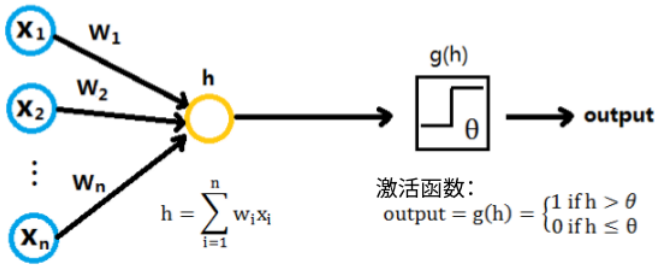
Contents

1 感知机模型

1.1 模型定义:



1.2 模型简化:

(i) output=0,即$\boldsymbol{h \leqslant \theta}$ 时:

$$
\begin{aligned}
&w_1x_1 + w_2x_2 + \cdots + w_nx_n \leqslant \theta \\
&w_1x_1 + w_2x_2 + \cdots + w_nx_n + (-\theta) * 1 \leqslant 0 \\
&w_1x_1 + w_2x_2 + \cdots + w_nx_n + w_0x_0 \leqslant 0 \\
&h = \sum_{i=0}^{n} w_ix_i \leqslant 0 \quad output = 0
\end{aligned}
$$

(ii) output=1,即$\boldsymbol{h > \theta}$ 时:

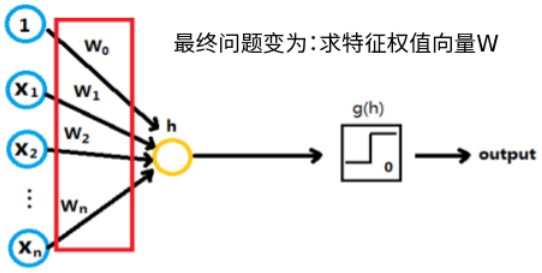
$$
\begin{aligned}
&w_1x_1 + w_2x_2 + \cdots + w_nx_n > \theta \\
&w_1x_1 + w_2x_2 + \cdots + w_nx_n + (-\theta) * 1 > 0 \\
&w_1x_1 + w_2x_2 + \cdots + w_nx_n + w_0x_0 > 0 \\
&h = \sum_{i=0}^{n} w_ix_i > 0 \quad output = 1
\end{aligned}
$$

(iii) 简化后的激活函数:

$$
output = g(h) = \begin{cases} 1 & if \quad h > 0 \\ 0 & if \quad h \leqslant 0 \end{cases}
$$

(iv) 简化后的特征权值:
$(-\theta) * 1 \Rightarrow w_0 x_0 \Rightarrow$增加偏移特征$x_0 = 1$的权重$w_0 = -\theta$

$$W = (w_0, w_1, w_2, \cdots, w_n)$$



1.3 模型输入:

(i) m组样本的特征向量矩阵:

$$\begin{Bmatrix} x_{10} = 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ x_{20} = 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{m0} = 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{Bmatrix}$$

(ii) m组样本的标注向量:

$$\begin{Bmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{Bmatrix}$$

1.4 模型计算:

(i) 如何调整特征权值: $(w_0, w_1, w_2, \cdots, w_n)$ 使得感知器的输出向量与样本标注向量全部相同?

$$\begin{Bmatrix} output_1 \\ output_2 \\ \cdots \\ output_m \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{Bmatrix}$$

(ii) 根据感知器判断错误的样本修改特征权值W [注: 标红部分]。

标注值Y	输出值OUTPUT
0	0
0	1
1	0
1	1

(iii) 若标注值$\boldsymbol{y=0}$,$\boldsymbol{output=1}$, 则说明

$$\boldsymbol{h=\sum_{i=0}^{n} w_i x_i > 0}$$

$$\boldsymbol{\sum_{i=0}^{n} w_i x_i - \triangle < 0}$$

$$\boldsymbol{\sum_{i=0}^{n} w_i x_i - \alpha * \sum_{i=0}^{n} x_i x_i < 0}$$

$$\boldsymbol{w_0 x_0 + w_1 x_1 + \cdots + w_n x_n - \alpha x_0 x_0 - \alpha x_1 x_1 - \cdots + \alpha x_n x_n < 0}$$

$$\boldsymbol{w_0 x_0 - \alpha x_0 x_0 + w_1 x_1 - \alpha x_1 x_1 + \cdots + w_n x_n - \alpha x_n x_n < 0}$$

$$\boldsymbol{x_0 (w_0 - \alpha x_0) + x_1 (w_1 - \alpha x_1) + \cdots + x_n (w_n - \alpha x_n) < 0}$$

$$\boldsymbol{\sum_{x=0}^{n} x_i (w_i - \alpha x_i) < 0}$$

故，特征权值的迭代方法为：

$$\boldsymbol{w_{i\_new} = w_i - \alpha x_i} \tag{1}$$

(iv) 若标注值$\boldsymbol{y=1}$,$\boldsymbol{output=0}$, 同理可得：

$$\boldsymbol{w_{i\_new} = w_i + \alpha x_i} \tag{2}$$

(v) 由（1）和（2）迭代算法为：

$$\boldsymbol{w_{i\_new} = w_i + \alpha (y_i - output_i) x_i}$$

1.5 接口设计：

(i) 激活函数接口：

$$\boldsymbol{h=\sum_{i=0}^{n} w_i x_i}$$

$$\boldsymbol{output = g(h) = \begin{cases} 1 & if \quad h > 0 \\ 0 & if \quad h \leqslant 0 \end{cases}}$$

```
int calc_activation(double x[], double weights[], int feature_num) {
    double dot = 0;
    for(int i = 0; i <= feature_num; i++) {
        dot += x[i] * weights[i];
    }
    if (dot > 0) {
        return 1;
    }
    return 0;
}
```

i从0开始，feature_num是实际样本特征不包含偏移特征x0

(ii) 训练函数接口：

$$w_{i\_new} = w_i + \alpha\left(y_i - output_i\right)x_i$$

```
void perceptron_train(double X[][MAX_FEATURE_DIMENSION], int y[], double weights[], int
feature_num, int sample_num, double alpha, int iterate_num) {

    for (int i=0; i<iterate_num; i++) {

        double delta[MAX_FEATURE_DIMENSION] = {0};
        for (int m=0; m<sample_num; m++) {
            //计算第m个样本的感知机输出output
            int output = calc_activation(X[m], weights, feature_num);
            //计算第m个样本的detal变化量 += 迭代速率*方向*特征值
            for (int n=0; n <= feature_num; n++) {
                delta[n] += alpha * (y[m] - output) * X[m][n];
            }
        }
        //每轮迭代后，更新特征权值向量，w_new = w_old + delta
        for (int n=0; n <= feature_num; n++) {
            weights[n] += delta[n];
        }
    }
}
```