

# 第七章 人工神经网络ANN

## 一、神经网络

受到人脑结构的启发，神经元是生物大脑的基本构成单元，感知器是一个整合输入信号处理得到输出信号的设备，在传统的人脑中里面有上百万的神经元连接在一起，它们之间存在权重的关系

每个神经元只有两种状态活跃或者非活跃(取值为0或者1)，代表神经元的兴奋或者抑制，这个状态可以用MP函数表示

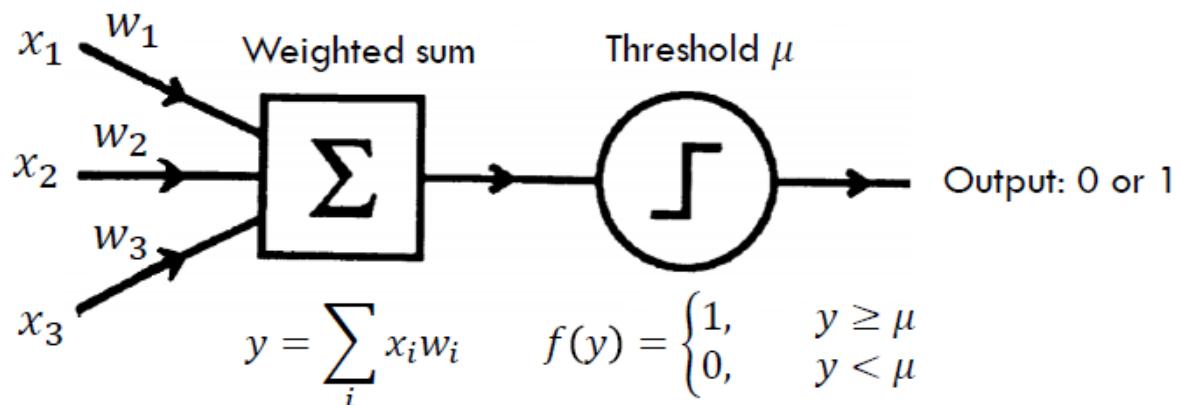
$$S_i = f(\sum_j \omega_{ij} S_j + \theta_i)$$

---

## 二、人工神经元

### 1. 神经元传输过程

输入——>线性聚合——>转移——>输出



一般来说输入数值会被归一化为[0,1]区间内

### 2. 为什么需要激活函数

- 如果没有激活函数，权重和偏置只有线性变换，那么这个神经网络就仅仅只是一个线性模型，一个一次多项式，对于解决复杂问题具有限制性
- 此外激活函数具有可微性，所以他们可以轻易进行反向传播，采用优化策略，当反向传播的时候去测量神经网络中的梯度损失函数

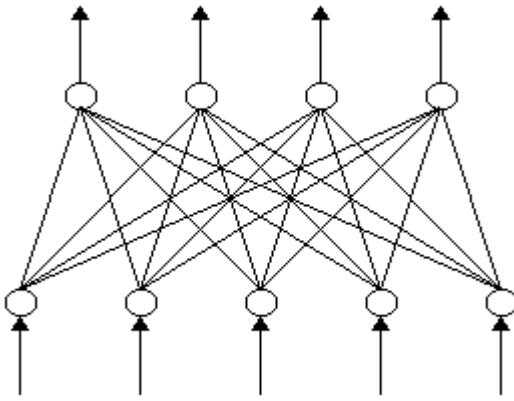
常见的激活函数有Sigmoid函数 $f(x) = \frac{1}{1+e^{-x}}$

以及双曲正切函数 $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

---

## 三、单层感知机

## 1. 单层感知机计算流程



单层感知机可以看做是一个线性分类器，这个可以看成公式

$$y = f\left(\sum_{i=1}^n \omega_i x_i + b\right)$$

其中 $f(\cdot)$ 代表阶跃函数，当输入大于0，输出1，输入小于等于0时候输出0

一般来说权重初始化时候设定权重 $\omega_i$ 随机初始化，偏置 $b$ 初始化0

后续反向传播时候有以下公式

$$\begin{aligned}\omega_i^{new} &= \omega_i + (y_{true} - y_{pred})x_i \\ b^{new} &= b + (y_{true} - y_{pred})\end{aligned}$$

