|  |  |
| --- | --- |
| **成绩** | （采用四级记分制） |



****

**本科毕业论文（设计）**

**题目：**基于DQN强化学习的*Demon Attack*智能体

**学生姓名 孙潇桐**

**学 号 2021117405**

**指导教师 郝星星**

**院 系 软件学院**

**专 业 软件工程**

**年 级 2021**

**教务处制**

**二○二三年十二月**

诚信声明

本人郑重声明：本人所呈交的毕业论文（设计），是在导师的指导下独立进行研究所取得的成果。毕业论文（设计）中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人或集体已经发表或在网上发表的论文。

特此声明。

论文作者签名：

日 期： 2023年12月1日

**摘** **要**

本论文基于深度强化学习技术，设计并实现了一个基于深度 Q 网络（DQN）的智能体，旨在通过在雅达利游戏机上经典游戏 Demon Attack 中的训练，使智能体学会在给定环境中执行动作以最大化累积奖励。通过引入深度学习的元素，特别是卷积神经网络对状态进行建模，提高了对复杂环境的建模能力。通过实验结果显示，系统在训练过程中逐渐提高累积奖励，证明了深度 Q 网络在强化学习中的有效性。系统表现出良好的收敛性和稳定性，展现了在复杂环境中做出高效决策的潜力。

**关键词：**深度 Q 网络，强化学习，卷积神经网络

**Abstract**

This paper presents the design and implementation of an intelligent agent based on deep reinforcement learning using a Deep Q Network (DQN). The aim is to train the agent to maximize cumulative rewards by learning to perform actions in the classic Atari game *Demon Attack* on the Atari game console. The integration of deep learning, particularly convolutional neural networks (CNNs) for state modeling, enhances the system's capability to model complex environments. Experimental results demonstrate the gradual improvement of cumulative rewards during the training process, confirming the effectiveness of the deep Q network in reinforcement learning. The system exhibits good convergence and stability, showcasing the potential for making efficient decisions in complex environments.

**Keywords：**Deep Q Network, Reinforcement Learning, Convolutional Neural Network

**目 录**

[1 绪论 6](#_Toc153144292)

[1.1 课题研究的背景和意义 6](#_Toc153144293)

[1.2 论文内容和创新点 6](#_Toc153144294)

[2 国内外技术研究 7](#_Toc153144295)

[2.1 强化学习的发展 7](#_Toc153144296)

[2.1.1 时间差分学习 7](#_Toc153144297)

[2.2.2 深度Q神经网络 8](#_Toc153144298)

[2.1.3 自我博弈 8](#_Toc153144299)

[2.2 本章小结 8](#_Toc153144300)

[3 我的DQN算法设计 9](#_Toc153144301)

[3.1 架构设计 9](#_Toc153144302)

[3.2 训练过程 9](#_Toc153144303)

[3.3 奖励函数 10](#_Toc153144304)

[4 智能体的实现 10](#_Toc153144305)

[4.1 环境介绍 10](#_Toc153144306)

[4.2 关键算法实现 11](#_Toc153144307)

[4.2.1 智能体的模型 11](#_Toc153144308)

[4.2.2 训练过程 11](#_Toc153144309)

[4.3 训练结果 12](#_Toc153144310)

[总结与展望 13](#_Toc153144311)

[参考文献 14](#_Toc153144312)

1 绪论

1.1 课题研究的背景和意义

随着深度学习和强化学习在计算机视觉领域的不断发展，我致力于开发一个基于深度 Q 网络（DQN）的智能体，旨在通过训练学会在给定环境中执行动作以最大化累积奖励。这一工作对于解决复杂任务和在实际场景中执行智能决策具有重要意义。

这样的强化学习系统可应用于多个领域，其中一个典型应用是在视频游戏中训练智能体。通过强化学习，我的智能体可以学会在游戏中执行各种动作，最终达到最大化累积奖励的目标。这次我训练的智能体的目标是能够较好的游玩在雅达利游戏机上十分经典的雅达利游戏*Demon Attack* ，这个项目旨在帮助我理解强化学习的本质。因为强化学习不需要很多的实验数据，这是我选择做强化学习项目的重要原因，而且虽然这次强化学习项目的智能体只在游戏中和游戏进行交互，但是如果条件允许，智能体是可以和现实世界进行交互的，这就给了智能体无限的可能性。

1.2 论文内容和创新点

本论文致力于解决在雅达利游戏机上经典游戏 *Demon Attack* 中训练智能体的问题，以最大化其在给定环境中执行动作所获得的累积奖励。为了实现这一目标，我引入了深度 Q 网络（DQN）和强化学习技术，并在训练过程中进行了一系列创新性的设计。

