

人工智能

早期的人工智能令人兴奋不已



机器学习

机器学习开始兴起



深度学习

深度学习取得突破 驱动人工智能蓬勃发展



50's 1960's

1970's

's 198

1980's

1990's

2000's

2010's



Tensor: 张量,数据结构,类多维数组,包含三个属性: 名称,维度,类型

Flow: 流,直观表达张量之间转化的过程

Tensorflow: 使用节点图来表示计算的编程系统

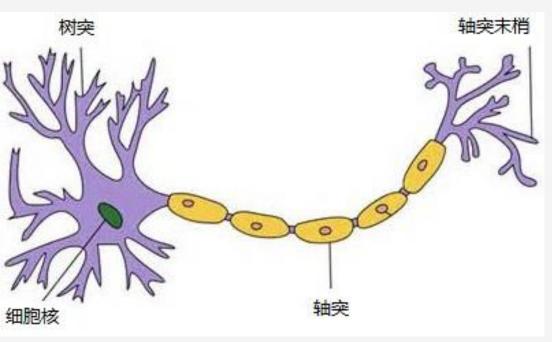
张量运算生成节点,节点包含计算参数,类型,维度信息

Tensorboard: 可以使神经网络可视化的工具

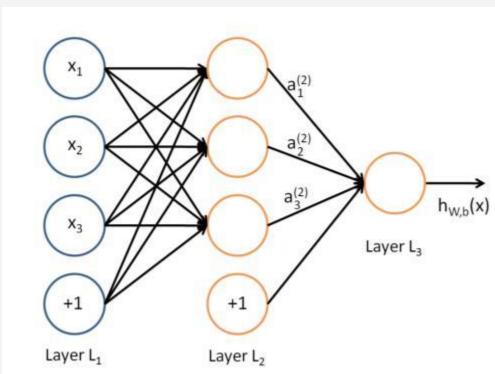
从神经网络发展了解深度学习

Neural Networks

人体大脑神经网络

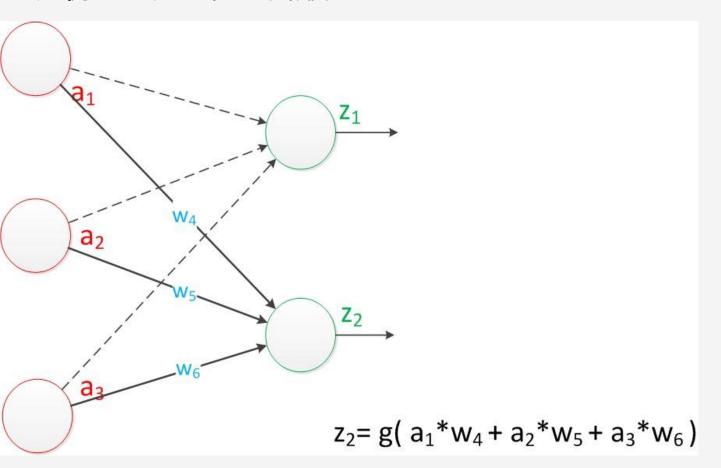


