

lec07 Perceptron (Loss Function Minimisation)

Loss Function

Dataset: $\mathcal{D} = \{(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)\}$

Weight: $W = (w_0, w_1, \dots, w_d)$

Loss Function: $L(W, \mathcal{D})$

1. 根据 W 计算 Loss 值
2. 通过改变 W 最小化 $L(W, \mathcal{D})$, 得到新的权重 $W' = (w'_0, w'_1, \dots, w'_d)$
3. 根据 W' 计算 L , 返回步骤 2

Step Function

\mathcal{D} 中

$$X = (x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(d)})^T, \text{ for every } k = 1, \dots, n$$

$$a_k = b + \sum_{i=1}^d w_i x_k^{(i)}$$

对于单个训练对象 (X_k, y_k) , 定义损失函数为

$$L(b, W, X_k, y_k) = \begin{cases} 1, & \text{if } X \text{ misclassified} \\ 0, & \text{Otherwise} \end{cases}$$

对于一个训练集 \mathcal{D} , 定义损失函数为

$$L(b, W, X_k, y_k) = \sum_{k=1}^n L(b, W, X_k, y_k)$$

这是一个分段常数函数, 值为训练集中被错误分类的实例数量; 导数为 0, 不能使用梯度下降法

对于 $h(t) = \max(0, t)$:

对于单个训练对象 X_k , 定义损失函数为

$$L(b, W, X_k, y_k) = h(-y_k \cdot a_k)$$

对于一个训练集 \mathcal{D} , 定义损失函数为

$$L(b, W, \mathcal{D}) = L(b, W, X_k, y_k) = \sum_{k=1}^n h(-y_k \cdot a_k)$$

$$L = \begin{cases} 0, & \text{if } X_k \text{ 被分类正确} \\ -y_k \cdot a_k \geq 0 & \text{if } X_k \text{ 被分类错误} \end{cases}$$

Minimisation

对于损失函数

$$L(b, W, \mathcal{D}) = L(b, W, X_k, y_k) = \sum_{k=1}^n h(-y_k \cdot a_k)$$

使用梯度下降算法

$$\begin{pmatrix} b \\ w_1 \\ \vdots \\ w_d \end{pmatrix} \leftarrow \begin{pmatrix} b \\ w_1 \\ \vdots \\ w_d \end{pmatrix} - \mu \nabla_{b, w_1, \dots, w_d} L(b, W, \mathcal{D})$$

$$\nabla_{b, w_1, \dots, w_d} L(b, W, \mathcal{D}) = \sum_{k=1}^n \nabla_{b, w_1, \dots, w_d} L(b, W, X_k, y_k) = \sum_{k=1}^n \nabla_{b, w_1, \dots, w_d} h(-y_k \cdot a_k)$$

其中 $\nabla_{b, w_1, \dots, w_d} L(b, W, \mathcal{D})$ 表示自变量 b, w_1, \dots, w_d 在 L 上的偏导

计算 $\nabla_{b, w_1, \dots, w_d} h(-y_k \cdot a_k)$

对于 $h(t)$, 显然 $t < 0$ 时, $h'(t) = 0$; $t \geq 0$ 时, 虽然 h 在 $t = 0$ 时不可导, 但是我们可以在此处人为设置 $h'(0) = 1$, 所以 $h'(t) = 1$

则

$$h'(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

$$\frac{\partial h(-y_k \cdot a_k)}{\partial b} = h'(-y_k \cdot a_k) \cdot \frac{\partial}{\partial b} (-y_k \cdot a_k) = -y_k$$

$$\frac{\partial h(-y_k \cdot a_k)}{\partial w_i} = h'(-y_k \cdot a_k) \cdot \frac{\partial}{\partial w_i} (-y_k \cdot a_k) = -y_k \cdot x_k^{(i)}$$

$$\nabla_{b, w_1, \dots, w_d} h(-y_k \cdot a_k) = \begin{pmatrix} \frac{\partial h(-y_k \cdot a_k)}{\partial b} \\ \frac{\partial h(-y_k \cdot a_k)}{\partial w_1} \\ \frac{\partial h(-y_k \cdot a_k)}{\partial w_2} \\ \vdots \\ \frac{\partial h(-y_k \cdot a_k)}{\partial w_d} \end{pmatrix} = -y_k \cdot \begin{pmatrix} 1 \\ x_k^{(1)} \\ x_k^{(2)} \\ \vdots \\ x_k^{(d)} \end{pmatrix}$$

梯度下降算法:

$$\begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} \leftarrow \begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} + \mu \sum_{k=1}^n y_k \cdot \begin{pmatrix} 1 \\ x_k^{(1)} \\ x_k^{(2)} \\ \vdots \\ x_k^{(d)} \end{pmatrix}$$

Online Gradient Descent

核心思想: 每次错误分类后更新参数。上面的梯度下降算法一每次都需要计算整个数据集, 太慢了

对于单个错误分类实例 (X_k, y_k) , 更新量变为:

$$\begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} \leftarrow \begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} - \mu \nabla_{b, w_1, \dots, w_d} L(b, W, X_k \cdot y_k)$$

$$\begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} \leftarrow \begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} - \mu \nabla_{b, w_1, \dots, w_d} h(-y_k \cdot a_k)$$

Update Rule

对于单个被错误分类的实例 (X, y) ，它拥有的激活分数是 $a = b + \sum_{i=1}^d w_i x_i$ ，则权重按照如下方式更新：

$$\begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} \leftarrow \begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} - \mu \nabla_{b, w_1, \dots, w_d} h(-y \cdot a)$$

$$\nabla_{b, w_1, \dots, w_d} h(-y \cdot a) = -y \cdot \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}$$

$$\begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} \leftarrow \begin{pmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} + \mu \cdot y \cdot \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}$$