- 事实上，逻辑与、或、非运算问题都是线性可分的问题。可以证明[Minsky and Papert, 1969]，若二分类问题是线性可分的，即存在一个线性超平面能将它们分开，则感知机的学习过程一定会收敛，否则感知机的学习过程将会发生振荡，甚至不能解决异或这样简单的非线性可分问题。
- 要解决非线性可分问题，就需要使用多层功能神经元。比如两层感知机，输入层与输出层之间的一层神经元，被称为隐含层，隐含层和输出层神经元都是拥有激活函数的功能神经元。

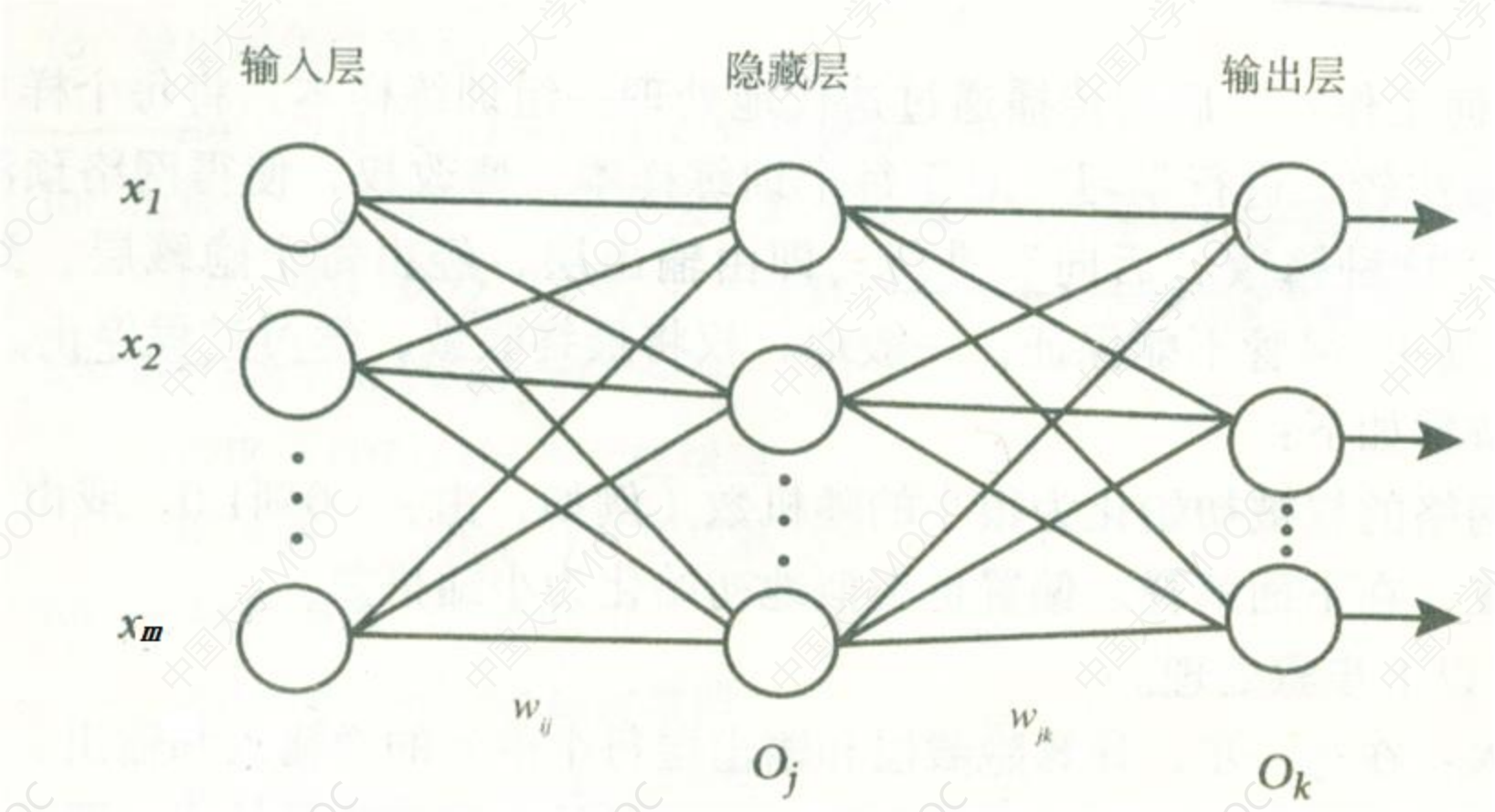
五、多层前馈神经网络

多层前馈神经网络的定义：

- 多层前馈神经网络又名多层前馈全互连接网。
- ✓ **多层是指：**除了输入层和输出层以外，还存在一个或者多个隐含层。
- ✓ **前馈是指：**外界信号从输入层，经由隐含层到达输出层，不存在信号的逆向传播。
- ✓ **全互连接是指：**每层神经元与下层神经元全互连接，神经元之间不存在同层连接， 也不存在跨层连接。

