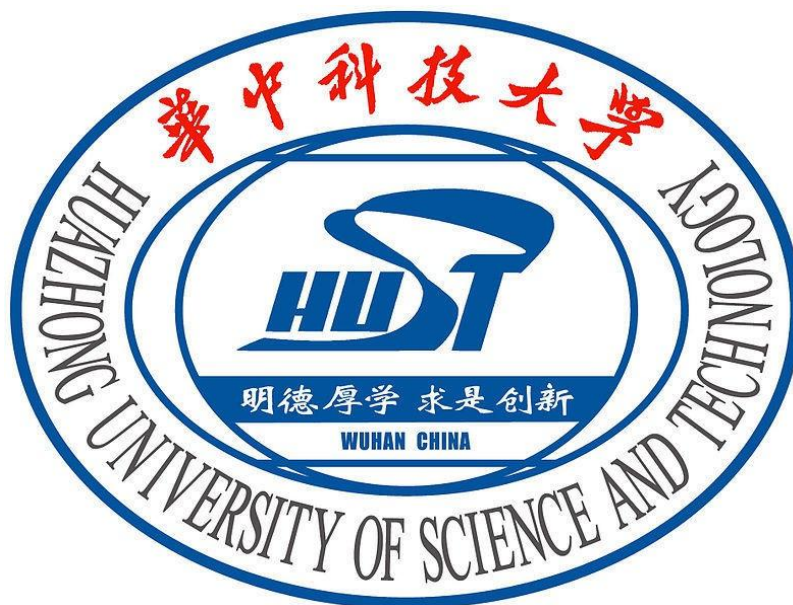


2022 年华南科技大学电气与工程学院

证券投资训练营

预习报告



院 系 : 电气与工程学院

班 级 : 电气 2005 班

姓 名 : 魏子健

学 号 : U202012350

任课教师: 郑玮

目录

一、 预习目标 (对应课程目标 1).....	3
1.1 对这门课的期望.....	3
1.2 对人工智能的理解.....	3
1.3 预习计划.....	3
二、 基础知识预习 (对应课程目标 1).....	3
2.1 工具安装和环境配置.....	3
2.2 基础知识预习.....	3
三、 预习项目设计(对应课程目标 2, 3).....	3
3.1 项目选题.....	3
3.2 程序设计.....	3
3.3 运行结果.....	7
四、 预习总结 (对应课程目标 4, 5, 7).....	8

一、 预习目标（对应课程目标 1）

1.1 对这门课的期望

- 加强自己的软件编程能力
- 学习人工智能、神经网络相关的代码，能够训练自己的神经网络
- 将人工智能运用到实际问题中，解决一些问题
- 了解更多的计算机前沿知识

1.2 对人工智能的理解

- 通过一定方法使机器能够像人脑一样思考或执行任务

1.3 预习计划

- 学习一些人工智能知识；
- 自己运行并修改一些神经网络程序。

二、 基础知识预习（对应课程目标 1）

2.1 工具安装和环境配置

- Python（pycharm/vscode）+Anconda
- GitHub

2.2 基础知识预习

- 通过《数学与建模仿真》课程学习掌握的 python 语言，能够实现简单的机器学习；
- 通过 Github、CSDN 学习一些简单的神经网络实例。

三、 预习项目设计（对应课程目标 2，3）

3.1 项目选题

2022 电工杯赛题《高比例风电电力系统储能运行及配置分析》

3.2 程序设计

用退火算法得到不同情况下 P1、P2、W 的最优解：

```
1. import numpy as np
2. import random
3. import pandas as pd
4. import csv
5. import time
6. data = pd.read_csv(r'C:\Users\82314\Desktop\负荷功率.csv', encoding='GBK')
7. ww = np.expand_dims(data[['风电功率']], axis=1)[: , 0]
8. ff = np.expand_dims(data[['负荷总功率']], axis=1)[: , 0]
```

```

9.     lenth1 = len(ww)
10.
11.     p1 = np.random.rand(96, 1)
12.     p2 = np.random.rand(96, 1)
13.     p3 = np.random.rand(96, 1)
14.     W = np.random.rand(96, 1)
15.
16.     c = 100 # 碳系数
17.     zong = 400 # 总负载
18.     left = [180, 90]
19.     right = [600, 300]
20.
21.     xmin = 180
22.     xmax = 600 # pg1
23.     ymin = 90
24.     ymax = 300
25.     zmin = 45
26.     zmax = 150
27.     for j in range(lenth1):
28.         zong = ff[j]
29.         w0 = ww[j]
30.         def initial_solution():
31.             particle_x = xmin + (xmax - xmin) * np.random.rand()
32.             particle_y = ymin + (ymax - ymin) * np.random.rand()
33.             solution = np.array([600., 300.])
34.             return solution
35.
36.
37.         # 对一个解进行微调
38.         def random_move(solution,best_solution):
39.             random_solution = solution.copy()
40.             random_solution += np.random.uniform(-10, 10, 2) * np.random.random() + \
41.                 np.random.rand() * (best_solution - solution)
42.
43.             return random_solution
44.
45.
46.         def calc_weight(solution):
47.             real_weight = 0
48.             for i in range(2):
49.                 real_weight += solution[i]
50.             return real_weight
51.
52.

