人工智能导论
课程号：90111201
2023-2024 学年第一学期

作业 3&4

截止日期：2023.12.31

姓名: 陈 子 昂
学号: 221900197
成绩: ________

Problem 1 (3pt) . 考虑如下数据集，运行 K-Means 算法对其进行聚类，K 设置为 3，距离度量采用欧式距离，三个聚类的类中心分别初始化为：(6.2,3.2), (6.6,3.7), (6.5,3.0)。请给出在前三轮迭代过程中，三个聚类中心以及每个聚类所包含的样本的变化。

$$\mathbf{X} = \begin{bmatrix} 5.9 & 3.2 \\ 4.6 & 2.9 \\ 6.2 & 2.8 \\ 4.7 & 3.2 \\ 5.5 & 4.2 \\ 5.0 & 3.0 \\ 4.9 & 3.1 \\ 6.7 & 3.1 \\ 5.1 & 3.8 \\ 6.0 & 3.0 \end{bmatrix}$$

Solution: 记三个类中心依次为 A,B,C，用 1-10 依次为 X 中数据编号

第一次迭代：

A （5.17，3.17）样本：1、2、4、6、7、9、10

B （5.5，4.2）样本：5

C （6.45，2.95）样本：3、8

第二次迭代：

A （4.8，3.05）样本：2、4、6、7

B （5.3，4）样本：5、9

C （6.2，3.025）样本：1、3、8、10

第三次迭代：

A （4.8，3.05）样本：2、4、6、7

B （5.3，4）样本：5、9

C （6.2，3.025）样本：1、3、8、10 □

Problem 2 (2pt) . 考虑一个包含 4 个维度为 2 的样本的数据集：(-1,-1), (0.5,-0.5), (1,1), (-0.5,0.5)，采用 PCA 算法将其降维到 1 维，请计算降维后样本的方差。

Solution: 考虑一个包含四个维度为二的样本的数据集：

$$X = \begin{bmatrix} -1 & -1 \\ 0.5 & -0.5 \\ 1 & 1 \\ -0.5 & 0.5 \end{bmatrix}$$

步骤 1：标准化数据

数据集 X 的每个特征的均值 μ 和标准差 σ 分别为：

$$\mu = \frac{1}{n}\sum_{i=1}^{n} X_i = (0,0), \quad \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i-\mu)^2} = \sqrt{\frac{5}{8}}$$

标准化后的数据 X_{std} 是：

$$X_{\text{std}} = \frac{X-\mu}{\sigma}$$

步骤 2：计算协方差矩阵

协方差矩阵 C 为:

$$C = \frac{1}{n-1} X_{\text{std}}^{\top} X_{\text{std}} = \begin{bmatrix} \frac{4}{3} & \frac{4}{5} \\ \frac{4}{5} & \frac{4}{3} \end{bmatrix}$$

步骤 3:找到特征值和特征向量

C 的特征值 λ 和特征向量 v 满足:

$$Cv = \lambda v$$

步骤 4:选择主成分

选择具有最大特征值的特征向量作为主成分。

步骤 5:转换数据

数据 X 转换到新的空间 X_{trans}:

$$X_{\text{trans}} = X_{\text{std}} v = \frac{2}{\sqrt{5}} \begin{bmatrix} -2 \\ 0 \\ 2 \\ 0 \end{bmatrix}$$

步骤 6:计算降维后样本的方差

降维后样本的方差为:

$$\text{Var}(X_{\text{trans}}) = \frac{1}{n} \sum_{i=1}^{n} (X_{\text{trans},i} - \mu_{\text{trans}})^2 = 1.6$$

□

Problem 3 (5pt) . 在介绍支持向量机 (SVM) 的时候,我们提到,SVM 的基本思想是当存在多个线性分类超平面时,找到一个最大化分类间隔 (margin) 的超平面,请思考如果存在多个具有相同 margin 的分类超平面,应该如何处理?给出你的算法或优化目标

Solution: 对于具有相同 margin 的超平面:$y = \omega \cdot x + b$ 可以通过对下列方法选出一个最优的:

(1) $Minimize : \|w\|$

(2) 正则化处理

(3) 增加额外的优化目标

□

Problem 4 (5pt) . 在介绍深度神经网络时,我们提到阻碍神经网络变深的一个问题是梯度消失现象,请你谈一下对该问题的理解,并尝试讨论分析可能的缓解方法。

