神经网络

Created	@March 25, 2022 12:03 AM
Class	DeepLearning
Type	Study Group
Materials	
Reviewed	☐
Property	NN
Status	Done 🙌

神经元模型(M-P神经元）

定义：神经网络是具有适应性的简单单员组成的广泛并行互连的网络，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交互反应。

神经元模型指的是简单单元，也就是M-P神经元模型。神经元接收到来自n个其他神经元传递过来的输入信号，这些输入信号通过**带权重**的连接（connection）进行传递，神经元

接收到的总输入值将与神经元的**阈值θ**进行比较，然后通过“激活函数”（activation function）处理以产生神经元的输出。

$$y = f(\sum_{i=1}^{n} w_i x_i - \theta) = f(w'x + b)$$

单个M-P神经元：感知机（sgn作激活函数）、对数几率回归（sigmoid作激活函数）

多个M-P神经元：神经网络

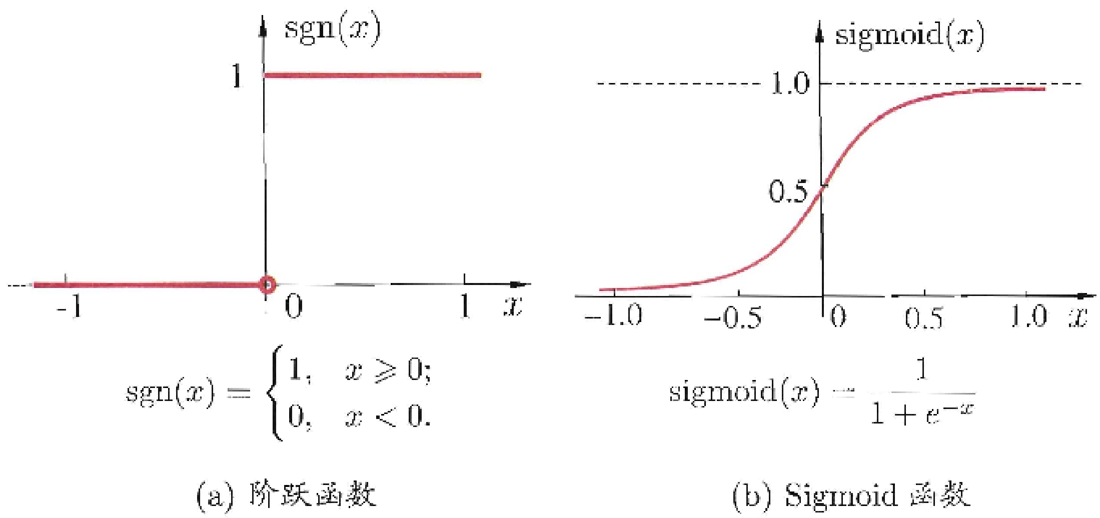


图 5.2 典型的神经元激活函数

阶跃函数sng(x)：输入值映射为输入值0和1.其中1-对神经元兴奋，0-对神经元抑制。

sigmoid函数：把可能在较大范围内变化的输入值挤压到（0，1）输入值范围内，因此也称为“挤压函数”（squashing function)

感知机与多层网络

感知机

两层神经元组成，输入层接受外界信号，输出层是M-P神经元。——“阈值逻辑单元“（threshold logic unit)

- 从纯数学的角度来看：

感知机模型：激活函数sgn的神经元

$$y = sgn(w'x - \theta)$$

x 为样本的特征向量，是感知机模型的输入，ω，θ是感知机模型的参数，ω是权重，θ是阈值。

- 从几何的角度来看：

给定一个线性可分的数据集T，感知机的学习目标是求得能对数据集T中的正负样本完全正确划分的超平面，其中ω’x-θ即为超平面方程。

线性不可分：找不到一条直线可以正确划分数据集正负样本。

超平面的性质：

1. 超平面方程不唯一
2. 法向量ω垂直于超平面
3. **法向量ω**和**位移项b**确定一个唯一超平面
4. 法向量ω指向的那一半空间为正空间，另一半为负空间。

感知机学习策略：随机初始化ω,b，将全体训练样本带入模型找到误分类样本，假设此时误分类样本集合M∈T，对任意一个误分类样本(x,y)∈M来说，ω‘x-θ≥0时，模型输出值为1，样本真实标记为0；反过来，模型输出值为0，样本真实值为1。

所以，给定数据集T，损失函数可以定义为：

$$L(\omega,\theta) = \sum_{x \in M} (\hat{y} - y)(w'x - \theta)$$

可以知道，损失函数是非负的。如果没有误分类点，损失函数值为0，误分点越少，误分点离超平面越近，损失函数的值就会越小。

怎么优化损失函数？极小化损失函数的解可以定义为：

$$min_{\omega,\theta} L(\omega,\theta) = min_{\omega,\theta} \sum_{x_i \in M} (\hat{y}_i - y_i)(\omega' x_i - \theta)$$