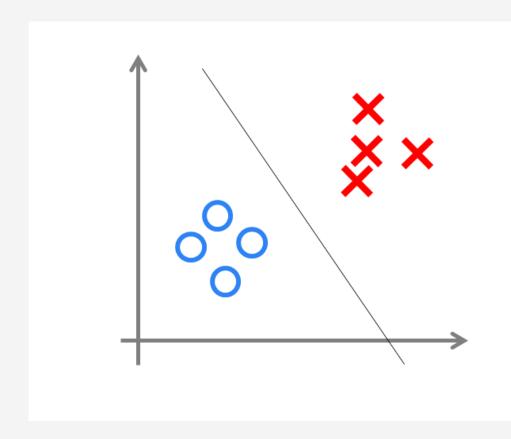




Neural Networks

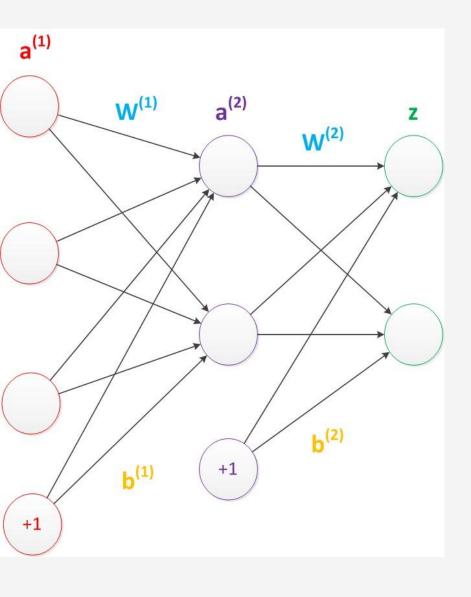
感知机——单层神经网络模型





多层感知机——双层神经网络模型

Neural Networks



加入一层隐藏层

层数=深度

$$g(\mathbf{W}^{(1)} * \mathbf{a}^{(1)} + \mathbf{b}^{(1)}) = \mathbf{a}^{(2)}$$

 $g(\mathbf{W}^{(2)} * \mathbf{a}^{(2)} + \mathbf{b}^{(2)}) = \mathbf{z}$

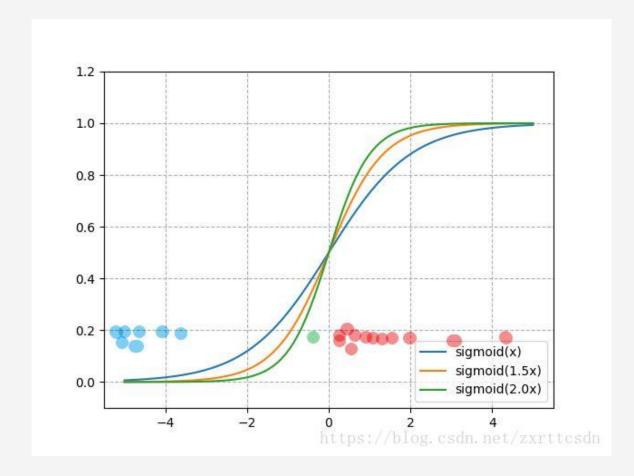
g为激活函数

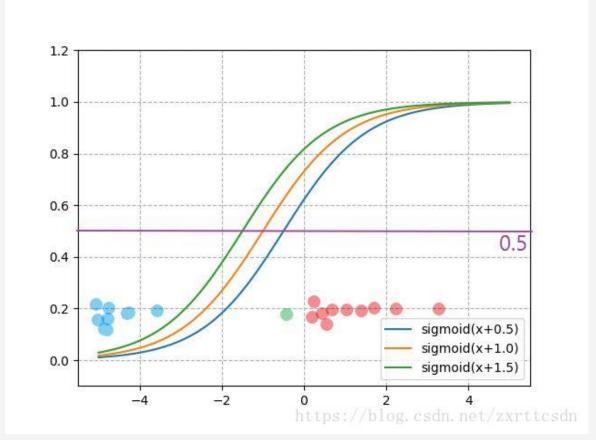
Sigmoid:
$$g(x) = \frac{1}{e^{-x} + 1}$$

