

谈宁 (17317904059) || tanning62273206@gmail.com

求职意向: 全职软件开发工程师 (后端/前端/移动端) 开始时间 2024/6

个人优势

计算机专业硕士在读, 熟悉全栈开发, 熟悉计算机数据结构和系统设计, 编程基础扎实, 全栈开发项目经验, 熟悉 Java、Python、SQL、HTML/CSS/JS、React 等。

教育背景

得克萨斯大学奥斯汀分校 计算机硕士 在读

伊利诺伊大学芝加哥分校 数学与计算机本科 2019/1-2021/5

芝加哥洛约拉大学 金融本科 2017/8-2018/12

工作经历

深泉教育 兼职教师 2023/8-至今

- 教授数学, 包括 AP 微积分, SAT 数学。
- 教授计算机, 包括 Java, Python。
- 教授英语, 包括托福, 雅思。

实习经历

上海超级计算中心 23/05-23/08

- 帮华为的 AI 平台 Mindspore 做技术支持, 完成了 21 个算法模型的验证并形成文档。
- 21 个算法模型中抽取可以单机多卡的模型做测试。
- 数据集整理, 并部署到 AI hub。
- 应用 neo4j 帮助上海中医药大学做了中药图谱, 关联药材, 古籍等。
- 验证了市公共算力平台的基本功能, 并书写了一部分市公共算力平台使用手册。

项目经历

Twitch 视频推荐

- 通过 Spring Boot 搭建了一个全栈网页应用, 可以支持搜索视频 (直播流、短视频)。
- 利用 MYSQL (部署到 AWS RDS) 以及 Spring Data JDBC 访问和操作数据。
- 使用 Spring HTTP client 实现了 RESTful 的 API, 可以获取来自 Twitch 的视频资源。
- 通过 Spring Security 提供身份验证和授权, 以保护应用程序免受恶意攻击。
- 前端使用 ReactJS 和 Ant Design 构建相应的页面。

- 开发了基于内容相似性的推荐算法来支持用户可能喜欢的视频资源。
- 使用 AWS App Runner 来容器化服务并且做部署。

APP 购买

- 前端使用 React JS 搭建一款 Saas 购买平台，可以支持上传、发售、结算、搜索等功能。
- 使用 React Router v4 和 JWT 来实现用户注册管理、登录登出等功能。
- 后端使用 Golang 处理前端的请求并且部署到 Google Cloud (Google App Engine)。
- 使用 Elasticsearch (架构在 GCE 上) 来提供各种属性的搜索功能。
- 集成 Stripe API 以完成在线支付流程以及查阅历史信息。

Starlink: 基于 React JS 的 Starlink 轨迹可视化

- 使用 ReactJS 和 D3 设计和开发了一个可视化仪表盘，根据地理位置实时跟踪卫星。
- 构建了基于地理位置、高度和持续时间选择器，以精确筛选卫星搜索结果。
- 使用 D3 在世界地图上动画显示所选卫星的路径，提高用户友好性。
- 将仪表盘部署到亚马逊网络服务 (Amazon Web Service) 以进行演示。

Tinnews: 一款类似 Tinder 的新闻应用

- 基于 Google 组件架构 MVVM 模式设计了 Instagram 风格的新闻应用。
- 使用 JetPack 导航组件实现了底部导航栏和页面导航功能。
- 利用第三方 CardStackView (RecyclerView) 支持新闻的滑动喜欢/不喜欢操作。
- 使用 Room Database、LiveData 和 ViewModel 构建本地缓存和离线模式支持。
- 集成 Retrofit 和 LiveData 从 RESTFUL 端点 (newsapi.org) 获取最新新闻数据。

