人工智能导论项目报告——
《东方非想天则》格斗 AI 之探索

钱禹昂 221900149 袁理 221900178 田永铭 221900180
陆迅 221900174 蒋李杰 221900173

2023.1.14

摘要: 格斗 AI 的研究在 AI 领域一直占有一席之地。紧密结合课堂所学强化学习等知识，我们小组努力研究并成功做出了强大的格斗 AI。我们依赖的格斗游戏是《东方非想天则》，它是由上海爱丽丝幻乐团与黄昏边境研发的一款格斗游戏，以出彩的格斗设计和优美的画风深受玩家喜爱。在我们做出该工作之前，全网尚未有该游戏训练出来的格斗 AI（网络上只有脚本实现）。我们的工作取得了很大的成功。

1 成果简介

我们小组认真学习了强化学习、DQN 算法、C-51 算法等人工智能领域方面的知识，共同讨论，通力合作，严谨科学地进行训练，成果分为三代 AI，这三代 AI 不断改进提升，最终代 AI 已经是非常优秀的格斗 AI。

我们的成果可以概括为:

- 在小组交流合作中，我们对强化学习有了更深的理解，成功应用其于制作格斗 AI 上，帮助我们解决我们在该游戏上打不过电脑的窘境。

- 我们独立思考并实现了三代《东方非想天则》格斗 AI，最终代 AI 已经能够轻松打败该游戏最高难度的电脑 AI，甚至可以预期有与高玩比拼的实力。
- 我们总结提炼了完成项目过程中的经验，将其整理成论文，将为以后研究该方面格斗 AI 的人提供便利。

2 基础实现

在这一部分我们介绍我们小组项目的最初的基础实现。

2.1 基础算法——DQN 算法

为实现我们想要的格斗 ai，我们小组以 DQN 算法为基础，先后实现了 DQN 多种变种，如 Double DQN，C-51。DQN，即深度 Q 网络（Deep Q-network），是指基于深度学习的 Q-Learing 算法。Q-Learing 算法维护一个 Q-table，使用表格存储每个状态 s 下采取动作 a 获得的奖励，即状态-价值函数 Q(s,a)，这种算法存在很大的局限性。在现实中很多情况下，强化学习任务所面临的状态空间是连续的，存在无穷多个状态，这种情况就不能再使用表格的方式存储价值函数。为了解决这个问题，我们可以用一个函数 Q(s,a;w) 来近似动作-价值 Q(s,a)，称为价值函数近似 Value Function Approximation，我们用神经网络来生成这个函数 Q(s,a;w)，称为 Q 网络（Deep Q-network），w 是神经网络训练的参数。

2.2 环境搭建

该游戏没有已经实现的虚拟环境，不太可能自己模拟，于是我们只能在真实环境中训练 AI。那么 AI 如该何获取环境信息？我们尝试了两种办法：

1. 利用 openCV 传给 AI 游戏画面的截图: 效率不高，数据量非常大，干扰信息多

2. 直接内存侵入获取信息: 通过一些工具获得一些关键信息的在内存中的地址，如血量，坐标，速度，正在进行的动作等, 优势在于获取信息准确，并且效率高，但代价是丢失了画面上的一些信息

2.3 输入与交互

游戏中角色的行动由 10 个按键控制，共有 1024 个动作空间，我们考虑建立动作空间到二进制数的映射——“1”表示相应的按键按下，“0”表示相应的按键松开。项目初期，AI 的输入部分通过一个 map 存储十个按键，AI 在获取环境信息后返回一个 0 至 1023 的整数，然后对这个整数按位取与，对应 map 中十个按键的按下与松开。据此可以简单地实现输入交互。

2.4 训练学习

在 agent 与 enviroment 交互的基础上，我们参照 DQN 算法初步实现了训练的框架代码，此时 reward 函数仅基于血量的变动，神经网络和各超参都未经仔细调整。我们通过一些代码使得 AI 能够自己循环选择开局格斗不断进行训练。至此，一个最初版的 AI 得以训练起来。

