

2020-2021 年度春夏学期浙江大学人工智能本科专业

《人工智能基础》课程实训题目

1. 斑马问题与八皇后问题（逻辑推理实验）

1.1. 实验背景

逻辑编程是一种编程典范，它设置答案须匹配的规则来解决问题，而非设置步骤来解决问题。过程是“事实+规则=结果”。人工智能的发展与逻辑编程的发展是一个相辅相成的过程，早期的人工智能以规则和逻辑推理作为主要研究方向，这在逻辑编程的发展中发挥了重要的影响，另外更好更快的逻辑编程也推动了人工智能的发展，例如专家系统、知识图谱和自动定理证明。

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言。在数据驱动学习时代，Python 的崛起已经是一个不争的事实，并且成为人工智能算法的第一语言。在本次实验中，我们学习将 Python 应用于逻辑编程，并尝试自主撰写逻辑规则解决斑马问题以及八皇后问题。

1.2. 实验内容

斑马问题：5 个不同国家且工作各不相同的人分别住在一条街上的 5 所房子里，每所房子的颜色不同，每个人都有自己养的不同宠物，喜欢喝不同的饮料。根据以下提示，你能告诉我哪所房子里的人养斑马，哪所房子里的人喜欢喝矿泉水吗？

- a) 英国人住在红色的房子里
- b) 西班牙人养了一条狗
- c) 日本人是一个油漆工
- d) 意大利人喜欢喝茶
- e) 挪威人住在左边的第一个房子里
- f) 绿房子在白房子的右边
- g) 摄影师养了一只蜗牛
- h) 外交官住在黄房子里
- i) 中间那个房子的人喜欢喝牛奶
- j) 喜欢喝咖啡的人住在绿房子里
- k) 挪威人住在蓝色的房子旁边
- l) 小提琴家喜欢喝橘子汁
- m) 养狐狸的人所住的房子与医生的房子相邻
- n) 养马的人所住的房子与外交官的房子相邻

八皇后问题：如何能够在 8×8 的国际象棋棋盘上放置八个皇后，使得任何一个皇后都无法直接吃掉其他的皇后？为了达到此目的，任两个皇后都不能处于同一条横行、纵行或斜线上。

1.3. 实验要求

- a) 基本掌握逻辑编程的思想，了解逻辑编程与命令式编程的区别
- b) 能够依据给定的事实以及规则编写代码，解决逻辑约束问题（CLP）

1.4. 实验环境

使用 Python 语言。

1.5. 参考资料

Python:

<https://www.runoob.com/python3/python3-tutorial.html>

kanren:

<https://github.com/logpy/logpy>

布置日期：2021 年 3 月 22 日

提交作业：2021 年 4 月 10 日晚上 12 点前

2. 黑白棋 (Mini AlphaGo)

2.1. 实验内容

黑白棋(Reversi), 也叫苹果棋, 翻转棋, 是一个经典的策略性游戏。一般棋子双面为黑白两色, 故称“黑白棋”。因为行棋之时将对方棋子翻转, 变为己方棋子, 故又称“翻转棋”(Reversi)。棋子双面为红、绿色的称为“苹果棋”。它使用 8x8 的棋盘, 由两人执黑子和白子轮流下棋, 最后子多方为胜方。随着网络的普及, 黑白棋作为一种最适合在网上玩的棋类游戏正在逐渐流行起来。中国主要的黑白棋游戏站点有 Yahoo 游戏、中国游戏网、联众游戏等。

游戏规则:

棋局开始时黑棋位于 e4 和 d5, 白棋位于 d4 和 e5, 如图所示。



- 1) 黑方先行, 双方交替下棋。
- 2) 一步合法的棋步包括: 在一个空格新落下一个棋子, 并且翻转对手一个或多个棋子。
- 3) 新落下的棋子与棋盘上已有的同色棋子间, 对方被夹住的所有棋子都要翻转过来。可以是横着夹, 竖着夹, 或是斜着夹。夹住的位置上必须全部是对手的棋子, 不能有空格。
- 4) 一步棋可以在数个(横向, 纵向, 对角线)方向上翻棋, 任何被夹住的棋子都必须被翻转过来, 棋手无权选择不翻某个棋子。
- 5) 除非至少翻转了对手的一个棋子, 否则就不能落子。如果一方没有合法棋步, 也就是说不管他下到哪里, 都不能至少翻转对手的一个棋子, 那他这一轮只能弃权, 而由他的对手继续落子直到他有合法棋步可下。
- 6) 如果一方至少有一步合法棋步可下, 他就必须落子, 不得弃权。
- 7) 棋局持续下去, 直到棋盘填满或者双方都无合法棋步可下。
- 8) 如果某一方落子时间超过 1 分钟, 则判该方失败。