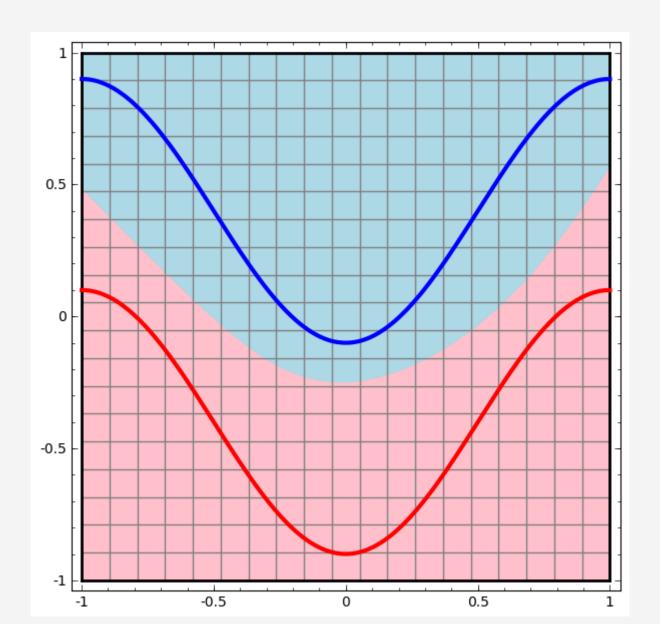
tanh:
$$g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

ReLU:
$$g(x) = MAX(0, x)$$

Bias









Model Train

神经网络的学习,其实就是通过大量有标签数据去训练,找到最合适的权值,从而使模型达到最低的错误率

样本的预测目标为y_p,真实目标为y。那么,定义一个值loss,计算公式如下:

$$\mathbf{loss} = ||y_P - y||^2$$

Loss Function

欧式距离: $||y_P - y||^2$ 模型是不佳的

引入Softmax + cross Entropy

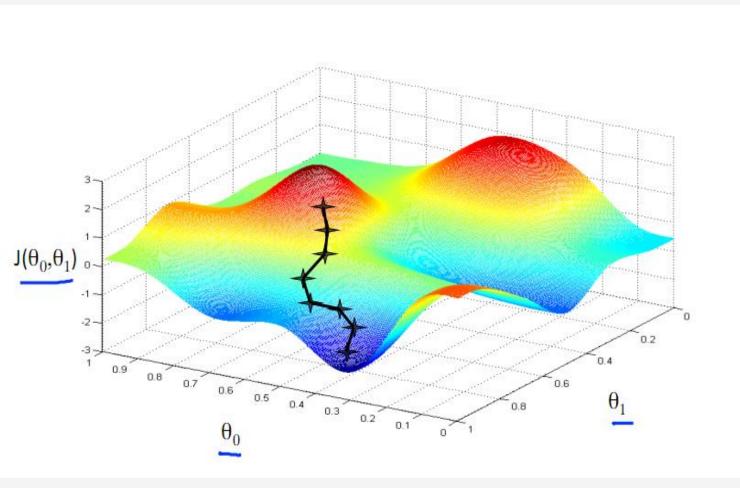
Softamx:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$
 for $j = 1, ..., K$.

cross Entropy: $H(y_p, y) = -(y \log y_p + (1 - y) \log(1 - y_p))$

Optimize

Gradient Descent



梯度实际上就是多变量微分的一般化。

- 在单变量的函数中,梯度其实就 是函数的微分,代表着函数在某 个给定点的切线的斜率
- 在多变量函数中,梯度是一个向量,向量有方向,梯度的方向就 指出了函数在给定点的上升最快的方向

Optimize

Learning rate: 学习率,控制参数的更新速度

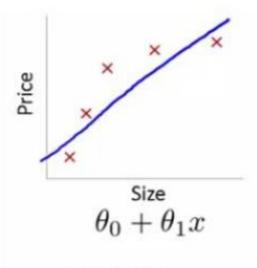
表 4-2 当学习率过大时,梯度下降算法的运行过程

轮数	当前轮参数值	梯度×学习率	更新后参数值
1	5	2×5×1=10	5-10=-5
2	-5	2× (-5) ×1=-10	-5- (-10) =5
3	5	2×5×1=10	5-10=-5

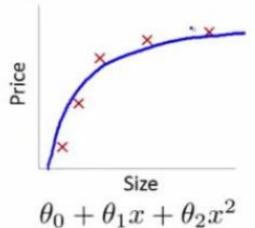
引入指数衰减法,使得Learning rate逐步降低,在初期较大学习率 找到合适解决值,再通过后期低速学习来调整模型