3 核心探索、优化过程

第二部分的实现显然不能够做出一个合格的格斗 AI。在这一部分，我们将呈现我们实现前后三代 AI 的曲折过程和探索成果。

3.1 第一代 AI——差强人意

基于初版的模型，我们进行了训练，但是训练结果并不尽如人意。如下图所示，呈现出的结果完全不收敛，并且方差还在增大。在观察 ai 操作的时候，我们也发现它只会通过躲避来减少自身的扣血，这显然不是我们期待的结果。经过

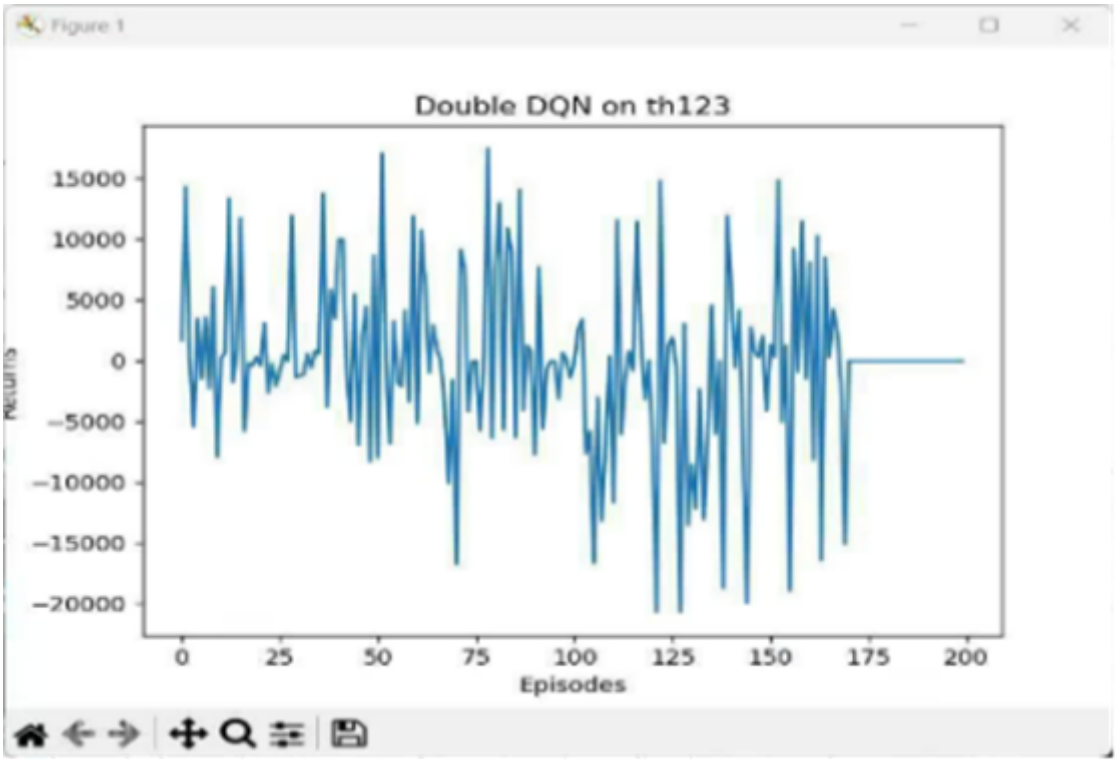

图 1: 第一代 AI 训练效果图

我们的分析，问题首先出在模型过于简单、奖励函数不合理、AI 无效动作太多。针对此，我们进行了如下改进：

1. 加入卷积层：单层网络效果完全不理想，模型过于简单，针对该问题，我们组选择通过加入卷积层等方式来增加模型的复杂度，以此来补足单层网络存在的缺陷

2. 增强 reward 函数：初始版本中，仅使用双方血量变化作为 reward 函数，导致严重缺少进攻意图 AI 仅仅缩在墙角或不断飞到天空，以躲避对方进攻，胜率极低。面对奖励函数不合理的问题，对 reward 函数进行大修改，从原来的只比较血量变化，修改为综合考虑各个情况（攻击命中，连击数，对手硬直等），给出相应奖罚，以此根绝消极防守的选项

