厦門大學

机器学习
课程论文

题　　目：　基于深度强化学习的游戏自动控制

小组名单：　万鲲鹏 孙玉莹 谢欣雨

万鲲鹏 23320201154024

学　　号：　孙玉莹 23320201154022

谢欣雨 23320201154033

专业班级：　电子信息

任课教师：　丁兴号

日　　期：　2021 年 1 月 1 号

人员分工

姓名	学号	主要工作
万鲲鹏	23320201154024	主要负责 DRL、RL 飞机大战、超级马里奥控制代码实现，查找阅读相关文献，负责大作业报告中第三章系统模型、第四章中飞机大战强化学习控制、第五章中超级玛丽报告撰写，同时用 Visio 绘制图 3、4、5、7，进行实验制作视频 demo。
孙玉莹	23320201154022	收集整理数据集，协助实现代码实现，查找阅读相关文献，主要负责论文实验结果分析、结论、致谢、论文排版工作，整理报告
谢欣雨	23320201154033	收集整理数据集，协助实现代码实现，查找阅读相关文献，主要负责论文摘要、引言、绪论、算法原理的撰写，整理报告

目 录

基于深度强化学习的游戏自动控制

摘 要

2012 年深度神经网络模型的成功构建，解决了传统强化学习的诸多不足，使得直接从原始数据中自适应地提取精确的高层次特征成为可能，学者们开始着手研究于深度学习在强化学习中的应用。而为了测试结合深度学习模型和强化学习模型在不同的游戏中的表现，本文通过开发一个卷积神经网络模型，应用强化学习设计了两个专门的智能体分别来完成《超级马里奥兄弟》和《飞机大战》这两款游戏。该模型通过基于游戏画面场景的状态分析完成了图像的识别分类，从游戏画面帧的原始像素数据中学习控制策略，并训练模型使之在输入未知和多维数据时选择正确的操作，其本质上是一个对游戏场景中特定状态的模式识别过程。这种方式可以进行学习，且算法可以学会自动对图像进行特征提取，对于训练中未出现的场景和状态也同样可以进行分类和预测。本文中通过将博弈环境抽象为状态向量，并使用 DQN 算法来处理复杂的游戏环境，实现了以上两款游戏的自动控制。

关键词：强化学习；强化学习；游戏；自动控制

引 言

深层卷积神经网络其上含有的卷积核可通过和前层平面的每个小邻域相卷积来探测具有代表性的图像特征。在应用强化学习实现游戏的自动智能体的过程中，我们的研究目标是利用图像中的不同对象训练卷积神经网络，构建一个强化学习的控制器智能体，它可以学习游戏环境并且通过将整个环境抽象描述为几个离散的键值属性来表示状态[1]，使用 Q-Learning 算法作为以最大化奖励为目标的决策策略，从而使之在输入未知和多维数据时选择正确的操作。

值的注意的是，计算机在训练过程中所获得的信息与人类玩家完全一致，其仅包括游戏的每一帧图像、小鸟的动作列表(跳跃、无操作)、计算机做出动作之后的奖励、游戏是否结束的标志等。使用强化学习的难点在于如何定义状态、行动、奖励和解决时滞问题，并且当智能体获得奖励时，我们必须确定究竟是哪一动作在获得奖励中发挥了作用以及其作用发挥的权重。

本报告的其余部分组织如下:在论文的第 2 部分我们主要简要概述了游戏的 AI 界面和 Q-Learning 算法;在第 3 部分中解释了在实验中我们是如何定义状态、行动、奖励在强化学习中使用并对实验结果进行了展示与分析。

1 绪论

1.1 研究动机与目的

应用强化学习实现游戏的自动智能体的思路一经提出便受到了广大学者们的关注和讨论，相较于人类玩家的游戏模式，计算机自主进行游戏的模式可以得到有价值的观察，这对于作为游戏设计师而言，可以更好地理解游戏参数和玩家体验之间的关系，提供更多更有效的改进算法的思路，通过调整的游戏参数去探索游戏空间从而调整游戏难度以获得理想的玩家体验，并且还可助力于之后的游戏开发。此外，它还能带动无人驾驶、机器人导航和物体追踪等同类领域的研究工作。

1.2 研究背景及意义

1.2.1 深度学习与强化学习背景

2006 年机器学习界的泰斗发表的«A fast learning algorithm for deep belief nets»[2],文章中主要阐述的观点有：

