联创ai组熬测

序

- 时间：10/13日 21:00 - 次日6:00，启明学院在晚上12点之后会关门，提前做完题目之后，离开是允许的(如果在12点之前)
- 熬测过程中，若有身体不适请不要硬撑，马上与在场的负责人员沟通
- 810 房间有无线网 UniqueStudio-810， 密码是 uniquestudio
- 完成题目时可以进行网络搜索，但是不能互相交流，请不要抄袭网上的代码

提交事项：

- 每一题需要提交的文件存放于对应的文件中，文件夹名与题目文件夹名一致
- 所有文件统一打包，用 组别+名字命名

要求：

- 在完成题目时，请用一个markdown文档记录你对每个题目的认识和理解，最好能记录做每道题的起始和结束时间，文件名为组别+名字+notes
- 在做题时，请尽量留下有效的注释，以便阅读
- `Python`代码请尽可能遵循标准代码规范，推荐Google风格指南[EN， ZH]
- 请对自己的代码负责

关于题目：

- 每道题有自己的分值，代表着最高完成度下可以获得的分数，而不是难度。如果只是应付性的做题，即使完成了也并不一定代表着有很高的分数。
- 请在能力范围内选择实现难度高，效果好的模型与算法会获得更高的分数
- 优雅的代码风格和高质量的注释以及详尽的记录会额外加分
- 一题多解也是加分项
- 题目的量和难度都较大，理论上不可能做完，请注意取舍

祝熬测顺利

基础题

1. 数据预处理 (100)(Feature Engineering)

此数据集来自某公司对离职情况的统计，我们给出了3个文件: X_test.csv， X_train.csv， y_train.csv 分别为测试集的特征，训练集的特征和训练集的标签

要求：

- 请对数据集进行数据预处理
- 必须使用 sklearn.svm.SVC 的模型进行拟合数据，并给出测试集的y预测值

提示：

- 请尽可能地使用 你能想到的数据处理方法进行数据处理
- 请尽可能地进行调参以获得最优的acc
- 数据处理尽可能自己完成，使用的库尽量不超过 numpy， pandas，使用sklearn的相关库会降低评分
- 请尽可能地提升**acc**
- 请提交 y_pred.csv

2. 最小值寻找(60)(Search Minimum)

空条承太郎在学习机器学习时，被一种名为粒子群优化的算法吸引住了，这是一种寻找参数空间内最优值的算法。现在他想用这种算法来帮助他的舅舅仗助完成暑假作业，作业内容如下：

要求：(30')

- 利用 `numpy` 实现粒子群优化，来寻找到函数 $f(\vec{x}) = \sum_{i=1}^{5}[x_i^2 - cos(2\pi x_i)]$的最小值，其中:(30')
  - $\vec{x}$ 为 5 维向量，即 $\vec{x} = (x_1, x_2, \dots, x_n)$。
  - $\forall x_i \in \vec{x}, x_i \in [-5.12, 5.12]$

注意：

- 请用有效的可视化手段将函数值优化的过程显示出来

题目还有以下额外加分项：

- 设计一个Partical类，用以储存相关信息 (10')
- 使用argparse库储存模型的超参数，使我们在命令行运行.py文件时能够修改对应的超参 (20')

提高题

1. 进化的神经网络(100)(Evolution Nets)

空条承太郎在尝试用梯度下降和牛顿法优化逻辑回归后，突然对进化策略产生了*兴趣*，请你帮助他深入对应的研究：

要求

- 请拟合数据集：evolution.csv
  - 数据对(x_1, x_2, y)代表点(x_1, x_2) 所属的类别 y
- 不可使用sklearn库

要求：

- 用numpy建构一个有一个隐藏层的全连接神经网络，其中激活函数分别为 tanh 和 sigmoid函数，隐藏层size=4，loss为 $J = -\frac{1}{m}\sum_{i=0}^{m}\left(y^{(i)}\log(\hat{y}^{(i)}) + (1-y^{(i)})\log(1-\hat{y}^{(i)})\right)$
- 使用进化策略对神经网络进行优化，尽可能最小化Loss，提高acc

注意：

- 请用有效的可视化手段将**loss**与**acc**优化的过程显示出来

2. 计算图(70+30+100)(Computational Graph)

请你使用`Python`，完成一个可扩展的计算图

要求：

- 可以对 初等函数 和 矩阵乘法 进行求导及梯度反向传播
- 不允许使用除`numpy`和`pandas`之外的任何第三方库。
- 理想效果 (仅供参考，不必完全一致):

```
>>> B = ([[0.3470,  0.3916,  0.7803],
        [0.5127,  0.1748,  0.2130],
        [0.9227,  0.0407,  0.2667]])

>>> C = ([[0.5500,  0.5612,  0.6200],
        [0.2347,  0.0989,  0.2991],
        [0.6708,  0.0784,  0.7413]])

>>> A = mean(matrix_multiply(B,  C) + B**2 - exp(C)/3 + 2*log(C))    #正向传播
>>> print(A)

Out [1]  ([[-0.8470,  -1.2917,  -0.0561],
         [-2.5920,  -4.6437,  -2.2899],
         [ 0.0967,  -4.9078,  -0.4450]])

>>> A.backward()    # 反向传播
>>> print(B.grad,  C.grad)

Out [2]  ([[0.2695,  0.1573,  0.3390],
         [0.3063,  0.1091,  0.2130],
         [0.3974,  0.0793,  0.2249]])

         ([[0.5379,  0.5291,  0.4876],
         [0.9676,  2.2743,  0.7604],
         [0.3989,  2.9341,  0.3620]])
```

