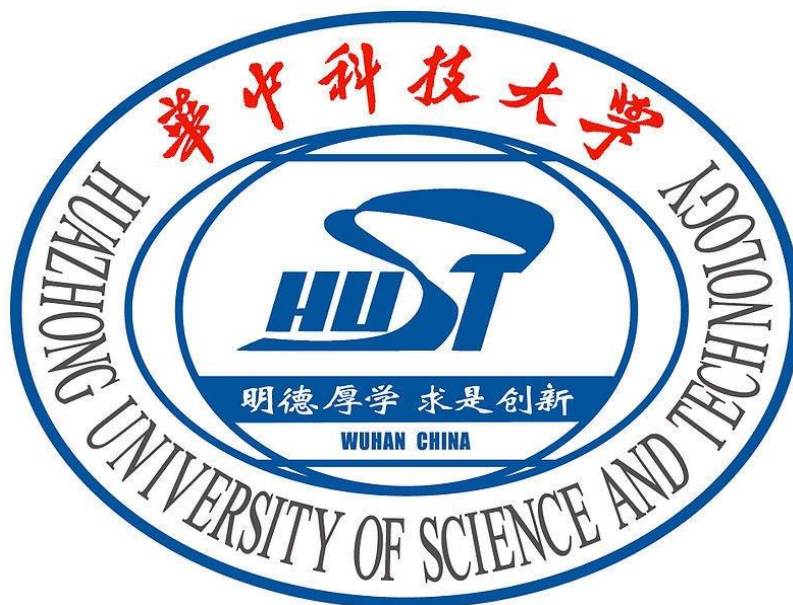


2022 年华南科技大学电气与工程学院

证券投资训练营

预习报告



院 系：电气与工程学院

班 级：电气 2009 班

姓 名：李瑞昊

学 号：U202012480

任 课 教 师：郑玮

目录

一、 预习目标	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	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	4
3.1 项目选题	4
3.2 程序设计	4
3.3 运行结果	5
四、 预习总结	6

一、预习目标

1.1 对这门课的期望

Python 编程语言具有的“清晰”、“简略”等特点，而且 Python 可以说是全能的，系统运维、图形处理、数学处理、文本处理、数据库编程、网络编程、web 编程、多媒体应用、pymo 引擎、黑客编程、爬虫编写、机器学习、人工智能等等，Python 的应用无处不在。因此掌握 python 的基础知识对自己未来的学习工作具有帮助性。

通过本门课程，希望自己能掌握深度学习，神经网络等方面的基础知识，并且能够运用这些知识去解决一些实际的问题，提高自己的编程能力。

1.2 对人工智能的理解

谈到人工智能，我们首先想到的是一些实际的例子，比如打败柯洁的阿尔法狗 (Alpha Dog)，波士顿公司的机器人，又或是科幻电影中的人机大战，人工智能 (Artificial Intelligence)，就是我们常说的 AI。由字面可知，就是通过人类的研究使得机器能像人一样甚至是超越人类的智慧去从事工作。人工智能可分为两类：弱 AI 和强 AI。弱 AI 也称为狭义的 AI，是经过训练的 AI，专注于执行特定任务。强 AI 是指机器拥有与人类等同的智能，它具有自我意识，能够解决问题、学习和规划未来。但是强 AI 仍完全处于理论阶段，还没有实际应用的例子。

目前，人工智能已经在许多现实方面得到了应用。语音识别方面，如 Apple 的 Siri 语音助手；客户服务方面，如大厅中常见的聊天机器；计算机视觉方面，如最近比较火热的汽车无人驾驶技术；自动股票交易方面等等。

关于对人工智能的期望，一方面自己希望未来人工智能继续发展，为我们的生活带来更便捷的体验，另一方面，众所周知科技发展是一把双刃剑，人工智能的发展也必然会带来一系列的社会问题，比如机器取代一些岗位，导致的人口失业问题，又或是机器能否发展出自己的思维所带来的伦理的思考等等。希望未来能实现人与机器更加和谐的发展。

1.3 预习计划

首先在电脑上安装 vscode 和 anaconda 作为自己的编程工具，并配置好相应的环境，然后通过 B 站等视频软件了解 vscode 和 anaconda 的使用，以及 python 编程的基本语法和操作，掌握机器学习和深度学习的基础知识。

二、基础知识预习

2.1 工具安装和环境配置

通过官网安装了 vscode 和 anaconda 软件，并且配置了相应的编程环境。在 anaconda 中安装了需要用到的程序包，能保证程序的正常运行。

2.2 基础知识预习

通过预习，自己学习了程序包的导入，基本的数值操作变量和赋值，流程控制，使用和定义函数，基本的数据结构，符号微分，数值微分，数值差分，基本的线性代数运算和非线性代数运算，数据的读取，画图的知识，拟合，线性规化，非线性规化，