在网络中包含多个隐藏层，其通过一定的训练方法后可以得到更优越的特征表达式，这些特征能更清晰地描述原始数据的本质特征，有利于将特征可视化或者任务分类。

可以使用逐层初始化（layer-wise pre-training）的方法来解决传统神经网络训练中的难题。该方法是通过无监督特征学习[3,4,5,6]（Unsupervised Feature Learning）来实现的。

2012 年 Google Brain 项目引起了广泛关注，其使用大规模并行计算平台进行算法部署和模型训练，构建了一个“深度学习的网络”（DNN，Deep Neural Networks）的模型[7]。由此之后深度学习给诸多领域都带来了新的发展，其中包括语音识别、图像处理等众多领域。

随着深度学习在各个领域的重大发展，使得其为强化学习领域的发展带来可能。传统强化学习[2]主要依赖于组合人工特征和线性价值函数或策略表达来实现，但是，系统的性能很大程度依赖于所提取的特征提取的效果，同时，人工提取特征存在只对特定任务有效的特性，且难度较大，耗费时间较长等特性。深度学习的发展使得直接从原始数据中自适应地提取精确的高层次特征成为可能。将深度学习应用于强化学习开始进入人们的视野。本文通过研究有趣的游戏来测试结合深度学习模型和强化学习模型在不同的游戏中的表现，对强化学习的算法进行分析。同时我们相信这项技术将可以应用到更多的领域。

1.2.2 游戏

本文尝试将将深度学习应用到游戏中去。首先分析视频游戏中的步骤：

玩家点击开始按键，游戏场景开始切换，玩家对出现的画面进行信号的处理。

大脑将视觉信号转换为语义信息，其中包括物体的类别，不同物体所在的位置信息。其后玩家根据大脑反馈的信息选择接下来的具体操作。并通过按键将此决策传递到游戏内部。

游戏接收指令并处理完成后，场景切换到下一帧，玩家可能会因此得到相应的奖励或惩罚。如此循环，直到游戏结束。

由分析可得出此视频游戏的流程图，分为玩家无法干涉的环境部分和玩家控制的游戏部分。接下来使用 agent 模拟人类玩家，即让 agent 去代替人来玩家执行以上玩家执行的操作。Agent 通过与人类玩家的对比得出需进行以下操作：

接收游戏场景信号和回复信号，对收到的信号进行处理，如特征提取等。进而识别位置信息等。

根据历史信息及经验，对当前的游戏场景进性分析后作出相应的游戏操作。

将采取的相应游戏操作通过按键传递给游戏模拟器。

我们希望 agent 可以像人类一样通过不断的更新经验值来使游戏玩的越来越好，将 DL 强大的特征提取能力和 RL 的环境学习能力结合的深度强化学习模型（Deep Reinforcement Learning, DRL）[8].

1.3 研究现状

应用强化学习(RL)[9]玩游戏一直是当今 AI 的研究热点，它可以通过场景训练模拟玩家行为从而实现智能体。在此研究领域，Mnih 等人于 2013 年提出了结合深层神经网络的强化学习的 DQN（Deep Q-Network）模型[10]，并成功训练出相应的智能体使之可自主完成 2600 种 Atari 游戏，并且在多个游戏[11][12]上达到了超越人类的专家级水平，受其启发，我们在这两款游戏中，很大程度地模仿了前人的方法，以深层神经网络作为近似函数代表 Q-Learning 中的 Q 值（动作值），并通过使用以前经验的“经验回放”从中随机抽取部分经验来更新网络，以便将模型得到的经验和延迟更新联系起来，从而获得目标值以提高稳定性。

2 算法设计

2.1 Q-Learning

由于智能体只在任何一个时间步长观察当前帧，即其只能对任务进行局部观察，这意味着当前屏幕的信息不足以理解智能体的整个游戏环境。所以我们考虑一系列的行为和状态，并从中学习策略。我们假设环境中的所有序列都可在有限的时间步长内终止，从而将游戏建模为一个有限马尔可夫决策过程[13]。此时状态之间的具体转换概率是未知的，因此我们无法采用常规的强化学习进行迭代，而 Q-Learning 算法却可以在不使用状态转移概率的情况下收敛，符合我们的需要。这也是我们选择 Q-Learning 算法的缘由所在。