Google Colab

- 使用梯度下降法，逻辑回归，反向传播以及其他机器学习算法来训练数据并使之可视化。
- 使用强化学习解决了小车上山 (mountain car) 问题。

Fact-checking Outputs from ChatGPT

- 了解大型语言模型 (LLMs) 如 ChatGPT 产生的非事实性输出，并验证这些输出的准确性。
- 使用 Python 和 PyTorch，结合 Hugging Face 的 transformers 包及其依赖项，进行数据分析。
- 使用 FActScore (Min 等人, 2023) 提供的数据集，其中包含 ChatGPT 输出的提议，

并对其进行验证。这些提议是从人类注释的事实中提取的。

- 验证这些提议与从维基百科中检索到的段落的一致性。每个事实由人类标记为“S”（支持）、“NS”（不支持）或“IR”（无关）。任务简化为预测S vs. NS。
- 实施基于词汇重叠和文本蕴涵的事实核查方法。首先，使用词袋模型的重叠度量进行预测；然后，使用预训练的DeBERTa-v3模型进行文本蕴涵。
- 最后，对文本蕴涵模型的结果进行错误分析。检查模型的假阳性和假阴性，并提交分析报告。

机器学习与自动驾驶项目 - 基于 SuperTuxKart 的视觉驾驶系统

- 设计并实现了基于 SuperTuxKart 游戏的低级控制器，用于自动驾驶小车，通过优化车辆行为和动作，有效实现了在各种赛道上的自动驾驶。
- 使用 PyTorch 框架，构建并训练了一个卷积神经网络（CNN）规划器，通过图像输入预测了游戏中的车辆行驶的目标点，实现了基于图像的自动驾驶。
- 开发了数据收集和处理流程，利用低级控制器生成的数据集，训练了视觉驾驶系统的规划模型，以提高车辆行驶的精确性和效率。
- 参与了不同赛道上的评估和测试工作，成功完成了在多个赛道上的自动驾驶任务，展示了对机器学习和计算机视觉技术的实际应用。
- 通过该项目，熟练掌握了深度学习模型的设计、训练和优化技能，展现了对自动驾驶系统开发的深刻理解和能力。

SuperTuxKart 冰球比赛智能玩家设计与实现

- 实现了基于图像的智能玩家，该玩家利用卷积神经网络处理游戏中的图像信息，通过图像推断冰球位置和其他球员位置，并采用优化的手动控制器以尽可能多地得分并赢得比赛。满足了50ms每步的时间限制，使用了深度学习网络处理图像信息。
- 设计并开发了基于状态的智能玩家，通过对车辆和球的状态进行分析，使用单个深度神经网络实现智能控制，并在游戏中实现尽可能多的得分以赢得比赛。满足了10ms每步的时间限制，并采用了 `torch.jit.script`。
- 通过与TA编写的智能玩家进行多场比赛，测试和评估了设计的智能玩家，在比赛中取得了令人满意的得分并获得了比赛胜利。
- 团队协作：与小组的其他成员合作，合理分工，协同开发了两种智能玩家，并有效解决了在开发过程中出现的问题。
- 提升效率：通过对代码和算法进行优化，确保智能玩家在游戏中稳定运行，满足比赛时间限制，并在赛场上取得较高的得分和胜利。
- 项目总结报告：详细撰写了对两种智能玩家设计、开发和测试的总结报告，解释了方法、结果和经验教训，展现了出色的沟通和表达能力。

数据集异常分析与修正

- 在自然语言推理（NLI）或问答（QA）领域的基准数据集（如斯坦福NLI、MultiNLI、SQuAD或HotpotQA）上，分析预训练模型的性能及其局限性。
- 运用对比集、检查表集、对抗挑战集和模型消融等多种分析方法，评估模型的表现及鉴别数据集异常。
- 采用ELECTRA-small模型进行训练，并对其在基准数据集上的性能进行探索和改进。
- 提出方法改进模型性能，如集中学习困难子集、基于集成的去偏方法、对抗数据训练以及对比训练。
- 实施、微调并评估模型性能提升策略，通过数据增强和重新训练等手段减少数据集异常和改善模型性能。
- 准备详尽的技术报告，阐述项目的调查结果、改进措施和对增强模型性能的深入分析。