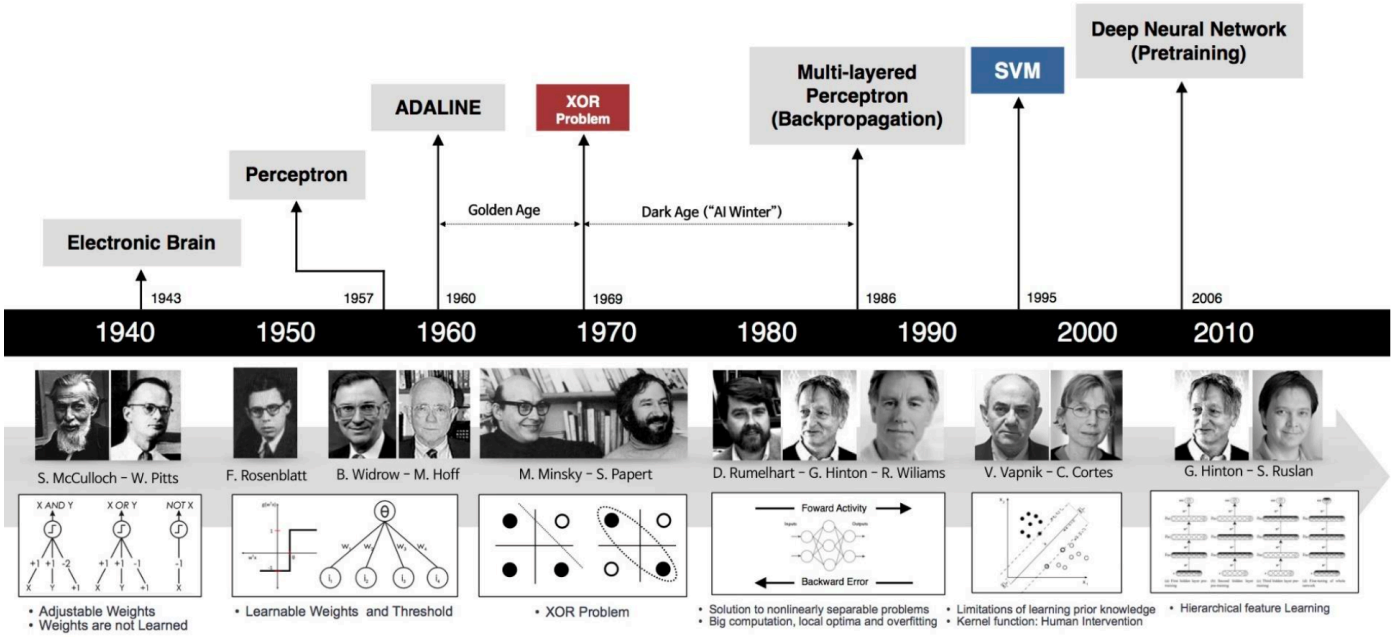
前馈神经网络

1. 神经元模型
2. 感知器、多层感知器
3. BP算法
4. 前馈神经网络

神经网络是最早作为一种连接主义为主的模型。

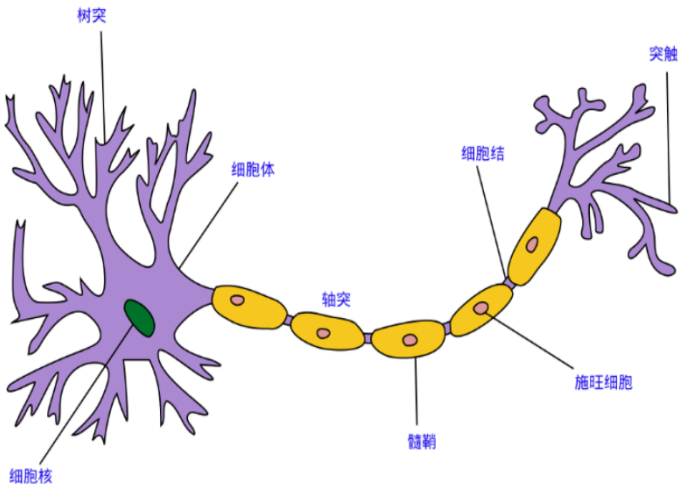
神经元模型

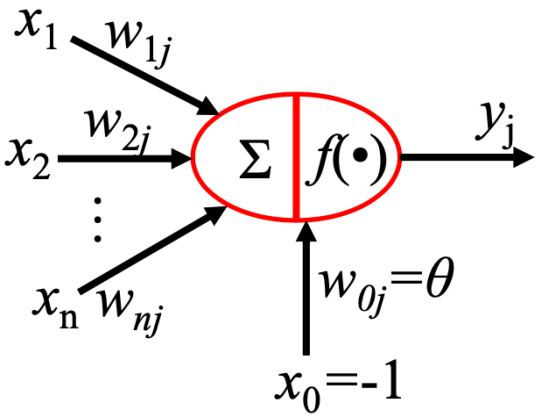
神经网络的发展历程



神经元(M-P)

1943 年，美国神经生理学家沃伦·麦卡洛克(Warren McCulloch) 和数学家沃尔特 ·皮茨(Walter Pitts)对生物神经元进行建模，首次提出了一种形式神经元模型，并命名为McCulloch-Pitts模型，即后 来广为人知的M-P模型。



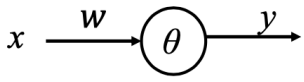


在M-P模型中，神经元接受其他n个神经元的输入信号(0或1)，这些输入信号经过权重加权并求和，将求和结果与阈值(threshold) θ 比较，然后经过激活函数处理，得到神经元的输出。

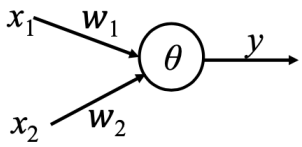