在 Q-Learning 算法中，我们将学习环境视为一个状态机，通过值迭代来寻找最优策略，从而使总回报（奖励）最大化，对于表中的每一对状态、动作，它维护一个由 Q 表示的期望总回报(即时+未来)的值。对于特定状态下的每一个动作，都会给予奖励，Q 值总是使用贝尔曼方程进行迭代更新：

$$Q(s_{t+1}, a_{t+1}) = r + \alpha \max_a Q(s,a) \qquad (2\text{-}1)$$

但在实际处理过程中，我们并不直接使用估计 Q 值来更新 Q 值，而是通过梯度下降的逐步逼近的做法，使其收敛到最优的 Q 值，转移规则如下：

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha\,(r + \gamma \max_a(Q(s_{t+1}, a_{t+1})) \qquad (2\text{-}2)$$

在 Q-Learning 算法中，其输入点由s_t，a_t，r_t，s_{t+1}四项组成，分别代表四个主要因素:当前状态、选择动作、奖励和未来状态，函数通过其输入点来构建一个整体回报最大且可据此预测未来动作的模型[14]。

式(2-2)表示对于每个当前状态，我们将 Q 值更新为当前值、当前奖励和最大可能未来值的组合。其中 α $\in [0,1]$为影响学习收敛的速度的学习因子，$\gamma \in [0,1]$为权衡长短期回报间折扣因子，它表示在优化过程中有多少未来状态被考虑在内，$Q(s_t,a_t)$为当前状态的 Q 值，$Q(s_{t+1},a_{t+1})$为未来状态的 Q 值，r 为奖励，$r+\gamma \max_a(Q(s_{t+1},a_{t+1})$表示根据当前的 Q 表，在状态$s_t$执行动作$a_t$转移到状态$s_{t+1}$后得到的真实的 Q 值。其原理图如下：

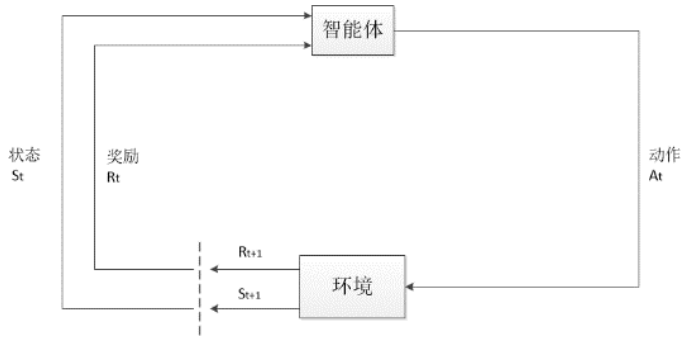

图 1 Q-learning 原理图

实际上，在训练智能体时，我们使用的是 ε 贪婪的 Q-Learning 算法，该算法是 Q-Learning 的一个小变体，它根据概率 ε 随机选择每一步的行动或根据概率为$1-\varepsilon$的 Q 表，选择最佳行动。执行该操作后，Q 表更新为 1。

2.2 Deep Q Networks

由于 Q-Learning 的核心是 R 表与 Q 表，其行表示状态，列表示动作，因此，若我们使用 Q-Learning 进行迭代来更新值的过程中，需要读取和维护整个状态表，才能计算出未来期望状态值，而无论是维护还是读取，其代价都是比较大的。

而我们知道，神经网络天生比较适合用来表示十分复杂的形式，所以用神经网络来维护 Q 表是比较合适的方式。DQN 就是 Q-Learning 神经网络的结果，其属于一种多层卷积神经网络，其输出为一个给定状态s_t和网络参数θ的动作值向量，是一个从 Rn 到 Rm 的函数，其 n 和 m 分别表示是状态空间与动作空间的维数[6]。因此，在 DQN 中，只需输入一个状态，神经网络即可直接生成 Q 值，其有着比 Q-Learning 快得多的更新速度，由此，我们通常用 DQN 来处理状态更加复杂的问题。

由于在 Q-Learning 中，连续存储的经验数据具有高度相关性。而由于样本之间的强相关性，若直接以相同的顺序在连续样本中完成 DQN 参数的更新显然是十分低效的。因此，我们想到：是否可通过随机化样本的方式来破坏样本的连续性从而降低其相关性，以达到减少更新的方差的目的。为此我们设置了一个经验回放存储器用于保存马里奥或我方战机过去所有的经历，采用“经验回放存储器+随机采样”的方式来破坏样本的连续性并用于进行 DQN 参数上的梯度下降，设置其 DQN 参数会在固定的时间间隔内进行更新，由于小批量选择结果具有随机性，所得的用于更新 DQN 参数的数据是去相关的，因此，此时得到的训练结果更加有效。经验回放解决了之前提到的奖励通常具有时滞性的问题。