Optimize •

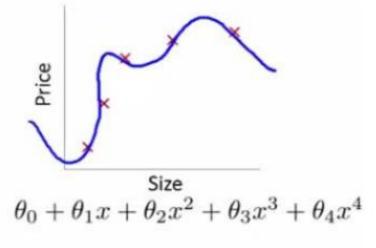
Over-fit



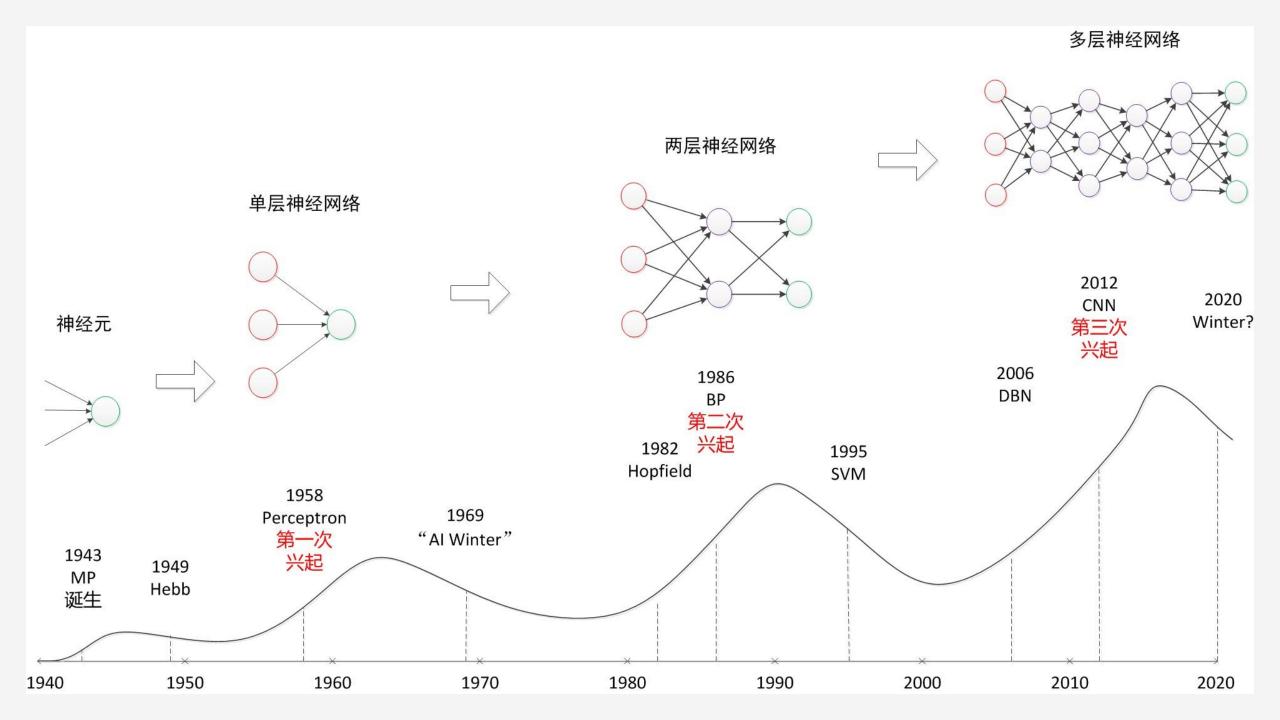
High bias (underfit)



"Just right"



High variance (overfit)



手写识别 MNIST

手写体识别,一个经典的深度学习入门项目

有浅显的的神经网络结构

数据预处理:

图片二维像素矩阵转化为数组,取值【0,1】,0表示白色背景,1表示黑色前景

数据量很大,不可能一次进入模型,要将数据进行分割,每次读取一小部分称为batch