优化算法等等，除此之外，自己也了解了深度学习，神经网络的相关知识。

三、 预习项目设计

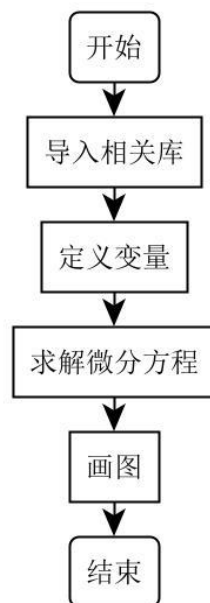
3.1 项目选题

选题：一个小球，在地面以初速度 20m/s 向上抛，每次落到地面回会弹起，速度变为原来的 80% ，画出他弹跳 5 次的图像，横轴为时间，纵轴为位移。

自己选择了解决小球弹跳问题作为自己的预习作业。自己的 `python` 基础比较薄弱，这个问题较为基础，自己可以独立完成，而且这个问题可以比较综合的检验自己对 `python` 基础知识的掌握情况。

对这个问题，自己期望能给出完整的可运行代码，能够计算小球的运动情况，最终画出小球弹跳的图像。

3.2 程序设计



3.3 运行结果

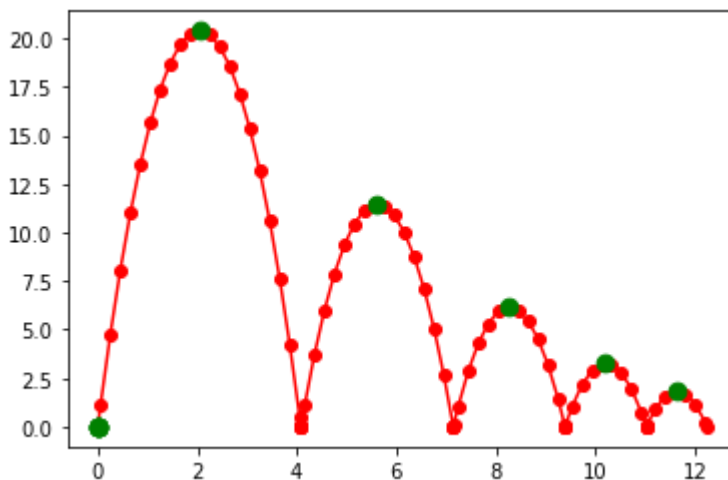
代码	
介绍: python	
	<pre>1. ### 2. #导入相关库 3. from scipy.integrate import solve_ivp 4. import numpy as np 5. import matplotlib.pyplot as plt 6. 7. ### 8. # 定义坐标 9. def dydt(t, y): #定义小球时间和高度的坐标 10. return np.array([y[1], -9.8]) 11. 12. # 定义小球到地面事件 13. def hitEvent(t, y): 14. return y[0] 15. 16. hitEvent.terminal = True 17. hitEvent.direction = -1 18. 19. # 定义小球到达最高点事件 20. def topEvent(t,y): 21. return y[1] 22. topEvent.direction = -1 23. 24. ### 25. #求解微分方程 26. 27. t0 = 0 #初始时间为0 28. tf = 30 #终止时间 29. v0 = 20 #初始上抛速度为 20m/s 30. y0 = np.array([0, v0]) #小球的高度初始值 31. tOut = np.array([t0]) #小球运动时间的计算输出值 32. y0Out = y0.reshape(2,1) #小球高度的计算输出值 33. 34. # 小球反弹的最高点位置 35. yTop=np.zeros([10,2]) 36. 37. # 反弹十次进行求解 38. for i in range(5): 39. sol = solve_ivp(dydt, [t0, tf], y0, events=[hitEvent,topEvent], max_step=0.2)</pre>

```

40. # 计算第一次弹跳过程小球的运动情况
41. lastEndTime=tOut[-1]
42. tOut = np.hstack((tOut, sol.t+lastEndTime))
43. yOut = np.hstack((yOut, sol.y))
44. yTop[i,0]=sol.y_events[1][0,0]
45. #计算碰撞 i 次小球反弹的最高点纵轴坐标
46. yTop[i,1]=sol.t_events[1][0]+lastEndTime
47. #计算碰撞 i 次小球反弹的最高点横轴坐标
48.
49. #计算小球下一个反弹过程
50. y0[0] = 0
51. y0[1] = -0.8 * sol.y[1][-1] #碰撞过后速度变为原来的 80%
52.
53. ###
54. #绘制图像
55. plt.scatter(yTop[:,1], yTop[:,0], c='g', marker='.',s=300,zorder=
5) #绘制小球最高点位置
56. plt.plot(tOut, yOut[0], c='r', marker='o') #绘制小球坐标

```

代码成功运行，可画出小球的运动情况图像，如下图。



四、预习总结

通过预习，自己学习了程序包的导入，基本的数值操作变量和赋值，流程控制，使用和定义函数，基本的数据结构，符号微分，数值微分，数值差分，基本的线性代数运算和非线性代数运算，数据的读取，画图的知识，拟合，线性规化，非线性规化，优化算法等等，除此之外，自己也了解了深度学习，神经网络的相关知识。