$$y = f\left(\sum_{i=1}^{n} \omega_{ij} x_i - \theta\right)$$

M-P 模型可以表示多种逻辑运算，如取反运算、逻辑或、逻辑与。

- 取反运算可以用单输入单输出模型表示，即如果输入为0则输出1，如果输入为1则输出0。由M-P模型的运算规则可得 $w = -2$，$\theta = -1$.

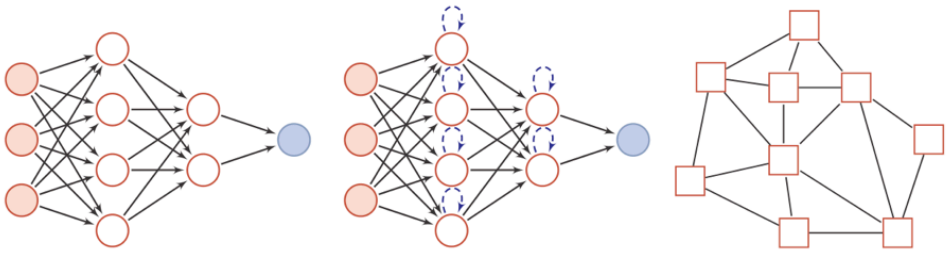


- 逻辑或与逻辑与运算可以用双输入单输出模型表示。以逻辑与运算为例， $w_1 = 1$，$w_2 = 1, \theta = 1.5$.



网络结构

人工神经网络由神经元模型构成，这种由许多神经元组成的信息处理网络具有并行分布结构。



(a) 前馈网络　(b) 记忆网络　(c) 图网络

其中圆形节点表示一个神经元，方形节点表示一组神经元。

感知器

单层感知器

1958 年，罗森布拉特(Roseblatt)提出了感知器，与 M-P 模型需 要人为确定参数不同，感知器能够通过训练自动确定参数。训练方式为有监督学习，即需要设定训练样本和期望输出，然后调整实际输出和期望输出之差的方式(误差修正学习)。