五、多层前馈神经网络

多层前馈神经网络的定义：



多层前馈网的结构图

五、多层前馈神经网络

多层前馈神经网络的定义：

- 其中输入层神经元接收外界输入，隐含层与输出层神经元对信号进行加工，最终结果由输出层神经元输出。也就是说，输入层神经元只是接受输入，不进行函数处理，隐含层与输出层才进行函数处理。
- 神经网络参数学习就是根据训练数据来调整神经元之间的连接权以及每个功能神经元的阈值。换言之，神经网络学到的东西都存放在连接权与阈值中。

五、多层前馈神经网络

多层前馈神经网络的学习：

- 多层前馈神经网络的表达能力比单层感知机要强得多。要学习多层前馈神经网络，单层感知机的学习规则是远远不够的，需要更强大的学习算法。误差逆传播（Error Back Propagation，简称BP）算法就是其中最杰出的代表，它是迄今最成功的神经网络学习算法。
- 现实任务中使用神经网络时，大多是在使用BP算法进行训练。需要指出的是，BP算法不仅可用于多层前馈神经网络，还可用于其他类型的神经网络，比如递归神经网络。但通常说BP网络的时候一般是指用BP算法训练的多层前馈神经网络。

五、多层前馈神经网络

BP算法：

基于梯度下降法则的公式推导可参阅：[周志华, 2016]; [Mitchell, 2003]。

1）初始化神经网络的权和偏置。

2）对实例中的每个训练样本：

① 计算每个隐含层节点的输入：$I_j = \sum_i X_i W_{ij} - \theta_j$

② 计算每个隐含层节点的输出：$O_j = 1/(1+e^{-I_j})$

③ 计算每个输出层节点的输入：$I_k = \sum_j O_j W_{jk} - \theta_k$

④ 计算每个输出层节点的输出：$O_k = 1/(1+e^{-I_k})$

⑤ 计算每个输出层节点的误差：$Err_k = O_k(1-O_k)(T_k - O_k)$

⑥ 计算每个隐含层节点的误差：$Err_j = O_j(1-O_j)\sum_k W_{jk} Err_k$

⑦ 更新输入层到隐含层的权值：$W_{ij} = W_{ij} + (l) X_i Err_j$

⑧ 更新隐含层到输出层的权值：$W_{jk} = W_{jk} + (l) O_j Err_k$

⑨ 更新每个隐含层节点的偏置：$\theta_j = \theta_j - (l) Err_j$

⑩ 更新每个输出层节点的偏置：$\theta_k = \theta_k - (l) Err_k$

3）判断终止条件是否满足，若满足则停止，否则重复执行步骤 2）。

五、多层前馈神经网络

BP算法的基本过程：

- BP算法的基本工作过程大致可以分为三个阶段：

✓ 1）信号的前向传播阶段：在这个阶段，要求计算出隐含层和输出层中每一神经元的网络净输入和网络输出。

✓ 2）误差的逆向传播阶段：在这个阶段，要求计算出输出层和隐含层中每一神经元的误差。

✓ 3）权值和阈值的更新阶段：在这个阶段，要求更新所有连接权的权值和所有M-P神经元的阈值。

五、多层前馈神经网络

BP算法可能面临的问题：

1）结构学习问题：

- ✓ 多层前馈神经网络包括：输入层、输出层、一个或多个隐含层。输入层神经元的个数由输入的数据维度（连续属性）和编码方法（离散属性）确定；输出层神经元的个数由待分类的类别数目和编码方法确定，比如4类问题的二进制编码需要2个神经元。
- ✓ [Hornik et al. , 1989]证明：只需要一个包含足够多神经元的隐含层，多层前馈神经网络就能以任意精度逼近任意复杂度的连续函数。
- ✓ 如何确定隐含层神经元的个数仍然是个悬而未决问题。实际应用中通常是根据经验来确定或者靠“试错法”来调整。

五、多层前馈神经网络

BP算法可能面临的问题：

2）初始化问题：

- ✓ 在BP算法中，连接权和偏置在网络学习之前，都需要将其初始化为不同的小随机数。“不同”保证网络可以学习；“小随机数”可以防止其值过大而提前进入饱和状态，达到局部极小值。
- ✓ 解决办法：重新初始化。