注意：

- 式中`A,  B,  C`均为自定义类;`mean,  matrix_multiply,  exp,  log`均为自定义函数。
- 初等函数包括: 幂函数，指数函数，对数函数，三角函数(sin，tan，cos)及其经过任意次有理运算，复合运算得到的函数

拓展

1. 在此基础上实现一个可反向传播的单层全连接神经网络。不允许使用除`numpy`和`pandas`之外的第三方库。(30)
2. 在此基础上实现包含一个卷积层（参数自订）、一个平均池化层（参数自订）、一个全连接层和一个`Sigmoid`层的`CNN`，不允许使用除`numpy`、`pandas`和`pillow`之外的第三方库。(100)

3.迷宫(100分)(maze)

- 背景 强化学习也是机器学习里面很重要的一部分，在笔试卷子上看到很多人都写了AlphaGo，于是就有了这道题，不过你要做的相比较围棋简单多了，你只需要教会电脑走迷宫就可以了.

- 要求

上面是这个迷宫的图，这是一个4*4的迷宫，起始点在(0，0)，你要到达的地方是(2，2)，这里藏着宝藏，而(2，1)和(1，2)是两个陷阱，掉进陷阱或者找到宝藏游戏结束，你需要让电脑找到一个策略，使用最短的路找到宝藏。environment相关的已经实现，具体的内容和使用方法可以查看 environment.py (不允许修改这个environme部分代码) environment部分使用的example:

```
from environment import environment
env = environment()
start = env.start()
while True:
    # 拿到当前状态下的可能的行动
    available_action = env.get_action(state)
    # 使用你的策略选择行动
    action = your_algorithm(xxxxx)
    # 得到回报，新的状态，是否结束的标志
    next_state,  reward,  end = env.get_reward(state,  action)
    # 判断是否结束
    if end == True:
        break
    # 切换到下一个状态
    state = next_state
```

- 提示: 强化学习有很多的专业术语需要了解:

- state
- action
- environment
- reward
- agent

推荐的算法: Q-learning， Sarsa， DQN等等
不许直接按照自己找的路径hardcode，要求更换迷宫之后仍然可以work

4.数据降维(80)(di-reduction)

- 背景

数据可视化是如今数据挖掘领域中的一大重要课题. 数据可视化不仅是单纯地把数据变成图表，而是从图表所蕴含的客观事实出发，给予观察者洞悉对象本质的能力.

- 数据集：
  - `4-attr_3-cls.csv`
  - `13-attr_3-cls.csv`
  - `30-attr_2-cls.csv`
  - `64-attr_10-cls.csv`
    - 数据集解释：`csv`文件，第一行为属性名，之后每一行为一个样本，每个样本的最后一个值为其所属类别。
- 要求
  - 本题要求各位将拥有多个（多于三个）属性的对象展示在坐标系中。除多种属性外，每个对象还拥有自己的“类别”，我们希望你的可视化结果可以直观反映“相同类别”的“聚类”效果.
  - 四个数据集都要进行处理
  - 可以使用的库：
    - `numpy, pandas, matplotlib, seaborn`
    - `Python`常用标准库
    - 深度学习框架
      - `TensorFlow`
      - `PyTorch(新手推荐)`
- 关键词：
  - `python broadcasting`
  - `PCA`
  - `t-SNE`
  - `AutoEncoder`

5.Word2Vec(100+30+30+30)(word2vec)

- 编写代码通过word2vec模型实现单词的向量化，禁止使用相关软件或开源程序。
- 要求：
  - 利用该语料集训练词向量模型，词向量维度，任选：(100)
    - 50
    - 100
    - 200
    - 300
  - 实现接口：
    - 计算词汇的相似程度：输入两个词，输出这两个词的相似度（指标自选）
    - （可选）相似词的输出：用户输入词汇，输入与该词汇最相似的n个词汇（n为参数）(30)
    - （可选）词向量运算：根据用户输入的`pos`序列与`neg`序列，输出与`sigma(pos)-sigma(neg)`最相近的词汇，例如：(30)
      - `pos=["queen", "man"], neg=["woman"]`
        - `most_similar(pos, neg)`
        - 则是计算`queen + man - woman = ?`
        - `=> queen - woman = ? - man`
        - 所以期望输出`[king, ....]`

- 词向量可视化：(30)
  - 将高维数据映射到可视化空间，可以调库
  - 适当考虑美观程度