而我们使用神经网络来预测动作，就必须了解训练神经网络的权重，但不同于输出层只有一个神经元的分类问题，此时的 DQN 输出层中有多个神经元来分别

对应多个动作，因此，就无法直接采用交叉熵函数作为损失函数，针对这种情况，我们采用两个结构一致参数不同的 DQN 网络：Q_net 与 Q_netT，每隔若干个时间步训练之后，就将 Q_net 的参数赋予 Q_netT，并使用一组参数 θ 来参数化逼近值函数，这种参数化逼近的方式使得在θ更新的同时，$Q(s_t,a_t|\theta)$同步完成整体更新，不仅避免了过拟合情况，还使得模型的泛化能力更强。DQN 模型的总体结构示意图如下：

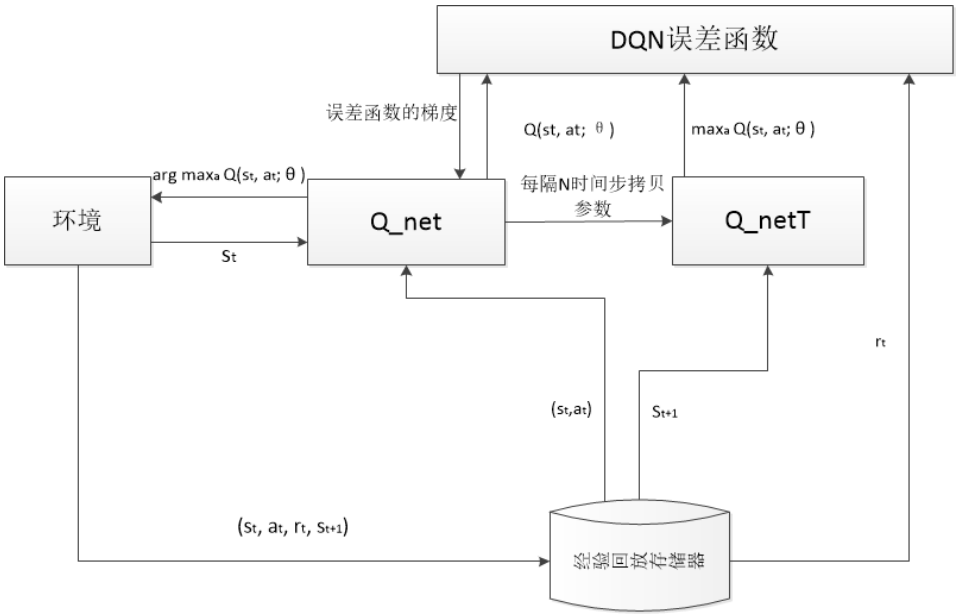

图 2 DQN 模型的总体结构示意图

在研究 Q-Learning 的损失函数之前，我们首先确立 Q-Learning 的优化目标——使得$T_D - Error$最小，见式（2-3）：

$$min(T_D - Error) = r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \tag{2-3}$$

若将 TD 目标$r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$作为标签，将$Q(s_t, a_t)$作为模型的输出 y，在加入参数 θ 后，原优化目标转化为式（2-4）：

$$\min(\arg_\theta): [r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}|\theta) - Q(s_t, a_t|\theta)]^2 \tag{2-4}$$

由此问题就转化为我们所熟悉的监督学习中的回归问题，显然我们所要确定的损失函数就是其均方误差，因此可用梯度下降算法最小化损失，从而更新参数 θ，得到的损失函数的梯度下降公式如式（2-5）所示：

$$\theta \leftarrow \theta + \alpha [r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}|\theta) - Q(s_t, a_t|\theta)] * \nabla_\theta Q(s_t, a_t|\theta) \tag{2-5}$$

但此 TD 目标是标签，所以$Q(s_{t+1}, a_{t+1}|\theta)$中的$\theta$是无法更新的，这种方法只有部分梯度，称为半梯度法。而后，DeepMind 在 Nature-2015 版本中将 TD 网络单独分开，通过每隔固定步数将值函数逼近的网络参数θ拷贝给其本身的参数θ'，由此保证 DQN 的训练更加稳定，式（2-6）即为含有目标网络参数θ'的梯度下降公式：

$$\theta \leftarrow \theta + \alpha [r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}|\theta') - Q(s_t, a_t|\theta)] * \nabla_\theta Q(s_t, a_t|\theta) \tag{2-6}$$