五、多层前馈神经网络

BP算法可能面临的问题：

3）步长设置问题：

- ✓ BP网络的收敛是基于无穷小的修改量，学习率控制着算法每一轮迭代中的更新步长。
- ➢ 步长太小，收敛速度就会过慢
- ➢ 步长太大，又可能会导致网络的不稳定、甚至瘫痪
- ➢ 自适应步长，让步长随着网络的训练而不断变化

五、多层前馈神经网络

BP算法可能面临的问题：

4）权值与阈值的更新问题：

- 基本的BP算法采用的是样例更新，即每处理一个训练样例就更新一次权值与阈值。样例更新的缺陷：参数更新频繁，不同样例可能抵消，需要迭代的次数较多。另外，训练样例的输入顺序对训练结果有较大影响，它更“偏爱”较后输入的样例。而给训练样例安排一个适当的顺序，又是非常困难的。
- 解决的办法就是采用周期更新，即每处理一遍所有的训练样例才更新一次权值与阈值。但在很多实际任务中，周期更新的累计误差下降到一定程度后，进一步下降会非常缓慢，这时样例更新往往会获得较好的解，尤其当训练集非常大时效果更明显。

五、多层前馈神经网络

BP算法可能面临的问题：

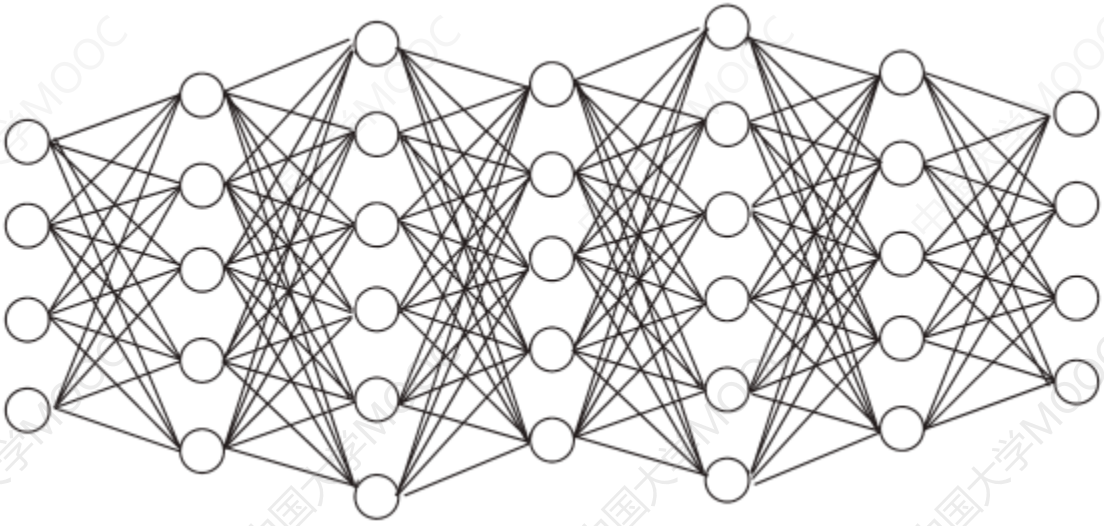
5）过拟合问题：

- ✓ 神经网络由于强大的表示能力，经常遭遇过拟合。具体表现为：训练误差持续降低，但测试误差却可能上升。
- ✓ 缓解过拟合的策略包括早停和正则化。早停就是在训练过程中, 若训练误差降低, 但验证误差升高, 则停止训练；正则化就是在误差目标函数中增加一项描述网络复杂程度的部分，比如连接权值与阈值的平方和。

六、深层神经网络

深层神经网络的定义

- 随着云计算、大数据时代的到来，计算能力的大幅提高可缓解训练低效性，训练数据的大幅增加则可降低过拟合风险，因此,以深度学习(deep learning)为代表的复杂模型开始受到人们的关注。
- 典型的深度学习模型就是很深层的神经网络：包含2个以上隐含层的神经网络。



六、深层神经网络

深层神经网络的定义

- 模型复杂度
  - ✓ 增加隐含层神经元的数目 （模型宽度）
  - ✓ 增加隐含层的数目（模型深度）
  - ✓ 从增加模型复杂度的角度来看，增加隐含层的数目比增加隐含层神经元的数目更有效。这是因为增加隐含层的数目不仅增加了拥有激活函数的神经元数目，还增加了激活函数嵌套的层数。
- 深层模型的难点问题
  - ✓ 多隐含层网络难以直接用经典算法（例如标准的BP算法）进行训练，因为误差在多隐含层内逆传播时，往往会发散，而不能收敛到稳定状态。