2.2. 实验要求

- a) 使用 MCTS 算法实现 Mini AlphaGo for Reversi
- b) MCTS 算法部分需要自己实现, 尽量不使用现成的包, 工具或者接口
- c) 在博弈过程中, Mini AlphaGo 每一步所花费的时间以及总时间需要显示出来
- d) 需要有简单的图形界面用于人机博弈交互
- e) 使用 Python 语言

2.3. 参考资料

Python:

<https://www.runoob.com/python3/python3-tutorial.html>

3. 图像恢复重建（线性回归）

3.1. 实验内容

图像是一种非常常见的信息载体，但是在图像的获取、传输、存储的过程中可能由于各种原因使得图像受到噪声的影响。如何去除噪声的影响，恢复图像原本的信息是计算机视觉中的重要研究问题。

常见的图像恢复算法有基于空间域的中值滤波、基于小波域的小波去噪、基于偏微分方程的非线性扩散滤波等，在本次实验中，我们要对图像添加高斯噪声，并对添加噪声的图像进行基于线性回归模型的去噪。

3.2. 实验要求

A. 生成受损图像。

- a) 受损图像(X)是由原始图像($I \in \mathcal{R}^{H*W*C}$)添加了不同噪声遮罩(noise masks) ($M \in \mathcal{R}^{H*W*C}$)得到的($X = I \odot M$)，其中 \odot 是逐元素相乘。
- b) 噪声遮罩仅包含{0,1}值。对原图的噪声遮罩的可以 RGB 通道分别以 0.8/0.4/0.6 的噪声概率产生的，即噪声遮罩 RGB 通道的像素值各有 80%/40%/60%的概率值为 0，否则为 1。

B. 使用区域二元线性回归模型，进行图像恢复。

C. 评估误差为所有恢复图像(R)与原始图像(I)的 2-范数之和，此误差越小越好。 $error = \sum_{i=1}^3 \text{norm}(R_i(:) - I_i(:), 2)$ ，其中(:)是向量化操作。

3.3. 实验环境

可以使用基于 Python 的 OpenCV 库进行简单的图像处理，实验要求内容请采用 Numpy/MindSpore 库进行相关数值运算。

3.4. 参考资料

OpenCV:

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html

Numpy:

<https://www.numpy.org/>

<https://www.runoob.com/numpy/numpy-tutorial.html>

MindSpore:

<https://www.mindspore.cn/docs/zh-CN/r0.7/index.html>

4. 特征人脸识别 (PCA)

4.1. 实验内容

特征脸 (eigenface) 是第一种有效的人脸识别方法, 通过在一大组描述不同人脸的图像上进行主成分分析 (PCA) 获得。

本次实验要求大家构建一个自己的人脸库: 大家可以选择基于 ORL 人脸库添加自己搜集到的人脸图像形成一个更大的人脸库, 要求人脸库中的每一张图像都只包含一张人脸且眼睛的中心位置对齐(通过裁剪或缩放, 使得每张人脸图像大小尺寸一致且人脸眼睛的中心位置对齐)。

在训练过程中, 首先要求计算平均脸, 然后将前 K 个特征脸保存下来, 利用这 K 个特征脸对测试人脸进行识别, 此外对于任意给定的一张人脸图像, 可以使用这 K 个特征脸对原图进行重建。

4.2. 实验要求

- a) 求解人脸图像的特征值与特征向量构建特征脸模型
- b) 利用特征脸模型进行人脸识别和重建, 比较使用不同数量特征脸的识别与重建效果

4.3. 实验环境

使用 Numpy 库进行相关数值运算。

4.4. 参考资料

Numpy:

<https://www.numpy.org/>

<https://www.runoob.com/numpy/numpy-tutorial.html>

5. 基于卷积神经网络的手写体识别（深度学习）

5.1. 实验内容

手写数字识别是学习深度学习基础的图像识别任务，也称作深度学习实验的“HelloWorld! ”。计算机通过手写体图片来识别图片中的字，与印刷字体不同的是，不同人的手写体风格迥异，大小不一，造成了计算机对手写识别任务的困难，此项目通过应用深度学习和 MindSpore 工具对 MNIST 手写数据集进行训练并建模。