首先，论文采用了深度 Q 网络结构，该网络包括卷积层和全连接层，以有效地处理高维输入，并学习状态与动作之间的映射关系。通过卷积神经网络对状态进行建模，系统具备更强大的对复杂环境的建模能力，相较于传统强化学习方法，展现了更好的适应性。

其次，为了鼓励智能体采取更积极主动的策略，我重新设计了新的奖励函数。通过在奖励函数中引入对智能体主动向敌人移动的奖励，以及对失去生命的惩罚，使得系统更倾向于寻求主动攻击敌方而非消极等待的策略。这种奖励函数设计不仅加速了系统的学习速度，还提高了智能体在游戏中的实际表现。

在训练策略方面，系统采用了灰度输入方式和适当的下取样，以提高运算效率。此外，我还设计了合理的训练策略，包括随机选择操作和使用模型得出结果的概率，以平衡探索和利用的需求。通过这些设计，系统在训练过程中表现出了较好的收敛性和稳定性。

最后，通过在 *Demon Attack* 环境中的实验，论文展示了系统的有效性和性能。实验结果显示，系统在训练过程中逐渐提高累积奖励，表明智能体能够有效地学习到在环境中执行动作的策略。论文对训练过程中奖励变化的分析进一步证实了系统的收敛情况和稳定性。

总体而言，该论文通过综合运用深度学习和强化学习技术，以及对网络结构、奖励函数和训练策略的创新设计，成功地实现了在 *Demon Attack* 游戏中训练智能体的目标，为解决复杂任务中的强化学习问题提供了有益的参考和启示。未来研究方向包括对系统性能的进一步优化、扩展到更复杂的任务和环境，以及与其他深度学习技术的整合，以提高模型的性能和适应性。

2 国内外技术研究

2.1 强化学习的发展

2.1.1 时间差分学习

强化学习的早期发展始于20世纪80年代初。Sutton等人在1988年的论文《Learning to Predict by the Methods of Temporal Differences》[1]中首次提出了时间差分学习的概念，为后来的强化学习奠定了基础。时序差分方法主要用于评估一个状态的价值，并通过不断地更新估计值来逼近真实的状态值。这种方法在强化学习算法中经常用于价值函数的估计，尤其是在与深度学习结合的深度强化学习中。时间差分技术主要分为三个部分：**观察环境**：智能体与环境进行交互，执行动作后观察到新的状态和获得的奖励。**更新估计值**： 使用当前的估计值（价值函数的估计）与下一时刻的估计值之间的差异，来更新当前时刻的估计值。这个差异即为时序差分。**逐步更新**： 在每个时间步都进行逐步更新，通过不断迭代，逼近真实的状态值。这个方法和我用的方法几乎一致，算是强化学习的一个比较基础但是有效的方法，奠定了强化学习的基本架构。

2.2.2 深度Q神经网络

2013年，DeepMind的团队通过发布的论文《Playing Atari with Deep Reinforcement Learning》[2]首次提出了深度 Q 网络（DQN）。该工作在Atari 2600游戏上取得了惊人的成绩，引发了深度强化学习的广泛关注。DQN的主要目标是通过深度神经网络来学习和近似一个状态-动作值函数（Q函数），以实现在复杂环境中进行强化学习任务的决策。我正好使用的就是这个方法，所以关于DQN的细节我会在后面提到。

2.1.3 自我博弈

2016年来自谷歌的机器人 AlphaGo击败围棋世界冠军柯洁，其背后就是强化学习的功劳。后来DeepMind实验室发布了一个不仅限于围棋的强化学习框架 AlphaZero[3]。其目标是发展一种通用的强化学习算法，无需人类专家知识，能够在多个领域表现出色。引入了CNN来识别情况并加入了蒙特卡洛算法进行搜索，然后通过自我博弈的方法得到大量的数据。

2.2 本章小结

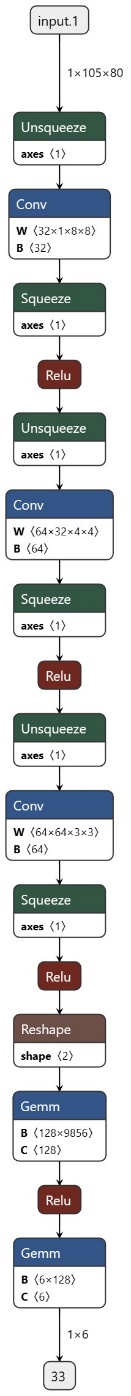
本章首先介绍了本论文的研究背景和意义，阐述了在深度学习和强化学习领域的发展趋势下，通过开发基于深度 Q 网络（DQN）的智能体来解决复杂任务和实际场景中智能决策的重要性。以雅达利游戏机上的经典游戏 Demon Attack 为例，展示了强化学习系统在视频游戏培训中的广泛应用，并强调了无需大量实验数据的优势。