```

```

53.     # 计算出这个解的好坏,
54.     def calc_energy(solution):
55.         f1 = 0.226 * solution[0] ** 2 + 30.42 * solution[0] + 786.80
56.         f2 = 0.588 * solution[1] ** 2 + 65.12 * solution[1] + 451.32
57.         flag = solution[0] + solution[1] + w0 - zong
58.         if flag >= 0:
59.             apple = 1.5 * (f1 + f2) / 4 * 700 / 1000 + c / 4 * (0.72 * solution[0] + 0.75 * solut
ion[1])\
60.                 + 0.3 * 250 * flag + 45 / 4 * w0
61.         elif flag < 0:
62.             apple = 1.5 * (f1 + f2) / 4 * 700 / 1000 + c / 4 * (0.72 * solution[0] + 0.75 * solut
ion[1])\
63.                 - 8 * 250 * flag + 45 / 4 * w0
64.         return apple
65.
66.
67.
68.
69.     # 计算接受概率
70.     def probability(delta, T):
71.         return np.exp(-delta / T) # 返回 e 的几次幂
72.
73.
74.     # 检查是否接收新的解
75.
76.     def deal(x1, x2, delta, T):
77.         # Delta < 0 直接接受,
78.         if delta < 0:
79.             return x2, True
80.         # Delta > 0 依概率接受
81.         p = probability(delta, T)
82.         if p > random.random():
83.             return x2, True
84.         return x1, False
85.
86.
87.     def print_status(trial, accept, best):
88.         print('Trial:', trial, 'Accept:', accept, 'Accept Rate:', '%.2f' %
89.               (accept / trial), 'Best:', best)
90.
91.
92.     # 初始温度
93.     Tmax = 1
94.     # 终止温度

```

```

95.     Tmin = 0.1
96.     # 温度下降率
97.     rate = 0.95
98.     # 每个温度迭代次数
99.     length = 300
100.    # 迭代 5 次就有不错的效果
101.    T = Tmax
102.
103.    # 初始化解
104.
105.    solution = initial_solution()
106.    a = calc_weight(solution)
107.
108.    best_energy = calc_energy(solution)
109.    # 保存当前最优解
110.    best_solution = solution
111.    best_weight = a
112.
113.    loop_count = 0
114.    trial, accept = 0, 0
115.    accepted = 0
116.    while T >= Tmin:
117.        for i in range(length):
118.            energy = calc_energy(solution)
119.            a = calc_weight(solution)
120.            # 更新当前最优解
121.            if best_energy > energy:
122.                best_energy = energy
123.                best_solution = solution
124.                best_weight = a
125.            # 对上一个解做个随机扰动
126.            random_solution = random_move(solution, best_solution)
127.            # 计算这个解的好坏, 和上一个解对比
128.            random_energy = calc_energy(random_solution)
129.            random_weight = calc_weight(random_solution)
130.            delta = random_energy - energy
131.            aa = calc_weight(random_solution)
132.            if aa + w0 > zong and (left <= random_solution).all() and (random_solution<=right).all
133.            (:):
134.                # 决定时候接受这个新的解或者保持上一个解
135.                solution, accepted = deal(solution, random_solution, delta, T)
136.            if accepted:
137.                accept += 1

```

```

138.         # 记个数, 看看试了多少次
139.         trial += 1
140.
141.         # 降低温度
142.         T *= rate
143.         loop_count += 1
144.         #if loop_count % 1 == 0:
145.             #print_status(trial, accept, best_energy)
146.
147.         p1[j] = solution[0]
148.         p2[j] = solution[1]
149.         w[j] = best_energy
150.
151.         print('-----END-----')
152.         print('最小成本: ', best_energy, '机组功率: ', best_solution, "Trial: ", trial)
153.         print(w0)
154. w = np.hstack((p1, p2, W))
155. title = ["p1", "p2", "W"]
156. Name = ['第四问 20 h=100']
157. data = w.reshape(96, 3)
158. i_str = str(Name)
159. path = i_str + '.csv'
160. try:
161.     with open(path, 'w', newline='') as t: # newline 是用来控制空的行数的
162.         writer = csv.writer(t) # 这一步是创建一个 csv 的写入器
163.         writer.writerow(title) # 写入标签
164.         writer.writerows(data) # 写入样本数据
165. except:
166.     pass
167. print('ok')

```

3.3 运行结果

成功得到 P1、P2、W 的最优解并写入 csv 文件:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	p1	p2	W																
2	241.2609	90	15016.87																
3	243.0273	90	15196.75																
4	272.226	90	16709.251																
5	268.5264	90	16406.119																
6	239.4033	90	15046.728																
7	248.9961	90	15367.053																
8	244.1061	90	15140.751																
9	246.8334	90	15193.363																
10	180	90	13091.781																
11	193.6017	90	13078.182																
12	180	90	13294.835																
13	180	90	13354.669																
14	180	90	13305.843																
15	196.9062	90	13038.139																
16	225.9696	90	14150.496																
17	202.6695	90	13226.511																
18	230.1333	90	14469.411																
19	252.2091	90	15644.028																
20	309.5538	90	18663.521																
21	314.2347	90	19104.624																
22	320.30313	91.172067	19742.88																
23	339.51003	98.556573	21394.501																
24	337.30643	97.71007	21507.207																
25	180	90	17722.661																
26	292.7892	90	19181.586																
27	332.2316	95.758296	21822.222																
28	342.59114	99.739058	22777.12																
29	354.3543	104.2611	23872.051																
30	253.9827	90	17830.77																
31	321.50169	91.634307	21434.448																
32	308.3313	90	20725.67																

四、 预习总结（对应课程目标 4， 5， 7）

- 掌握了简单的机器学习；
- 温习了使用 Python 处理、保存数据，提升了数据处理能力；
- 了解了人工智能的一部分内容，对神经网络有初步了解。