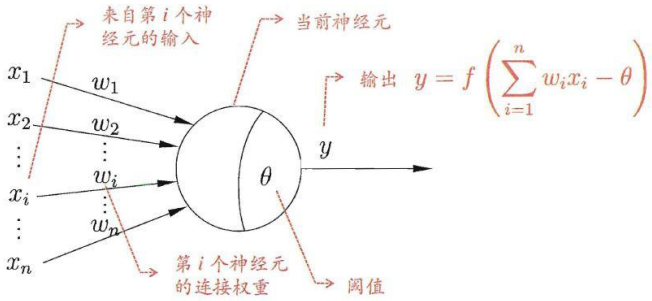
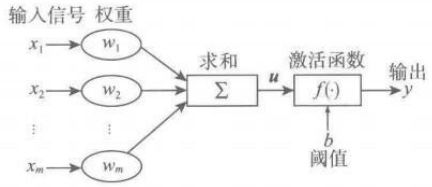
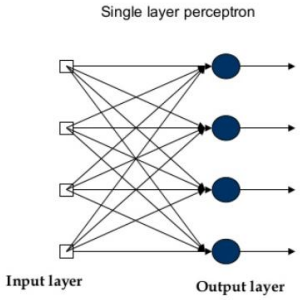
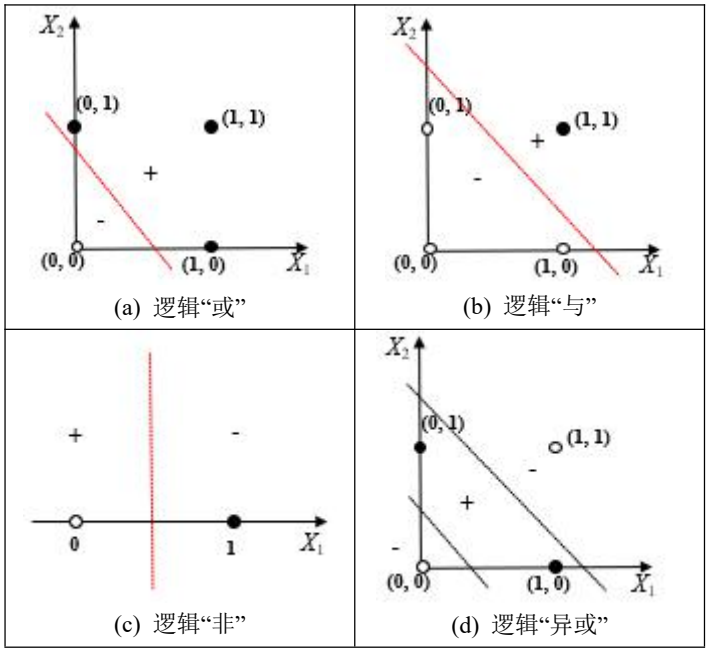
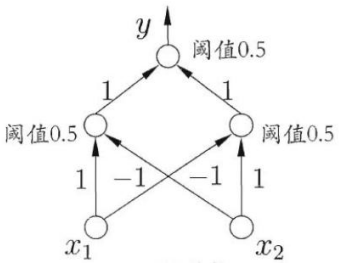
标题	神经网络学习笔记
概念	神经网络(Neural Network)是一种模仿生物神经网络构造的数学模型。要理解神经网络首先要知道它的几个重要概念：神经元模型、感知机、多层前馈神经网络与误差反向传播算法。
神经元模型	人工神经网络(Artificial Neural Networks，简称 ANNs)，亦称神经网络或连接模型，是一种模仿生物的神经网络行为特征，进行分布式并行信息处理的数学模型。人工神经网络的概念起源于生物神经网络，在神经系统中，神经元接收多个输入，具有空间整合特性和阈值特性，当神经元的膜电位超过某个阈值时兴奋，低于某个阈值时抑制。基于上述生物神经元的特性，神经学家 Warren McCulloch 和逻辑学家 Walter Pitts 于 1943 年提出了 M-P 神经元模型。  如图所示，x 模拟了神经系统中其他神经细胞给予当前神经细胞的刺激，刺激越大值越大；w 模拟了当前的神经细胞对于不同的刺激 x 的敏感度（权重）；θ表示神经元的阈值，只有神经元接收到的所有输入总和 $\sum_{n=1}^{n} w_i x_i$大于阈值时输出 y，小于阈值时输出 0；$y=f(\sum_{n=1}^{n} w_i x_i-\theta)$为激活函数，用来计算神经元的输出。也能用下图的数学模型来表示神经元模型。  根据以上信息我们可以得到一个基本的神经元表示（也即是分类公式） 线性求和：$h=\sum_{n=1}^{n} w_i x_i$ 阈值比较：$y=f(h-\theta)\begin{cases}1 & \text{if } h>\theta \\ 0 & \text{if } h<\theta\end{cases}$
感知机	感知机从结构上说就是多个 M-P 神经元模型的累叠。它由输入层和输出层构成，单层感知机结构如下图。