$$w_i \leftarrow w_i + \alpha(r-y)x$$
$$\theta \leftarrow \theta - \alpha(r-y)$$

其中，α 是学习率，r 和 y 分别是期望输出和实际输出。

感知器权重调整的基本思路：

- 实际输出 y 与期望输出 r 相等时，w 和 θ 不变
- 实际输出 y 与期望输出 r 不相等时，调整 w 和 θ 的值

$$w_i \leftarrow w_i + \alpha(r-y)x$$
$$\theta \leftarrow \theta - \alpha(r-y)$$

实际输出$y=0$, 期望输出$r=1$时（未激活）

- 减小θ
- 增大 $x_i=1$ 的连接权重 ω_i,
- $x_i=0$ 的连接权重不变

实际输出$y=1$, 期望输出$r=0$时（激活过度）

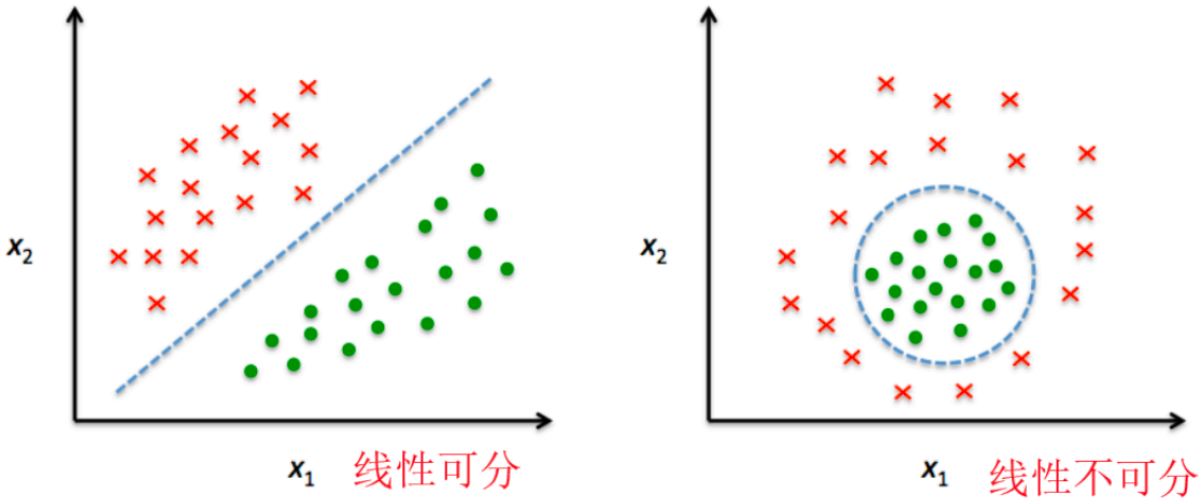
- 增大θ
- 降低 $x_i=1$ 的连接权重 ω_i,
- $x_i=0$ 的连接权重不变

下面给出感知器模型的训练过程

0. 训练准备
   - 准备 N 个训练样本x_i，和期望输出r_i
   - 初始化参数 ω_i和θ
1. 调整参数
   - 1.1. 迭代调整，直到误差为0或小于某个指定数值
     - 1.1.1. 逐个加入训练样本，计算实际输出
     - ：实际输出和期望输出相等时，参数不变
     - ：实际输出和期望输出不同时，通过误差修正学习调整参数
     - 重复上述步骤 1.1.1
   - 重复上述步骤1.1

多层感知器

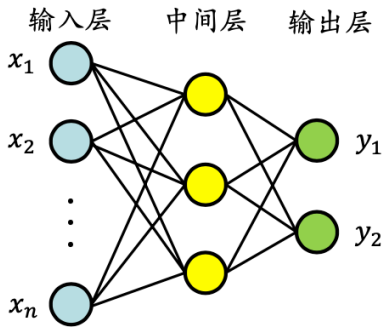
单层感知器只能解决线性可分问题，而不能解决线性不可分问题；为了解决线性不可分问题，我们需要使用多层感知器。



多层感知器指的是由多层结构的感知器递阶组成的输入值向前传播的网络，也被称为前馈网络或正向传播网络。

以三层结构的多层感知器为例，它由输入层、中间层及输出层组成

- 与M-P模型相同，中间层的感知器通过权重与输入层的各单元相连接，通过阈值函数计算中间层各单元的输出值
- 中间层与输出层之间同样是通过权重相连接



BP算法

多层感知器的训练使用误差反向传播算法(Error Back Propagation)，即BP算法。BP算法最早有沃博斯于1974年提出，鲁梅尔哈特等人进一步发展了该理论。

BP算法的基本过程

- 前向传播计算：由输入层经过隐含层向输出层的计算网络输出
- 误差反向逐层传递:网络的期望输出与实际输出之差的误差信号由输出层经过隐含层逐层向输入层传递
- 由“前向传播计算”与“误差反向逐层传递”的反复进行的网络训练 过程