3. 按键控制优化：原本的 1024 个动作空间中有太多的无效动作，我们删繁就简，仅仅考虑有效按键的状态，把动作空间从 1024 优化到 32，极大地优化了按键控制

经过这样的改进，我们再次训练，得到了第二代 AI。

3.2 第二代 AI——略有成果

在经历了各方面的改进后，第二代 AI 应运而生，相较于第一代，经过一轮改进的 AI 已经略有成果，后期能够达到 50+ 的胜率，但是第二代 AI 仍然存在一些缺陷：相较于第一代 AI 的消极避战，第二代训练后期出现了重复出招的问题，模型过度记忆了某些出招带来的奖励，忽视了更一般化的战术选择，因此缺少应变能力，更优的策略没有被探索到，训练不好的情况下胜率仅能过百分之 50。为了解决上述缺陷，我们采取了以下改进措施：

1. 修改超参数，增强探索性：为了使得 AI 能够更加主动地探索更多出招方式而不是限于一招，最直接地就是通过修改超惨增强探索性

2. 修改 reward 函数：针对第二代 AI 存在的出招固化问题，我们调整了 reward 函数，对多次重复出招给予惩罚，希望 AI 能够探索出更优的策略。

3. 优化环境空间, 筛选有效信息：删除了冗余的信息，仅仅保留对训练有用的信息

在这样的改进下，我们的 AI 训练效果有了不错的提升但是还是没能达到我们的高标准预期。

3.3 第三代 AI——锋芒毕露

为了打造更加强大的 AI，我们又努力钻研，采取了以下优化办法：

1. 加速游戏训练过程，增多训练代数：一是改进游戏结束条件，在达到一定标准（如血量少于 30%）时，即视为一局结束; 二是改变计时函数，从而加速游戏运行，可惜由于违反游戏使用规定等潜在风险而放弃。

2. 细节打磨:

   (a) 解决 AI 探索性过低、陷入局部最优问题：一方面提高 epsilon，使 AI 多尝试随机行动，提高探索性；另一方面降低学习率，使用自适应学习率的 adam 优化器，同时将初始学习率设为一个较低的值

   (b) 解决梯度爆炸问题：我们注意到 loss 大小有时突然很大，可能发生了梯度爆炸，采取以下措施：一是归一化 reward，原 reward 数值过大，对 reward 进行归一化，控制 reward 数值大小和比例；二是梯度剪裁，使用 adam 优化器的 L2 范数裁剪，进行梯度剪裁

   (c) 再试超参: 一个比较合适的超参相对容易找，而寻找效果极好的超参需要大量时间尝试，我们进行了大量尝试

3. 提高胜利指标: 我们不单单只追求胜利，而是希望有更好的结果，于是我们对 reward 函数进行了修改，强调剩余血量越多越好，战斗时长越短越好，命中率越高越好，连击段数越高越好，以此使得 AI 变得更快，更高，更强。

4. 其他尝试：
   突发奇想地，我们还想到可以教 AI 连招，这取得不错效果，不过有违背强化学习本意，最终舍去；我们还采取了针对性训练，即一次训练只一种游戏角色 AI，对战同一对手；此外。我们还查阅文献学习他人经验，不断学习、改进

最终 AI 效果:

在多番努力之后，我们制作的 AI 已经能够非常完美地战胜最高难度电脑 AI，胜利稳定在 90% 以上，能够轻松吊打我们的第二代 AI，并且在真人联机对打中取得非常好的效果。以下是我们的训练成果图：吸取了前两代 ai 的经验，第三代