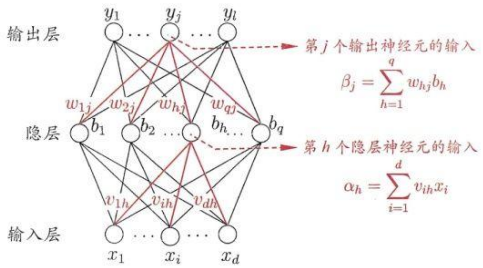
<table>
<tr>
<td></td>
<td>


从神经元的数学模型可以看出，感知机本质上是一个存在于样本空间的超平面，它将空间分成了两个部分：正半空间和负半空间。其中，产生输出 1 的输入位于正半空间，产生输出 0 的输入位于负半空间。也就是说，感知机可用于实现线性可分函数，能够很容易地实现逻辑与、或、非运算。图中的“+”和“-”表示正半空间和负半空间。在图中，分别用黑色的点和白色的点表示类别 1 和 0。可以看出，可以很容易的找到一个超平面去划分“或”、“与”、“非”中的不同类别的点。


但注意图(d)，感知机作为一个二分类的数学模型并不能处理“异或”的逻辑判断，原因是“异或”问题是一个线性不可分问题，模型找不到一条超平面像其他三个逻辑判断一样将数据空间明确划分，这便是明斯基所提出来的感知机模型的天生缺陷。</td>
</tr>
<tr>
<td>多层前馈神经网络</td>
<td>单层的感知机模型无法处理线性不可分问题，那就在单层感知机模型的基础上加多一层神经元，如下图所示，实现了对逻辑“异或”的判断。</td>
</tr>
</table>



进一步地，我们可以将大量的神经元按照不同的方式连接起来，得到不同类型的神经网络，实现更强大的功能，例如多层前馈神经网络(multi-layer feedforward neural networks)和循环神经网络(recurrent neural network)。多层前馈神经网络是最常见也是最常用的一种网络。多层前馈神经网络中多层指的是：在输入层及输出层中间还有拥有激活函数的激活层，隐含层至少一层，也可以有多层，也就是说一个多层前馈神经网络最少有三层，多则没有上限。示意图如下：



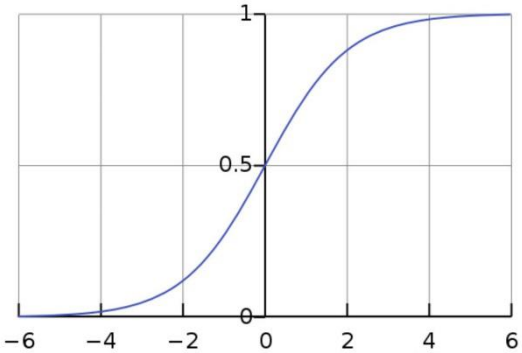
多层前馈神经网络示意图

而前馈指的是该层的神经元与下一层的神经元全连接，同层的神经元之间不连接，也不存在跨层连接，信息的输入只能从输入层至隐含层再至输出层。

在感知层模型中提到过，因为神经元模型的激活函数的性质，感知层实质上就是关于分类的数学模型。如果激活函数使用阶跃函数，多层前馈神经网络的输出就仍然还是 1 或者 0 两种结果。这将导致模型在训练时参数的调整出现困难。在多数情况，激活函数常都使用 sigmoid 函数

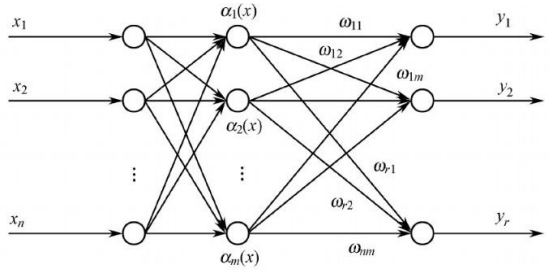
$$f(x) = 1/(1 + e^{-x})$$

与阶跃函数相比，sigmoid 函数具有平滑、连续的性质，因此在实际中应用的较多。下图给出的是 sigmoid 函数的图形，像一条 S 型曲线。可以看出，sigmoid 函数的输出范围是[0, 1]，相当于对神经元的输出做了归一化，可以用于将预测概率作为输出的模型。