5.2. 实验要求

- a) 了解如何使用 MindSpore 进行简单卷积神经网络的开发。
- b) 了解如何使用 MindSpore 进行简单图片分类任务的训练。
- c) 了解如何使用 MindSpore 进行简单图片分类任务的验证。

5.3. 实验环境

使用 Python 的 MindSpore 库。

5.4. 参考资料

Numpy:

<https://www.numpy.org/>

<https://www.runoob.com/numpy/numpy-tutorial.html>

MindSpore:

<https://www.mindspore.cn/docs/zh-CN/r0.7/index.html>

Mnist:

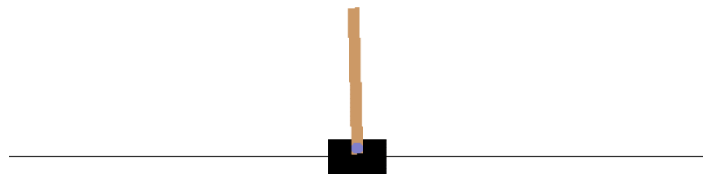
<http://yann.lecun.com/exdb/mnist/>

6. 深度 Q 函数学习

6.1. 实验内容

在 Q-learning 中，我们通过维护一张 Q 值表，通过贝尔曼(Bellman)方程对其进行不停的迭代尝试直至收敛，然后根据 Q 值表获取 Agent 在每个状态下的最优策略。但 Q 值表在状态和动作空间都是有限且低维的时候适用，当状态-动作空间高维且连续时，维护一张无限庞大的 Q 值表是不现实的。因此。DQN 提出将 Q-Table 的更新问题变成一个函数拟合问题，相近的状态将得到相近的动作输出，即使用神经网络对动作-状态的 Q 值进行建模估计。

OpenAI Gym 是一个用于开发和比较强化学习算法的工具包，在本次实验中，要求大家通过 DQN 算法来实现 CartPole-V0 游戏，游戏要求为：在一维无阻力的轨道上，有一辆小车，小车上连接有活动摇摆杆，摇摆杆将由竖直发生倾斜，玩家需要通过控制小车移动使小车上面的杆子保持平衡。当小车偏离中心 4.8 个单位，或杆的倾斜超过 24 度，任务失败。坚持时间越久，分数越高。



6.2. 实验要求

- d) 了解基础 Q-Learning 算法。
- e) 掌握贝尔曼(Bellman)方程意义及运算。
- f) 了解如何使用 Pytorch/MindSpore 实现 DQN 任务。

6.3. 实验环境

使用 Python 的 Pytorch/MindSpore 库。

6.4. 参考资料

Numpy:

<https://www.numpy.org/>

<https://www.runoob.com/numpy/numpy-tutorial.html>

MindSpore:

<https://www.mindspore.cn/docs/zh-CN/r0.7/index.html>

PyTorch 官网:

<https://pytorch.org/>

OpenAI Gym 白皮书:

<http://arxiv.org/abs/1606.01540>

7. 德州扑克（深度学习）

7.1. 实验内容

德州扑克是典型的不完全性信息博弈，主要分为双人、多人；有限注、无限注等四种类型。AI 算法在掌握德州扑克游戏规则的基础上，需要根据手上拿到的底牌、公共牌、所在位置、手握筹码、底池、对手行为（即当前局的对手策略）、对手模型（即对手的历史策略），从而在弃牌、跟注、加注等动作、筹码大小做出序列的决策选择。目标是赢得尽可能多的筹码。

建议选择 AAAI2020 论文 RLCard 作为基础开发平台，已经开发了 Deep Q-Network（DQN）、Neural Fictitious Self-Play（NFSP）、Counterfactual Regret Minimization（CRM）三个基础版本算法。请设计一个多人无限注版本的德州扑克算法程序，在 Mo 平台（<https://mo.zju.edu.cn/>）上进行实际测试。

7.2. 实验要求

- a) 分组完成，每组不超过 3 位同学
- b) 根据小组技术特长，选择合适的技术路线，建立多人无限注德州扑克 AI 算法、绘制学习曲线、博弈收益曲线
- c) 参与 Mo 平台的多人无限注德州扑克竞赛，根据竞赛结果获得排名

7.3. 实验环境

使用 Python 的 MindSpore 库。

7.4. 参考资料

德州扑克：

<https://zh.wikipedia.org/wiki/德州扑克>

Numpy：

<https://www.numpy.org/>

<https://www.runoob.com/numpy/numpy-tutorial.html>

MindSpore：

<https://www.mindspore.cn/docs/zh-CN/r0.7/index.html>