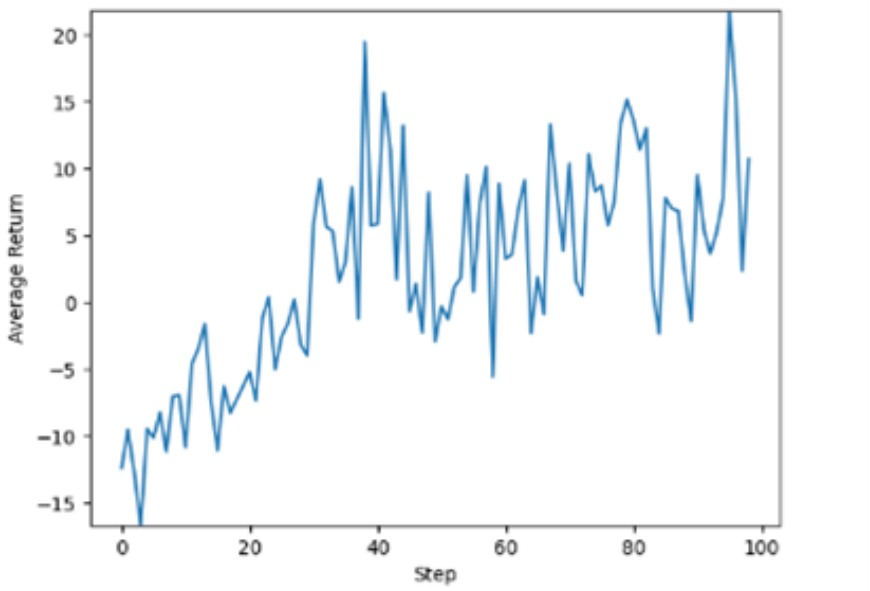

图 2: 第三代 AI 训练效果图

AI 成果显著。通过对图像，可以直观看出，我们的第三代 AI 的 reward 整体呈良好的上升趋势，并且最终在一个正值趋于稳定，进一步证明 AI 的训练效果良好，达到了我们的预期。

4 拓展延伸

在已有成果的基础上，我们还对项目进行进一步的拓展和延申：

1. 让 AI“鹬蚌相争”: 在面对最高难度电脑的情况下，已经训练得较为完善，很难产生进一步的突破；于是我们小组采用对抗式训练，将对手也设置为自己的 AI，加强训练

2. 优化信息获取 (图片 + 指令): 舍弃图象信息过于可惜，于是我们在已有的读取指令方式上，适当引入截图读取信息的途径，来作为辅助读取手段，加强训练

3. 更掘一隅：我们对我们项目的外延进行展望，预期在未来，我们的 AI 能够拥有更丰富的出招，预期能将我们的模型运用在其他格斗 AI 上，预期使得我们 AI 能过根据玩家所选难度进行动态调整

4. 整理成文：本项目完成过程中积累的丰富经验，已经过提炼整理成文

5 项目分工

此部分介绍我们的项目分工，全组成员通力合作，无一懈怠：

1. 平均 1 周开一次组会，激烈讨论，分享各自研究成果，探索下一周的研究方向

2. 独立各自设计、训练 AI，有好有坏，择优去劣

3. 共同制作演示文稿，整理成果并展示

4. 尝试各自产生的想法，如：随机出招训练、连招教学训练、进展平 a 远程放技能训练；除 DQN 外用 PPO 算法训练；各自独特的参数和 reward 设计

具体工作：

钱禹昂（组长, 功劳最大）：实现内存读取获取环境信息，AI 的按键控制部分及 DQN 算法训练框架代码的实现；尝试调整超参优化；尝试 dll 注入提高训练效率；实现对三代 AI 的各项优化；训练出三代 AI；参与论文写作

袁理：尝试使用 OpenCV 获取环境信息；训练各代 AI 提供训练数据；训练出一代 AI；参与优化第三代 AI；数据分析与思路提供；参与论文写作

田永铭：尝试使用 OpenCV 获取环境信息；二代 reward 函数的优化；训练各代 AI 提供训练数据；尝试调整超参优化；训练出二代 AI；撰写整合整篇论文

蒋李杰：前期资料查询与分析；承担训练各代 AI 工作，提供训练数据；参与优化第三代 AI；参与优化二代 reward 函数；提出应用梯度裁剪解决梯度爆炸问题；参与论文写作

陆迅：前期资料查询与分析；承担训练各代 AI 工作，提供训练数据；致力设计第二代 AI；参与优化第三代 AI；分析二代 AI 存在问题，提供优化思路；参与论文写作