随后，详细介绍了论文的内容和创新点。通过引入深度 Q 网络结构，重新设计奖励函数，优化训练策略等创新性设计，实现了在 Demon Attack 游戏中训练智能体的目标。这些创新点旨在提高系统的适应性、加速学习速度，并引导智能体采取更积极主动的策略。实验结果表明，这些设计有效提升了系统性能和训练效果。

在国内外技术研究方面，回顾了强化学习的发展历程，特别关注了时间差分学习的概念及其在强化学习中的重要性。介绍了深度 Q 网络的提出以及其在解决复杂环境中强化学习问题中的广泛应用。最后，提及了自我博弈方法的典型代表 AlphaZero，展示了其在多领域实现超越人类水平的成功案例。

3 我的DQN算法设计

3.1 架构设计

我的系统分吸取了AlphaZero[3]的经验，使用卷积神经网络来识别情况，再通过两个全连接层得到输出。模型可视化：

我使用的是和 Sutton[1] 一样的架构，每次操作都对模型进行更新，更新的过程将在3.2详细解释。

3.2 训练过程

每次对 Demon Attack 的环境做出动作，每次做出动作之后得到当前的状态（State），得到的奖励（reward），是否结束（term & trunc）和一些信息。

然后我们就能使用得到的信息更新网络：

1. 执行动作 并观察奖励 和图像 ：在环境中执行选择的动作，观察得到的奖励和下一时刻的图像。

2. 设置下一时刻状态 ，动作 ，和预处理后的状态序列 ：将当前状态、动作和下一时刻图像等信息保存为下一时刻的状态信息。

3. 将转换存储在 replay memory D 中：

4. 从 中随机抽样一个小批次（minibatch）的转换：

- 从 replay memory D 中随机抽样一小批次的经验。

5. 计算目标 Q 值 ：对于每个抽样的经验，计算目标 Q-value，其中对于终止状态，，对于非终止状态， 。

3.3 奖励函数

我认为在强化学习中，最重要的设计就是奖励函数的设计。在最开始的时候，我直接使用智能体在游戏中的得分来作为奖励函数。但是发现智能体得分似乎比较随机，而且也不怕死，看见炮弹都不躲开，于是我又加上了如果失去一条生命就扣除奖励函数的条件。但是，这样的话就会导致智能体对移动很消极，只会通过不断在原地发射炮弹来等着敌人装上炸弹。这算是一个比较好的得分策略，但是收敛实在是太慢，而且智能体不是真正的理解了游戏的本质，限制了智能体的上限。于是我就增加了一条：如果智能体向着敌人移动就给奖励，当前面有敌人的时候发射炮弹就有奖励。通过这样的设计，智能体的训练快了很多，而且得分更高了。

4 智能体的实现

4.1 环境介绍

我在我的笔记本上训练：

使用的关键软件的版本是：python@3.11, pytorch@2.1.0, gymnasium@0.29.1（提供 Demon Attack 的训练环境）。

硬件：CPU:AMD\_5900HS, GPU: NVIDIA\_RTX3050Ti\_laptop

4.2 关键算法实现

4.2.1 智能体的模型

使用了卷积神经网络优化，使用了三层卷积层来识别状态，加上两个全连接层来输出结果，下面是实现的具体代码：

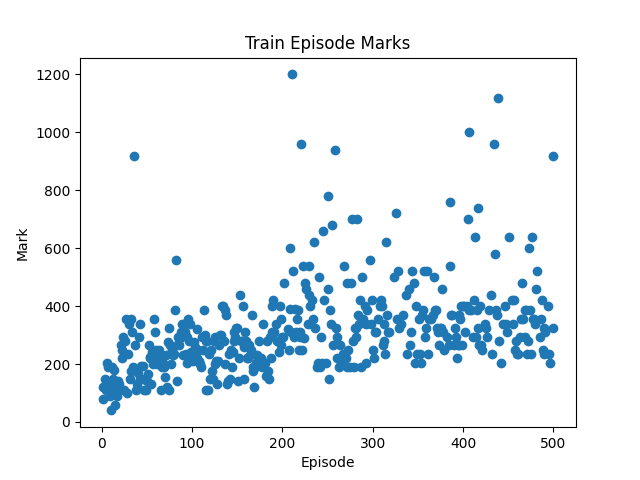
class QNetwork(nn.Module):  
 def \_\_init\_\_(self, output\_size):  
 super(QNetwork, self).\_\_init\_\_()  
 *# 1 \* 105 \* 80* self.conv1 = nn.Sequential(  
 nn.Conv2d(1, 32, kernel\_size=8, stride=4, padding=4),  
 nn.ReLU()  
 )  
 *# 32 \* 27 \* 21* self.conv2 = nn.Sequential(  
 nn.Conv2d(32, 64, kernel\_size=4, stride=2, padding=2),  
 nn.ReLU()  
 )  
 *# 64 \* 14 \* 11* self.conv3 = nn.Sequential(  
 nn.Conv2d(64, 64, kernel\_size=3, stride=1, padding=1),  
 nn.ReLU()  
 )  
 *# 64 \* 14 \* 11* self.fc1 = nn.Linear(64 \* 14 \* 11, 128)  
 self.relu = nn.ReLU()  
 self.fc2 = nn.Linear(128, output\_size)  
  