其中，$M \in T$为误分类样本集合。阈值θ可看作一个固定输入为 -1 的"哑结点”（dummy nodes），也就是$-\theta = -1 * \omega_{n+1} = x_{n+1} * \omega_{n+1}$。那求解极小化问题就可以简化为

$$min_{\omega} L(\omega) = min_{\omega} \sum_{x_i \in M} (\hat{y}_i - y_i)\omega' x_i$$

感知机学习算法：采用的是随机梯度下降。随机梯度下降是选取一次误分类点使其梯度下降。

$$\Delta\omega = -\eta(\hat{y}_i - y_i)x_i = \eta(y_i - \hat{y}_i)x_i$$

这里，η是学习率。通常解答出来的ω不唯一。

💡 注意：感知机只有输出层神经元进行激活函数处理，也就是只拥有一层功能神经元（functional neuron）。

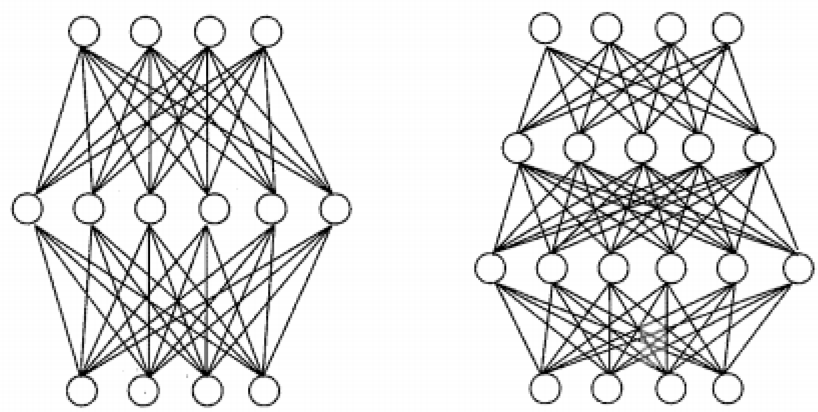
如果是非线性可分的问题，就要考虑多层功能神经元。也就是在输出层和输入层之间，还有隐含层（hidden layer），且隐含层和输出层神经元都是拥有激活函数的功能神经元。

神经网络

💡 多层前馈神经网络：每层神经元与下一层神经元全互连，神经元之间不存在同层/跨层连接，这种神经网络就是多层前馈神经网络（multi-layer feed forward neural）

没有神经网络之前，决定模型的上限的两个维度：数据量和特征，模型数据量大，训练的越好；特征越多性能越好，特征构造的越好，训练出来的模型就越重要。但是特征工程是需要一定的专业背景，需要有一定的专业知识才能更好的选择特征，构造特征。神经网络不需要构造特征，只要把观测到所有的特征都放进去学习，神经网络通过一些加工或组合去提取一些新的特征。

> 通用近似定理（后面补笔记）



(a) 单隐层前馈网络　　(b) 双隐层前馈网络

图 5.6　多层前馈神经网络结构示意图

神经网络记为NN，可以看作是一个特征加工函数。$x \in R^d \rightarrow NN(x) \rightarrow y = x^* \in R^l$.

单输出就是回归任务，输出层$R^l \rightarrow R$的神经元，比如没有激活函数的神经元：

$$y = \omega' x^* + b$$

对于分类任务，输出层$R^l \rightarrow [0, 1]$的神经元，比如：激活函数为sigmoid函数的神经元：

$$y = \frac{1}{1 + e^{-(\omega' x^* + b)}}$$

模型训练过程中，NN自动学习提取有用的特征，所以说机器学习向“全自动数据分析”又前进了一步。

回归任务中损失函数可以采用均方误差，分类任务则用交叉熵。

误差逆传播算法(BP算法）

基于随机梯度下降的参数更新算法：

$w_{\leftarrow}\omega + \Delta\omega$

$\Delta\omega = -\eta\Delta_{\omega}E$

只需推导出$\Delta_{\omega}E$这个损失函数E关于参数ω的一阶偏导数就可以。随机梯度下降不能保证一定能走到全局最小值点，更多情况下走到的是局部极小值点。

> 💡 “跳出”局部极小

- 以多组不同参数值初始化多个神经网络, 按标准方法训练后, 取其中误差最小的解作为最终参数. 这相当于从多个不同的初始点开始搜索, 这样就可能陷入不同的局部极小, 从中进行选择有可能获得更接近全局最小的结果.
- 使用“模拟退火” (simulated annealing) 技术 [Aarts and Korst, 1989]. 模拟退火在每一步都以一定的概率接受比当前解更差的结果, 从而有助于“跳出”局部极小. 在每步迭代过程中, 接受“次优解”的概率要随着时间的推移而逐渐降低, 从而保证算法稳定.
- 使用随机梯度下降. 与标准梯度下降法精确计算梯度不同, 随机梯度下降法在计算梯度时加入了随机因素. 于是, 即便陷入局部极小点, 它计算出的梯度仍可能不为零, 这样就有机会跳出局部极小继续搜索.

其他常见神经网络

1. RBF网络

2. ART网络
3. SOM网络
4. 级联相关网络
5. Elman网络
6. Boltzman网络