

Notes: Neural Networks

2020 年 4 月 26 日

目录

1 感知机 Perceptron	3
2 DNN 反向传播	3
2.1 要解决的问题	3

1 感知机 Perceptron

前提：数据是线性可分的！

点到平面的距离公式： $d = \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}}$

输入： m 个样本即 $(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}, y_0), (x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}, y_1), \dots, (x_1^{(m)}, x_2^{(m)}, \dots, x_n^{(m)}, y_m)$ ，
标签 y 是二元类别。

目标：找到一个超平面 (hyperplane) 即 $\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n = 0$ ，让其中一种类别的样本都满足 $\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n > 0$ ；而另一种类别的样本都满足 $\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n < 0$ ，从而线性可分！简化写法：增加一个特征 $x_0 = 1$ ，所以有超平面 $\sum_{i=0}^n \theta_i x_i = 0$ 向量表示：

$$\theta_{(n+1) \times 1} \cdot x_{1 \times (n+1)} = 0$$

感知机模型定义：

$$y = \text{sign}(\theta \bullet x)$$
$$\text{sign}(x) = \begin{cases} -1 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

Gram 矩阵 (可以看成没有减去均值的协方差矩阵), 感知机原始形式以及对偶形式

2 DNN 反向传播

2.1 要解决的问题

在监督学习中往往要利用一个 loss function 来度量当前模型从输入得到的结果与真实值的误差。通过最小化这个误差，把整个模型往最好的方向调整即得到最好的 weight 和 b；而最小化的方法便有梯度下降，牛顿法，拟牛顿法等...