

## 实验三 感知器准则算法实验（iris 数据）

### 一、实验目的

1. 进一步理解感知器算法的原理和实现方法。
2. 理解最小平方误差判别方法的原理和实现方法。
3. 掌握多类分类器的实现方法

### 二、实验内容及要求

1. 实验数据：iris 数据，分为三种类型：分别为 w1, w2 和 w3 类，每种类型中包括 50 个四维的向量，各类别出现的概率相等。

#### 2 实验要求

- 1) 从 iris 数据的每个样本中取出三个特征作为分类特征，并且将样本点画出；
- 2) 从每个类别的数据中抽取 45 个样本作为训练样本，5 个样本作为测试样本，
- 3) 用感知器批处理的方法实现 w1 类和 w2 类之间，w2 类和 w3 类，w1 类和 w3 类之间分类器的设计，写出判别函数，画出分类面，并记录收敛的次数。
- 4) 用感知器单步处理的方法实现 w1 类和 w2 类之间，w2 类和 w3 类，w1 类和 w3 类之间分类器的设计，写出判别函数，画出分类面，并记录收敛的次数。
- 5) 对于前面用感知器算法无法迭代出权向量的类别，使用最小平方误差判别的方法求出权向量，写出判别函数，画出分类面，并记录迭代次数。
- 6) 用多类分类器的逐步修正的方法对三个类别进行分类，写出每个类别的判别函数，画出分类面。
- 7) 将测试样本分别应用在分类器上，对测试样本进行判别，将判别结果进行显示。
- 8) 使用 python 语言来完成实验

### 三、实验原理

感知器是一种神经网络模型,对于两类线性可分的模式类 w1 和 w2, 首先对样本进行规范化处理,即将 w2 类的全部样本都乘以(-1), 这样对于两类的所有模式样本, 判别函数性质描述为:

$$d(y) = a^T y > 0$$

其中,  $a = [a_1, a_2, \dots, a_n, a_{n+1}]^T$ ,  $y = [x_1, x_2, \dots, x_n, x_{n+1}]^T$

感知器算法通过对已知类别的训练样本集的学习, 寻找一个满足上式的权向量。具体步骤如下:

1. 选择  $N$  个分属于  $w_1$  和  $w_2$  类的模式样本构成的训练样本集, 将训练样本写成增广向量的形式, 并进行规范化处理。

2. 用全部训练样本进行一轮迭代。每输入一个样本  $X$ , 计算一次判别函数  $a^T X$ , 在一轮样本全部计算之后, 若结果中有小于或等于零的情况, 此时要对权向量进行修正, 并且迭代次数加 1。

假设进行到第  $k$  次迭代时, 输入的样本中有  $m$  个样本的判别函数小于或等于零, 说明这  $m$  个样本被错分, 这时要对权向量进行更新, 即用  $m$  个错分样本值和对权向量进行修正:

$$a(k+1) = a(k) - \rho_k \nabla J = a(k) + \rho_k \sum_{y \in Y_e^k} y$$

3. 分析结果, 在这一轮的迭代中只要有一个样本的分类发生了错误, 则回到步骤 (2) 进行下一轮迭代, 用全部样本再训练一次, 建立新的  $a(k+1)$ , 直至用全部样本进行训练都获得了正确的分类结果, 迭代结束。这时的权向量值即为算法结果。

逐步修正法可以对多类进行分类, 对样本进行增广化处理后, 采用与感知器算法类似的单样本修正法来求解线性分类器, 给每一类样本一个判别函数, 算法参见教材 80-81 页。

#### 四、实验程序及结果: