

分组学习任务安排

老师：任鹏举

助教：熊帆、涂志俊

1. 持续时间：

2019 年 04 月 24 日——2019 年 05 月 24 日

注：实际时间根据每队情况可以做相应的调整，但需要提前向我们说明

2. 分组情况：

AI开放实验分组名单			
分组	组员名单	分组	组员名单
1	罗中沛	5	朱雨航
	胡澄洋		潘月明
	李玥莹		钱龙
2	闫若璠	6	郭王懿
	何博宇		李嘉硕
	刘鑫		孙潇龙
3	朱佳怡	7	安静怡
	邓思益		汤凯雯
	曲劭儒		关之浩
4	支宇晗	8	黄隆宁
	蒋烨		韩子凡
	刘一凡		刘爽

3. 任务说明(选其一)

(1) 数字语音识别

- 题目说明：语音识别（speech recognition）技术是一项计算机自动将人类的语音内容转换为相应的文字的技术，本次任务只针对 0-9 这十个语音进行分类，即给一段连续音频，能够识别出是 0-9 哪个数字或哪串数字(数据集中的发音是 zero-nine)
- 完成结果：
 - 将原数据分割为训练集和测试集(具体比例自己定)，在训练集上训练你的分类器，在测试集上测试最终的分类结果，不断优化直至最优；
 - 编写一个程序脚本，可以录制人的语音，去噪并调用该分类器自动转化为文本输出，录制的过程可以手动暂停也可以让程序自动分割你的语音信号；
- 相关资料：
 - TensorFlow 音频识别教程：
https://www.tensorflow.org/tutorials/sequences/audio_recognition
 - 数据集下载链接：
 谷歌云下载：
https://storage.cloud.google.com/download.tensorflow.org/data/speech_commands_v0.02.tar.gz
 百度云下载：
 链接: <https://pan.baidu.com/s/15GYwdDTdx9vwfXvYmlw90g> 提取码: Haol

(2) 强化学习训练 Agent 玩 Atari 游戏

- 题目说明：强化学习是 Agent 以“试错”的方式，通过与环境进行交互获得的奖赏来指导行为，以最大化数字型的奖赏信号为目的来构建一个从状态到动作的映射的学习。本次任务是阅读 DeepMind 的一篇文章，并用 Deep-Q-Network 复现

该论文;

- b. 完成结果: 完整将论文中的 DQN 算法用 TensorFlow 或者 Pytorch 实现, 并训练 Agent 流利地玩 Atari 中的游戏, 可以不局限于太空入侵者这款游戏;
- c. 相关资料:
 - ① 附件中的 Playing Atari with Deep Reinforcement Learning.pdf 和 DeepMind-DQN-nature14236.pdf, 第一篇是 DeepMind 2013 年发的, 2015 年经过一些修改之后再次发表在 Nature 上, 本次复现以后者的结构为准, 前者也有参考价值;
 - ② 环境: [Gym](#) 或 [A.L.E: Atari 2600 Learning Environment](#)

(3) SLAM 相关

- a. 题目说明: 在机器人导航领域, 最重要的一项技术是 SLAM (同步定位建图), 定位问题, 即确定机器人在场景中的位置, 建图问题, 即划当前环境中的可行驶区域, 细化到室内机器人导航, SLAM 问题可以被具体描述为: 如何确定机器人在室内的位置, 划分室内环境中的可行驶区域。本次任务是阅读一篇论文, 理解其算法原理, 并尝试复现。
- b. 相关资料: 附件中的 Autonomous Navigation Project Report.pdf, 描述了一种基于视觉 (摄像头), 划分视野范围内的可行驶区域与不可形式区域, 并基于导航算法规划路径, 实现机器人的自动行驶。
- c. 附注: SLAM 技术, 尤其是视觉 SLAM 技术, 自动驾驶技术涉及多方面知识, 是一个多学科融合的技术方向, 本任务里介绍的内容仅为对问题的简单描述, 让各位同学对项目有个大体的了解, 如果对此项目感兴趣, 愿意选择这个项目的同学, 需要不断的同组内老师针对项目内容做进一步的交流, 由于本项目较为困难, 题目说明中的任务内容仅作参考, 不成为打分的绝对依据, 会根据各位同学的实际情况进行打分。

(4) 解释器初步

- a. 问题背景:

人工智能应用发展迅速, 但落地不易。在无人机等面向边缘计算、电池供电的自主智能体领域落地, 往往需要低功耗、低成本、高实时性、易部署的计算系统, 而 FPGA 等高性能嵌入式芯片依赖特定平台相关语言, 开发周期长不易部署。

开发面向异构平台的神经网络模型优化工具, 可以有效结合软硬件, 让 python 等高层语言搭建的模型得以自动优化、快速部署落地。
- b. 题目说明:

学习解释器的工作原理, 并实现一个简单的算数表达式解析器。最终能够处理例如: $7 + 3 * (10 / (12 / (3 + 1) - 1))$ 的表达式。
- c. 完成结果:

跟随教程(6 部分)学习传统解释器的工作原理和构造, 教程后的习题用于自查完成教程的基础上能不借助任何教程构建题中算术表达式解析器(使用 Python)
- d. 相关资料:
 - ① Let's Build A Simple Interpreter **Part1 - Part6**
 - ② <https://ruslanspivak.com/lsbaws-part1/>
 - ③ <https://www.oschina.net/translate/build-a-simple-interpreter-part-1> (中文)
 - ④ 参考书籍
编译期工程(Engineering A Compiler)
现代编译原理(Modern Compiler Implementation) 有 C、Java、ML 三种版本

4. 周报提交:

从 2019 年 04 月 24 日开始, 每周周日晚十点前每个队向 tzj19970116@163.com 和 596904404@qq.com 邮箱各发送一份周报(只需要一份.docx 周报, 格式 ‘第一周周报_1组.docx’, 不需要压缩), 周报的模板在附件中, 之后按照第一到第四周来写报告:

- (1) 第一周即本周日(04.22)需要说明选题、队长队员及分工情况;
- (2) 如果任务还在进行中, 阐述一下任务进展和目前遇到的困难;
- (3) 如果任务完成, 请将最终完整的实验代码上传到自己的 GitHub 主页上, 分工情况、详细的思路, 实验结果和代码的运行流程写在 README.md 中, 周报中只需要放上代码链接并且讲述一下自己在学习和团队合作过程遇到的困难, 怎么解决的就可以