Notes: Neural Networks

2020 年 6 月 1 日

目录

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 $\left(x_1^{(0)}, x_2^{(0)}, \ldots x_n^{(0)}, y_0\right), \left(x_1^{(1)}, x_2^{(1)}, \ldots x_n^{(1)}, y_1\right), \ldots \left(x_1^{(m)}, x_2^{(m)}, \ldots x_n^{(m)}, y_m\right),$
标签 y 是二元类别。
目标：找到一个超平面 (hyperplane) 即 $\theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n = 0$，让其中一种类别的样本都满足 $\theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n > 0$；而另一种类别的样本都满足 $\theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n < 0$，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:

$$y = \operatorname{sign}(\theta \bullet x)$$

$$\operatorname{sign}(x) = \begin{cases} -1 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

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...

3 RNN BPTT(back propagation through time)

模型定义参考刘建平 RNN 推导

3.1 前向传播

1. 对于任意一个序列索引号 t：$h^{(t)} = \sigma\left(z^{(t)}\right) = \sigma\left(Ux^{(t)} + Wh^{(t-1)} + b\right)$，这里的 σ 是 tanh 函数
2. 序列索引号 t 时模型的输出 o^t 的表达式为：$o^{(t)} = Vh^{(t)} + c$，c b 一样，都是偏置向量
3. 最终在序列索引号 t 时预测输出 $\hat{y}^{(t)} = \sigma\left(o^{(t)}\right)$，这里的 σ 是 softmax 函数

3.2 反向传播

通过梯度下降法的一轮迭代，得到模型参数 U, W, V, b, c，**这些参数在序列的各个位置是共享的**，BP 时更新的是相同的参数！

由于在序列的每个位置都有损失函数 $L^{(t)} = -{y^{(t)}}^T log\hat{y}^{(t)}$，因此最终的损失 L 为：$L = \sum_{t=1}^{\tau} L^{(t)}$

下面的推导详见草稿纸笔记。。。