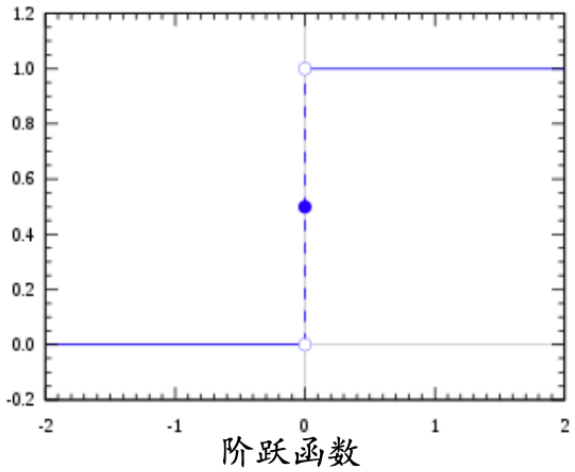
BP算法就是通过比较实际输出和期望输出得到误差信号，把误差信 号从输出层逐层向前传播得到各层的误差信号，再通过调整各层的连接权重以减小误差。权重的调整主要使用梯度下降法：

$$\Delta w = -\alpha \frac{\partial E}{\partial w}$$

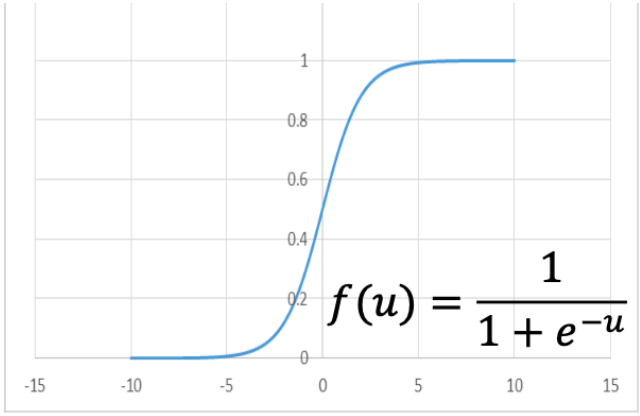
激活函数

通过误差反向传播算法调整多层感知器的连接权重时，一个瓶颈问题就是**激活函数**：

- M-P 模型中使用阶跃函数作为激活函数，只能输出 0或 1，不连续所以 不可导
- 为了使误差能够传播，鲁梅尔哈特等人提出使用可导函数Sigmoid作为激活函数