误差反向传播算法	在讲述上面概念时，我们并没有提及到训练神经网络模型时的主角，也就权重 w 与阈值θ。和训练线性模型一样，训练神经网络本质上也是要找到一组参数 (w,θ)，使得代价函数,也就是整体的输出值与实际值的误差最小。通常来讲，都是用梯度下降法来解决这个问题。然而由于神经网络的特殊结构，神经网络的全互联结构，使得神经网络有着数量庞大的参数总量，这无疑会降低网络训练速度。而神经网络的层级结构，使得它的输出 y 是一个复合了好几次的复合函数。如果我们直接求导，那么根据链式求导的规则，每一个参数的偏导数会写成一个看起来就很复杂的式子， 毫无疑问这样的公式算起来效率是很低的。所以在网络架构已经确定的情况下，如何确定神经网络中的连接权重，即根据训练数据来调整神经元之间的连接权重以及每个神经元的阈值，我们常使用误差反向传播算法(error BackPropagation，简称 BP 算法)。BP 算法是迄今最为成功、最为广泛使用的神经网络学习算法。BP 算法不仅适用于多层前馈神经网络，而且适用于其他类型的神经网络，比如循环神经网络，递归神经网络等。 BP 算法是一种快速求导的算法。以多层前馈神经网络示意图举例，假设第一层权重v_{1h}与理想值有偏差Δv，这里的误差会造成隐层第 h 个神经元的输出b_h产生$\frac{\partial b_h}{\partial v_{ih}}\Delta v$ 的误差。而b_h又是所有输出层神经元的输入，所以会影响到所有输出层神经元的输出，最终造成总代价产生$\frac{\partial cost\,function}{\partial v_{ih}}\Delta v$ 的误差。既然误差是一层层传过来的，那每一层的误差也可以用上一层（上指的是更靠近输出端）的误差来表示。于是可以通过代价函数->输出层->隐含层，一层一层反向传播，每一层的偏导数都用上一层的偏导数来表示，以此通过一次反向传播就把所有参数的偏导数都求解出来，最后通过梯度下降法求出最适合的参数 (w,θ)。这就是 BP 算法快速求导的原理。
RBF 神经网络	径向基函数神经网络(radial basis function neural net.work)是一种具有单隐层的 3 层前馈网络，与 BP 网络不同，RBF 网络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)，而激活函数采用径向基函数(如高斯函数)径向基函数是一种局部分布的中心点径向对称衰减的非负非线性函数，RBF 隐含层可把向量从低维度的 p 映射到高维度的 h，将低维度线性不可分转换为高维线性可分。这样，网络由输入到输出的映射是非线性的；而网络输出对可调参数而言却又是线性的，网络的权就可由线性方程组直接解出，从而大大加快学习速度并避免局部极小问题。 径向基函数是一个取值仅仅依赖于离原点距离的实值函数，也就是$\Phi(x)=\Phi(\|x\|)$，或者还可以是到任意一点 c 的距离，c 点称为中心点，也就是$\Phi(x,c)=\Phi(\|x-c\|)$。任意一个满足$\Phi(x)=\Phi(\|x\|)$特性的函数Φ都叫做径向基函数，标准的一般使用欧氏距离（也叫做欧式径向基函数），尽管其他距离函数也是可以的。最常用的径向基函数是高斯核函数, 形式为u_i=k($\|x-c_i\|$) $= exp\{-\|x-c_i\|\text{^}2/(2*\sigma)^2)\}$，其中$u_i$为为第 i 个隐节点的输出，c_i为核函数中心 ,σ为函数的扩展常数, 控制了函数的径向作用范围。 RBF 神经网络的拓扑结构是一种三层前向网络: 1. 输入层由信号源结点构成,仅起到数据信息的传递作用，对输入信息不进行任何变换; 2. 第二层为隐含层，结点数视需要而定，隐含层神经元的核函数(激活函数)为高斯函数。对输入信息进行空间映射变换; 3. 第三层为输出层，它对输入模式做出响应，输出层神经元的作用函数为线性函数，对隐含层神经元输出的信息进行线性加权后输出，作为

整个神经网络的输出结果。



径向基函数的网络结构

RBF 网络算法可分为 2 个阶段，它们所起的作用完全不同。在第一阶段，采用无监督训练方法确定基函数的参数(当基函数取高斯函数时，这些参数为基函数的中心 c 和扩展常数σ)，属于非线性变换；而在第二阶段，基函数固定，网络输出是隐层基函数的线性组合，所以调节该输出层权值属于线性映射。

$$\sigma_i = \frac{c_{\max}}{\sqrt{2h}} \qquad i = 1, 2, \ldots\ldots, h$$

h 为中心个数，$c_{\max}$ 为所选取中心点之间的最大距离。隐含层至输出层之间的神经元的连接权值可以用最小二乘法直接计算得到，即对损失函数求解关于 w 的偏导数，使其等于 0，可以化简得到计算公式为：

$$w = \exp\left(\frac{h}{c_{\max^2}}\left\|x_p - c_i\right\|^2\right) \quad p = 1, 2, \cdots, P; i = 1, 2, \cdots, h$$