Solution: 梯度消失现象是指在深度神经网络的训练过程中,由于连续的链式求导,远离输出层的层的梯度变得非常小,几乎为零。这个问题通常出现在使用导数在输入值较大或较小时接近于零的激活函数的深度网络中,在多层反向传播过程中,梯度会逐渐减小,最终导致梯度消失现象的产生。

可行的解决办法:

(1):通过预训练 + 微调将大量参数进行分组,局部先找到较好的设置,然后再基于局部较优的结果进行全局寻优。

(2):采用其他不会出现导数在输入值较大或较小时接近于零的情况的激活函数。

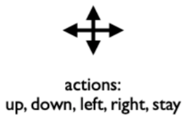
(3):在残差网络中使用的残差连接,引入跳跃连接,提高梯度的流动性。

(4):人为设定梯度的阈值。

□

Problem 5 (10pt) . 在下图所示 MDP 中, 表格代表了 9 个 state, 单元格内的数值表示达到这一状态能够得到的 reward, 假定执行动作状态转移是确定的，MDP 中的 $\gamma = 0.9$。请分别画出使用价值迭代和策略迭代前 5 轮每个 state 对应的状态值函数以及策略的变化情况。(提示：可参考 Lecturre16 幻灯片第 48 页的例子，每一轮同样画出表格，在对应的单元格内填上值函数和当前的最优策略。初始值函数可以全部设置为 0，初始策略可以自己指定。)

10	-1	-1
-1	-1	-1
-1	-1	-10



Solution: 用 U,D,L,R,S 依次代表动作中的 up,down,left,right,stay

价值迭代：

10	10	-1	S	L	L/D/S
10	-1	-1	U	U/D/L/R/S	U/L/S
-1	-1	-1	U/S/R	U/L/S	U/L

19	19	8	S	L	L
19	8	-1.9	U	U/L	U/L/S
8	-1.9	-1.9	U	U/L/S	U/L

27.1	27.1	16.1	U	L	L
27.1	16.1	6.2	U	U/L	U/L
16.1	6.2	-2.71	U	U/L	U/L

34.39	34.39	23.39	U	L	L
34.39	23.39	13.49	U	U/L	U/L
23.39	13.49	4.58	U	U/L	U/L

40.951	40.951	29.951	U	L	L
40.951	29.951	20.051	U	U/L	U/L
29.951	20.051	11.141	U	U/L	U/L

策略迭代：初始策略为向右移动，如果无法向右则留在原处

19	19	-1.9	S	L	L
19	8	-1.9	U	U	L
8	-1.9	-1.9	U	U	U

27.1	27.1	16.1	S	L	L
27.1	16.1	6.2	U	U	U
16.1	6.2	-2.71	U	U	U

34.39	34.39	23.39	S	L	L
34.39	23.39	13.49	U	U	U
23.39	13.49	4.58	U	U	U

40.951	40.951	29.951	S	L	L
40.951	29.951	20.051	U	U	U
29.951	20.051	11.141	U	U	U

46.856	46.856	35.856	S	L	L
46.856	35.856	25.956	U	U	U
35.856	25.956	17.046	U	U	U

□

Problem 6 (5pt) . 请给出一个可以应用强化学习的实际问题的例子，并建模其中的 MDP (Markov Decision Process) 四元组：描述出 state space, action space, transition function 各是什么，并设计一个 reward function?

Solution: 汽车的自动驾驶

state space：

- 车辆的位置和速度
- 周围车辆的位置、速度和方向
- 道路类型
- 天气情况

action space：

- 加速或减速
- 转向：左转、右转或保持直行
- 改变车道：向左或向右换道
- 启动和停车

transition function：

- 物理模型：汽车对加速、减速和转向的响应
- 环境模型：其他车辆的移动，交通信号灯和道路标志的变化

reward function：

- 发生碰撞或违反交通规则时 reward = reward - 100 如果一段时间内没有 reward = reward + 10
- 每成功行驶一定距离时 reward = reward + 5
- 如果一段时间内加速度变化的方差小于限定值 reward = reward + 2 否则 reward = reward - 1
- 美行驶一段距离的燃料消耗如果小于限定值 reward = reward + 2 否则 reward = reward - 1

□