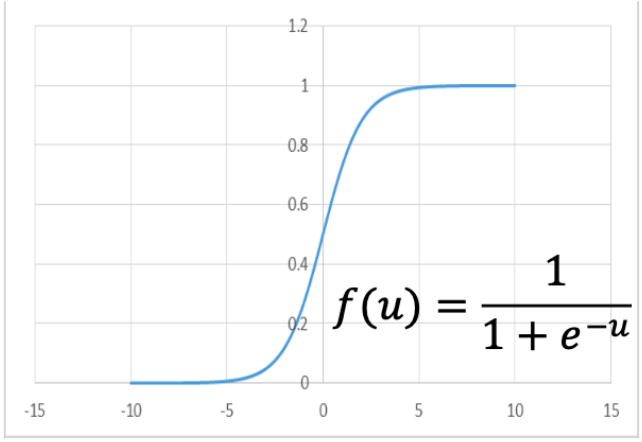


阶跃函数

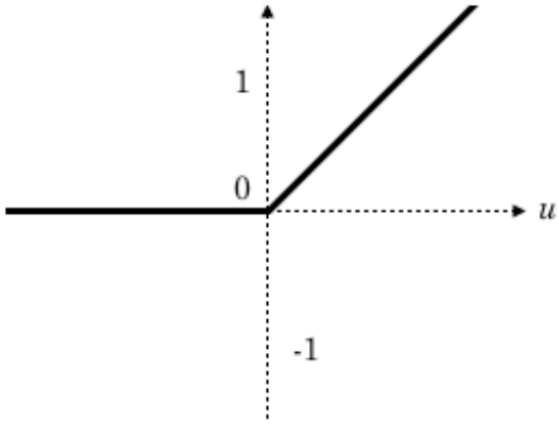


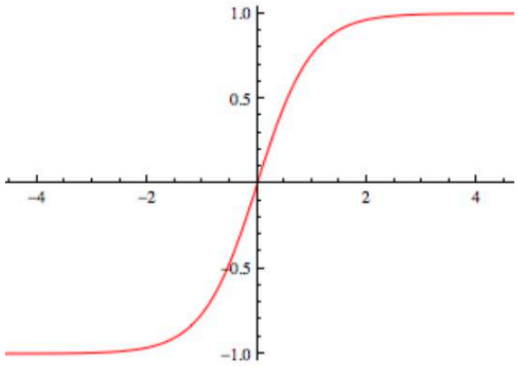
Sigmoid函数

Sigmoid函数的导数：$\frac{df(u)}{du} = f(u)(1 - f(u))$



其他常见的激活函数：ReLU (Rectified Linear Unit，修正线性单元)和tanh等





ReLu

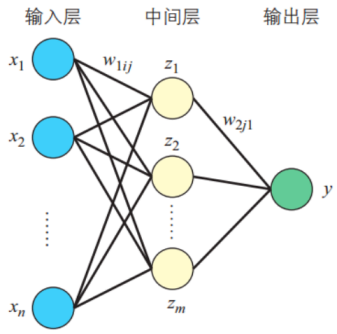
$$\mathrm{ReLu}(x) = \begin{cases} x, & x > 0 \\ 0, otherwise \end{cases}$$

tanh

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

BP算法示例

以包含一个中间层和一个输出单元 y 的多层感知器为例：w_{1ij} 表示输 入层与中间层之间的连接权重，w_{2j1} 表示中间层与输出层之间的连接权重， i 表示输入层单元，j 表示中间层单元。



- 首先调整中间层与输出层之间的连接权重，其中 $y = f(u)$， f 是激活函数，$u_{21} = \sum_{j=1}^{m} w_{2j1} z_j$，把误差函数 E 对连接权重$w_{2j1}$ 的求导展开成复合函数求导：

$$\begin{aligned} \frac{\partial E}{\partial w_{2j1}} &= \frac{\partial E}{\partial y} \frac{\partial y}{\partial u_{21}} \frac{\partial u_{21}}{\partial w_{2j1}} \\ &= -(r - y) y (1 - y) z_j \end{aligned}$$