根据上述的反向传播公式不断迭代更新权重，直到所有的样本都被正确分类

## 2. 单层感知机缺陷

如果在一个平面或者一个超空间中，存在一条直线或一个超平面可以正确分类所有样本，那么就可以称这些样本是线性可分的。**感知机只能对线性可分的样本进行分类**，对于非线性的不会收敛。例如对于异或问题，该模型就永远不会收敛

---

## 四、什么是人工神经网络

- 一个网络由多个简单地神经元节点构成，这些节点相互连接，每个节点既是信息存储单元也是信息处理单元
- 每个节点都对应一个权重
- 信息会沿着这些节点的连接传递，随着与权重的运算而改变
- 人工神经网络可以看做为一个有向图
- 通过调整连接的权值来解决特定任务
- 神经网络可以包含中间层，中间层的存在使得其可以识别更多的模式
- 一个神经网络可以产生多个输出

## 五、反向传播算法

### 1. 反向传播算法的主要步骤

- 构建一个网络：其中包含输入数据的表示，以及网络的层数和每一层网络的节点数

- 使用训练数据训练神经网络：正向传播和反向传播

- 调整网络

训练的最终目标：获取一组权重使得训练数据的所有样本元组被分类正确

## 2. BP算法的原理：

- 前向传播：原始输入被输入进输入层，输入和对应的权重会被传递到隐藏层，隐藏层对收到的数据进行计算，对于输入数据的加权总和，激活函数会应用到隐藏层的每一个节点，最后每一个节点的加权输出会传入输出层，从而计算最终的预测结果，跟隐藏层的计算过程一样
- 反向传播：误差会传回网络通过学习和调整内部的权重来改进性能，大多数方法使用均方误差来计算预测值和真实值之间的差距，一旦我们前向传播后我们就可以逐层反向传播误差到这个神经网络中

反向传播的关键是计算每个权重和偏置的梯度值，这个梯度值告诉我们每一个权重和偏置需要进行多大的调整来最小化误差，根据链式法则可以有效计算梯度。这里需要有一个学习率定义了学习调整的步长

反向传播的过程就是根据链式法则分别计算每个节点的权重的梯度，然后把这些权重按照学习率在梯度上进行更新

## 3. Sigmoid函数的问题：

梯度消失，随着时间步的推移，梯度会变得特别小

为什么梯度消失问题那么严重：在反向传播过程中，当梯度变得过小的时候，优化器就难以更新参数，从而导致模型收敛速度缓慢或无法收敛。这会使模型难以从数据中学习，从而产生错误的结果。

## 六、Softmax函数和交叉熵损失函数

### 1. Softmax函数的作用：

将原始得分向量转化为概率分布，这对针对分类的神经网络最后一层非常有用

对输入的元素进行指数运算，并且归一化，最后输出得到0~1之间的概率分布：对于

$v(x_1, x_2, \dots, x_N)$

$$\text{softmax}(v) = \left( \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}} \right), i = 1, 2, \dots, N$$

### 2. 交叉熵损失函数

$$H(p, \hat{p}) = - \sum_x p(x) \log \frac{1}{\hat{p}(x)} = - \sum_x p(x) \log((x))$$

其中 $(x)$ 是最终得到的预测概率， $p(x)$ 是真实标签，如果对应真实标签就是1

交叉熵损失函数大于等于熵，额外的信息被称为相对熵

交叉熵可以用来计算预测分布和数据分布之间的差异，交叉熵可以用来衡量在给定的真实分布下，使用非真实分布策略消除系统不确定性所需要付出的努力大小

- 熵：用于计算系统内随机性或无序程度，以衡量事件的不确定性。如果某个结果是确定无疑的，那么熵的数值就会较低。

- 交叉熵：也被称为对数损失或对数误差，是机器学习中一种常用的损失函数，用于衡量分类模型的性能。具体而言，它衡量的是分类模型所发现的概率分布与预测值之间的差异。
- 

## 七、相关问题

### 1. 学习率

如果学习率过大梯度下降会越过损失最小值，并且呈现发散

如果梯度过小算法可能需要更多的轮次去收敛或者陷入局部最优

### 2. 隐藏层的数量

隐藏单元的数量M（包括权重和偏置）是一个自由参数

经过调整以获得最佳的预测性能

确定M的一种方法是采用交叉验证。

还有其他一些方法（如正则化）可以用来控制神经网络的复杂度，以避免出现过拟合现象。

### 3. 神经网络节点的作用

隐藏层节点将非线性分离的样本转换为可线性分离的样本。

输出层节点将线性可分的样本分类为两类。