此外，我们一般将奖励固定在某个阈值$r \in [-1,1]$上，避免 Q 值过大的情况发生同时确保梯度是有条件的。

DQN 算法和 Q-Learning 的区别是用 CNN（深度卷积神经网络）代替 Q 矩阵。其 Q 矩阵的更新公式为：

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma max_{a'} Q\left(s',a'\right) - Q(s,a)\right] \quad (2\text{-}7)$$

CNN 根据 SGD（随机梯度下降算法）选取下一步的动作，以类似的方式给出损失函数：

$$L(\theta) = E[(targetQ - Q(s,a;\theta))^2] targetQ = r + \gamma max_{a'} Q\left(s',a';\theta\right) \quad (2\text{-}8)$$

3　系统模型

3.1 游戏系统模型

对于游戏而言，存在环境、游戏角色和玩家所构成的一个游戏系统，玩家可以通过键盘或手柄控制角色的运动，在环境中获得最大的一个收益。而本文则采用强化学习算法选择下一时隙的动作在环境中获得一个最大收益。为简化模型本文仅考虑游戏中的三个部分：角色、环境和智能体，拓扑结构如下图所示：

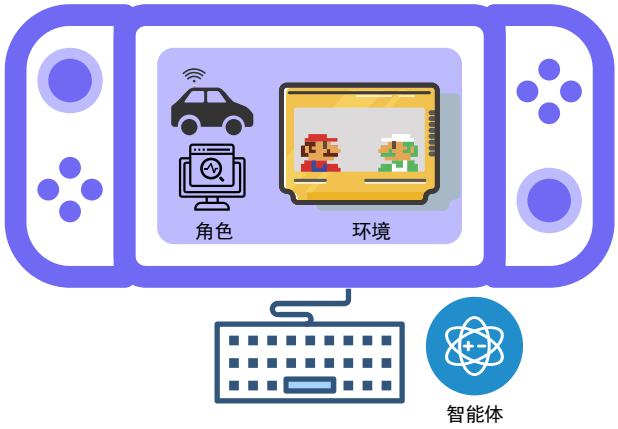

图 3 游戏系统拓扑图

智能体选择动作，游戏角色执行智能体选择的动作，环境进行反馈，角色在该动作下的动作$A^{(k)}$、。游戏图像$S^{(k)}$、角色所处位置$P^{(k)}$和游戏角色在该动作下的回报$R^{(k)}$，游戏系统的控制模型如下图所示。