这里 z_j 表示的是中间层的值。

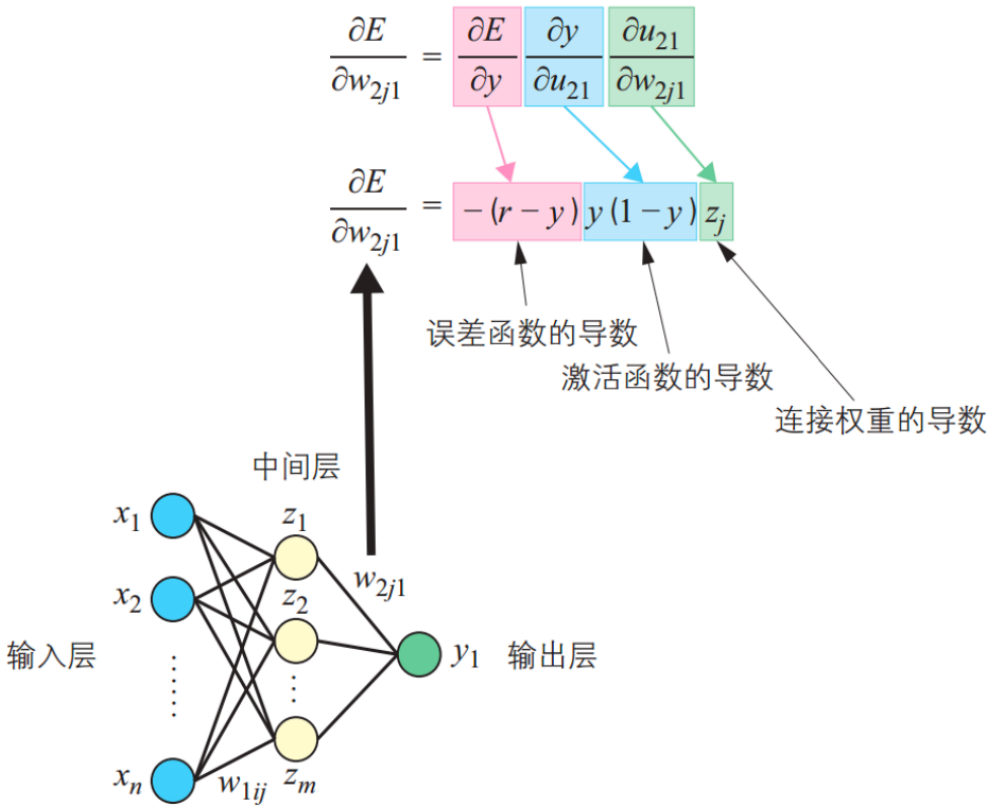
- 第二，中间层到输出层的连接权重调整值如下所示：

$$\Delta w_{2j1} = \alpha (r - y) y (1 - y) z_j$$

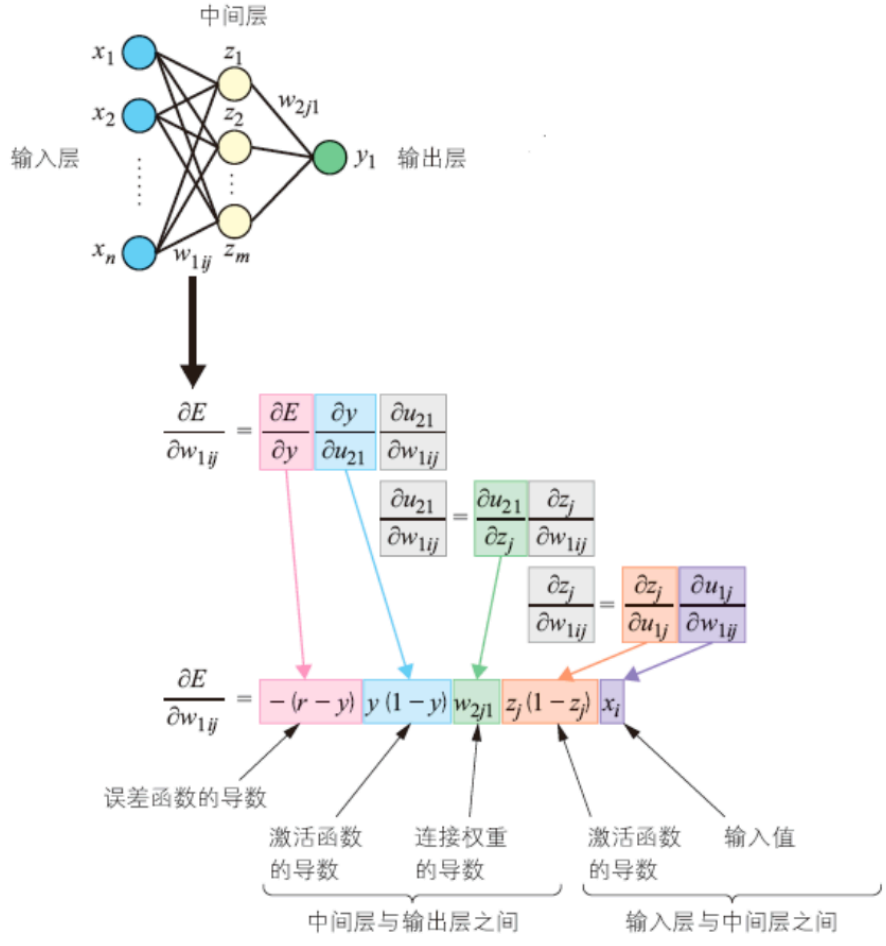
- 第三，调整输入层与中间层之间的连接权重

$$\begin{aligned} \frac{\partial E}{\partial w_{1ij}} &= \frac{\partial E}{\partial y} \frac{\partial y}{\partial u_{21}} \frac{\partial u_{21}}{\partial w_{1ij}} \\ &= -(r - y) y (1 - y) \frac{\partial u_{21}}{\partial w_{1ij}} \end{aligned}$$

中间层到输出层



输入层到中间层



前馈神经网络

参数学习

计算图与自动求导

优化