

第3次作业 (11-15讲)

1. 样本集分析

$$X_1 = [2, 2]^T, \quad X_2 = [3, 2]^T, \quad X_3 = [3, 3]^T, \quad X_4 = [4, 2]^T$$

(1) 类内离散度矩阵 (S_W)

计算样本均值：

$$m = \frac{1}{4} \sum_{i=1}^4 X_i = \frac{[12, 9]^T}{4} = [3, 2.25]^T$$

计算中心化样本：

$$\begin{aligned} X_1 - m &= [-1, -0.25]^T, & X_2 - m &= [0, -0.25]^T \\ X_3 - m &= [0, 0.75]^T, & X_4 - m &= [1, -0.25]^T \end{aligned}$$

计算外积并求和：

$$S_W = \sum_{i=1}^4 (X_i - m)(X_i - m)^T = \begin{bmatrix} 2 & 0 \\ 0 & 0.75 \end{bmatrix}$$

(2) 特征值与特征向量

特征值：

$$\lambda_1 = 2, \quad \lambda_2 = 0.75$$

特征向量：

$$v_1 = [1, 0]^T, \quad v_2 = [0, 1]^T$$

(3) 一维模式变换

变换矩阵（投影方向）：

$$w = [1, 0]^T$$

一维投影结果：

$$\begin{aligned} y_1 &= 2, & y_2 &= 3, & y_3 &= 3, & y_4 &= 4 \\ \text{一维模式: } &\{2, 3, 3, 4\} \end{aligned}$$

2. 最短距离法层次聚类

样本数据：

$$x_1 = (0, 1, 3, 1, 3, 4)^T, \quad x_2 = (3, 3, 3, 1, 2, 1)^T, \quad x_3 = (1, 0, 0, 0, 1, 1)^T$$
$$x_4 = (2, 1, 0, 2, 2, 1)^T, \quad x_5 = (0, 0, 1, 0, 1, 0)^T$$

初始距离矩阵（欧氏距离）：

$$\begin{bmatrix} 4.80 & 5.00 & 4.90 & 5.10 \\ & 4.90 & 3.87 & 5.00 \\ & & 2.65 & 1.73 \\ & & & 3.46 \end{bmatrix}$$

聚类过程：

1. 步骤1：最小距离 $d_{35} = 1.73$, 合并 $\{x_3, x_5\}$

◦ 新距离矩阵（类：35, 1, 2, 4）：

$$\begin{bmatrix} 5.00 & 4.90 & 2.65 \\ & 4.80 & 4.90 \\ & & 3.87 \end{bmatrix}$$

2. 步骤2：最小距离 $d_{(35)4} = 2.65$, 合并 $\{x_3, x_5, x_4\}$

◦ 新距离矩阵（类：354, 1, 2）：

$$\begin{bmatrix} 4.90 & 3.87 \\ & 4.80 \end{bmatrix}$$

3. 步骤3：最小距离 $d_{(354)2} = 3.87$, 合并 $\{x_3, x_5, x_4, x_2\}$

◦ 新距离： $d_{(3542)1} = 4.80$

4. 步骤4：合并最后两个类

最终聚类顺序：

- 合并 3 与 5
- 合并 (3,5) 与 4
- 合并 (3,5,4) 与 2
- 合并 (3,5,4,2) 与 1

3. 神经网络模型比较

感知器神经网络

- **结构**: 单层/多层 (通常指单层)
- **激活函数**: 阶跃函数
- **特点**: 只能解决线性可分问题, 无隐藏层时无法处理非线性问题
- **学习算法**: 误差驱动学习 (感知器学习算法)

BP神经网络

- **结构**: 多层前馈网络 (输入层、隐藏层、输出层)
- **激活函数**: 可微函数 (Sigmoid、Tanh等)
- **特点**: 能处理非线性问题, 通过误差反向传播更新权重
- **问题**: 存在梯度消失/爆炸问题

ReLU vs Sigmoid

1. **梯度消失问题**: Sigmoid在饱和区梯度接近0, 导致深层网络训练困难; ReLU在正区间梯度为1, 缓解此问题
2. **计算效率**: ReLU计算简单 (比较和置零), Sigmoid需要指数运算
3. **稀疏激活**: ReLU使部分神经元输出为0, 增加稀疏性, 可能提升泛化能力
4. **改进版本**: Leaky ReLU、PReLU等解决"死亡神经元"问题

4. 神经网络与深度学习

相同点

- 基本单元都是神经元
- 采用前向传播和反向传播机制
- 可用于分类、回归、识别等任务
- 结构上都包含输入层、隐藏层、输出层

不同点

- **深度**: 深度学习特指多个隐藏层 (通常 >2 层), 传统神经网络可能只有一层隐藏层
- **特征学习**: 深度学习自动学习多层次特征表示, 传统神经网络常依赖人工特征提取
- **应用范围**: 深度学习在图像、语音、NLP等领域表现卓越, 神经网络更泛指各种结构

人脸识别

- **推荐模型：**卷积神经网络（CNN），如ResNet、FaceNet
- **原因：**CNN能有效捕捉图像的局部与空间层次结构，通过多层卷积提取人脸特征，适合处理图像数据，在大量数据下泛化能力强

天气预报

- **推荐模型：**循环神经网络（RNN），如LSTM、GRU，或Transformer
- **原因：**天气预报是时间序列预测问题，RNN/LSTM能建模时间依赖关系，Transformer可捕捉长期依赖，适合处理气象数据中的时序模式