六、深层神经网络

深层神经网络的学习

- **预训练+微调**

- ✓ 预训练：无监督逐层训练是多隐含层网络训练的有效训练手段，每次训练一层隐含层结点，训练时将上一层隐含层结点的输出作为输入，而本层隐含层结点的输出作为下一层隐含层结点的输入，这称为“预训练”。
- ✓ 微调：预训练全部完成后，再对整个网络进行微调训练。微调一般使用BP算法。
- ✓ 例子：深度信念网络(DBN) [Hintonet al.，2006]。
- ✓ 结构：每一层都是一个受限Boltzmann机。
- ✓ 训练：无监督预训练+BP算法微调。
- ✓ 分析：预训练+微调的做法可以视为将大量参数进行分组，对每组先找到局部看起来比较好的设置，然后再基于这些局部较优的结果联合起来进行全局寻优。

六、深层神经网络

深层神经网络的学习

- **权共享**

✓ 一组神经元共享相同的连接权值。

✓ 例子：卷积神经网络(CNN) [LeCun and Bengio, 1995; LeCun et al.，1998]。

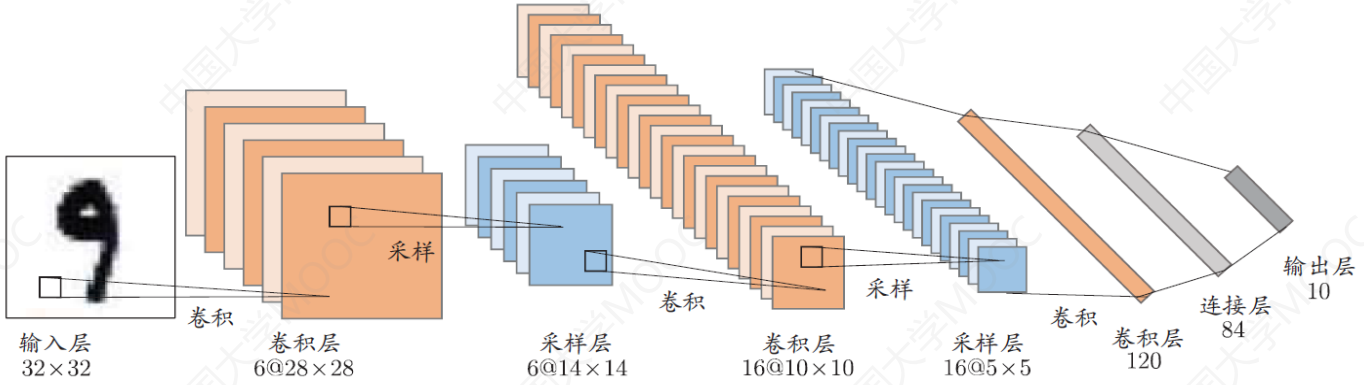
✓ 结构：CNN复合多个卷积层和采样层对输入信号进行加工，然后在连接层实现与输出目标之间的映射。

✓ 训练：BP算法。

✓ 分析：在训练中，无论是卷积层还是采样层，其每一组神经元（即图下中的每个平面）都共享相同的连接权，从而大幅减少了需要训练的参数数目；另外，在CNN中通常将Sigmoid 激活函数替换为修正的线性函数$f(x) = \max(0, x)$。

六、深层神经网络

深层神经网络的学习



卷积神经网络用于手写数字识别 [LeCun et al., 1998]

- **卷积层：**每个卷积层包含多个特征映射，每个特征映射是一个由多个神经元构成的“平面”，通过一种卷积滤波器提取输入的一种特征。
- **采样层：**也叫“汇合层”，其作用是基于局部相关性原理进行亚采样，从而在减少数据量的同时保留有用信息。
- **连接层：**每个神经元被全互连接到上一层每个神经元，本质就是传统的神经网络，其目的是通过连接层和输出层的连接完成识别任务。

六、深层神经网络

深度学习的理解

- 深度学习通过多层处理，逐渐将初始的低层特征表示转化为高层特征表示后，用“简单模型”即可完成复杂的分类学习等任务。由此可将深度学习理解为进行“特征学习”(feature learning) 或“表示学习” (representation learning)。
- 非深度学习技术用于解决现实任务时，描述样本的特征通常需由人类专家来手工设计，这称为“特征工程” (feature engineering)。众所周知，特征的好坏对模型的泛化性能有至关重要的影响，要人类专家手工设计出好特征并非易事；而特征学习则通过深度学习技术自动产生好特征，这使得机器学习向“全自动数据分析”又前进了一步。