 def forward(self, x):  
 x = self.conv1(x)  
 x = self.conv2(x)  
 x = self.conv3(x)  
 x = x.view(1, -1)  
 x = self.fc1(x)  
 x = self.relu(x)  
 q\_values = self.fc2(x)  
 return q\_values

4.2.2 训练过程

将3.2中的描述实现出来，具体代码如下：

for episode in range(num\_episodes):  
 state = preprocess\_observation(env.reset()[0])  
 total\_reward = 0  
 beg = time.time()  
 lives = 4  
 mark = 0  
 while True:  
 action = agent.select\_action(state)  
 next\_state, reward, term, trunc, info = env.step(action)  
 mark += reward  
 reward \*= 10  
 *# print(info)* done = term or trunc  
 next\_state = preprocess\_observation(next\_state)  
 if lives > info['lives']:  
 reward -= 10  
 lives = info['lives']  
  
 reward += pos\_calc(state, action, cnt)  
 total\_reward += reward  
 agent.train(state, action, reward, next\_state, done)  
 cnt += 1  
 state = next\_state  
  
 if done:  
 break  
  
 *# Append the total\_reward to the list for later analysis* episode\_marks.append(mark)  
 agent.save(episode)  
 print(f'Episode {episode + 1}/{num\_episodes}, Total Reward: {total\_reward}, '  
 f'Mark: {mark}, Time: {int(time.time() - beg)}s')

4.3 训练结果



这是我在训练过程中得到的散点图，横坐标是迭代的次数，纵坐标是得到的分数。可以看到，模型的下限在不断的提升，不过每局游戏都不一样，所以，还是有一些随机的成分。由于时间有限，训练500个循环已经花费了我20个小时，如果再多训练500个循环我估计效果估计会更好。

总结与展望

本论文基于深度强化学习技术，设计并实现了一个基于深度 Q 网络（DQN）的智能体，通过在雅达利游戏机上的经典游戏Demon Attack中的训练，旨在使智能体学会在给定环境中执行动作以最大化累积奖励。通过引入深度学习的元素，特别是卷积神经网络对状态进行建模，论文提高了对复杂环境的建模能力。通过实验证明，系统在训练过程中逐渐提高累积奖励，证明了深度 Q 网络在强化学习中的有效性。系统表现出良好的收敛性和稳定性，展现了在复杂环境中做出高效决策的潜力。

在论文的创新点方面，首先，采用了深度 Q 网络结构，结合卷积神经网络，提高了对复杂环境的建模能力，相较于传统强化学习方法具有更好的适应性。其次，重新设计了奖励函数，通过引入对智能体主动向敌人移动的奖励以及对失去生命的惩罚，加速了系统的学习速度，提高了实际表现。在训练策略方面，采用了灰度输入方式和适当的下取样，设计了随机选择操作和使用模型得出结果的概率，以平衡探索和利用的需求。

在训练结果方面，散点图展示了模型在训练过程中逐渐提高的总体趋势。虽然存在一些随机性，但模型的下限不断提升，表明系统在学习过程中取得了显著的进展。

展望未来的研究方向包括对系统性能的进一步优化、扩展到更复杂的任务和环境，以及与其他深度学习技术的整合，以提高模型的性能和适应性。另外，可以考虑在更多不同游戏环境下验证模型的泛化能力，以及进一步探索奖励函数的设计和训练策略的优化。

总体而言，本论文通过综合应用深度学习和强化学习技术，成功地解决了在Demon Attack游戏中训练智能体的问题，为解决复杂任务中的强化学习问题提供了我的思考和改进。

参考文献

[1] SUTTON R S. Learning to predict by the methods of temporal differences[J/OL]. Machine Learning, 1988, 3(1): 9-44. DOI:10.1007/BF00115009.

[2] MNIH V, KAVUKCUOGLU K, SILVER D, 等. Playing Atari with Deep Reinforcement Learning[M/OL]. arXiv, 2013[2023-11-23]. http://arxiv.org/abs/1312.5602.

[3] SILVER D, HUBERT T, SCHRITTWIESER J, 等. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm[M/OL]. arXiv, 2017[2023-12-10]. http://arxiv.org/abs/1712.01815.