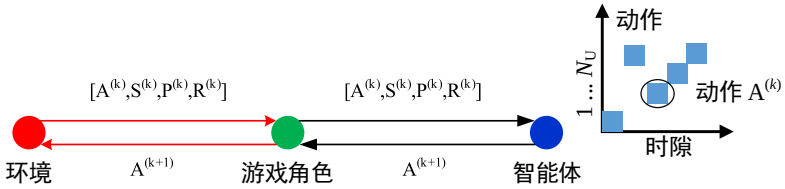

图 4 游戏系统控制模型

智能体通过强化学习算法，做出决策选择合适的动作$A^{(k)}$，环境做出反馈后获得游戏的收益。本游戏系统中所用的参数设置如下表所示：

表 1 系统参数

参数	含义
$A^{(k)}$	当前动作
$S^{(k)}$	图像数据
$P^{(k)}$	游戏角色所处位置
$R^{(k)}$	游戏角色在当前动作下的奖励

3.2 智能体的算法设置

本文采用强化学习算法，通过 Q 矩阵或 CNN(卷积神经网络)进行决策，根据当前状态选择下一时刻的动作。致力于让游戏中的角色选择最好的动作，获得最大的效益。

本文中主要采用 RLAGC 算法，通过 Q 矩阵，通过贪婪算法选择该状态下 Q 矩阵中值最大的动作，决策下一轮的游戏动作。RLAGC 算法伪代码如下所示：

表 2 RLAGC 伪代码

1:初始化算法参数$\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{Q}$
2:获取初始的动作$\boldsymbol{a^{(0)}}$
3: For $\boldsymbol{k=1,2,\cdots,n}$
4:获取上一时刻的状态信息$\boldsymbol{p^{(k-1)}}$
5 根据获得的状态信息，计算奖励函数$\boldsymbol{r^{(k)}}$
6:使用状态、选择的坐标（动作）和奖励函数，以学习率$\boldsymbol{\alpha}$、折扣因子$\boldsymbol{\gamma}$更新 Q 表
7:根据量化方案编码状态，并获取对应状态下所有动作的 Q 值
8:根据$\boldsymbol{\varepsilon}$-greedy 算法获取当前状态下收益最好的动作$\boldsymbol{a^{(k)}}$
9:智能体选择动作$\boldsymbol{a^{(k)}}$获得动作下的回报值$\boldsymbol{\gamma}$
10：通过 k 时隙的状态、动作和回报更新 Q 表
11: End for

本文中主要采用 DRLAGC 算法，通过卷积神经网络，通过随机梯度下降算法选择该状态下的较好功率，决策下一轮的发射功率。DRLATPC 算法伪代码如下所示：

表 3 DRLAGC 伪代码

1:初始化算法参数$\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{CNN}$
2:获取初始的动作$\boldsymbol{a^{(0)}}$
3:For $\boldsymbol{k=1,2,\cdots,n}$
4:获取上一时刻的状态信息$\boldsymbol{p^{(k-1)}}$、$\boldsymbol{S^{(k-1)}}$
5:根据获得的状态信息，计算奖励函数$\boldsymbol{r^{(k)}}$

6:使用状态、选择的坐标（动作）和奖励函数，以学习率$\boldsymbol{\alpha}$、折扣因子$\boldsymbol{\gamma}$更新 CNN
7:获取对应状态下所有动作的奖励值
8:根据$\boldsymbol{SGD}$算法获取当前时刻状态下收益较好的动作$\boldsymbol{a^{(k)}}$
9:智能体选择动作$\boldsymbol{a^{(k)}}$获得动作下的回报值$\boldsymbol{\gamma}$
10：通过 k 时隙的状态、动作和回报更新 CNN 网络
11:End for

4 飞机大战智能体强化学习控制

4.1 飞机大战实验模型

在《飞机大战》这一游戏的 AI 界面中，飞机只有左、右、下三个控制按键。其中敌机在每 30 步随机生成一个，并向前直线运动攻击对面的智能体。智能体通过三个按键选择不同的动作避开障碍，并以最大限度攻击对面的敌机，飞机每 5 步发射一次子弹。

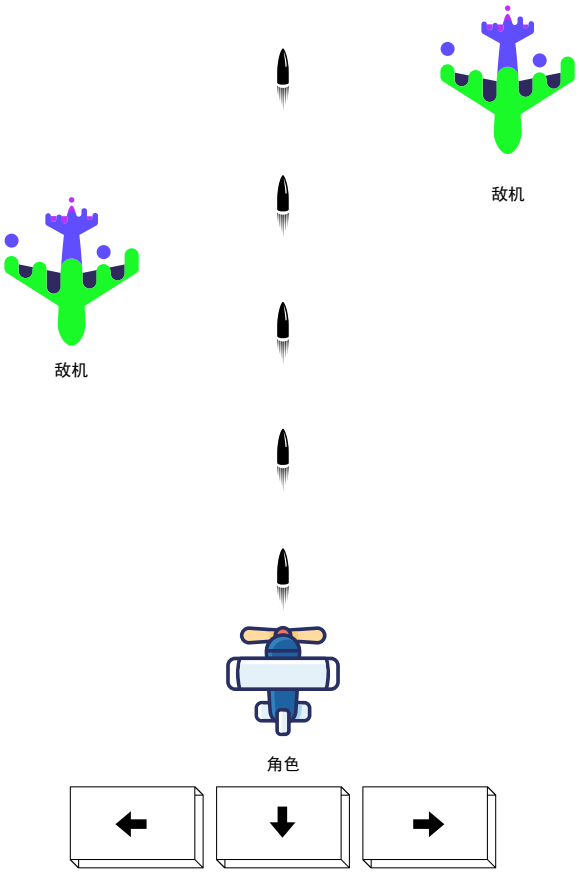

图 5 飞机大战实验模型图

奖励设置：当前步数没有被击中奖励 0.2 分、击中敌机奖励 2 分、被击落扣除 5 分。状态设置：状态设置为图像信息，图像宽度为 400 像素、高度为 800 像素，经过灰度处理后，通过 OpenCV 改变图像大小为高度 80、长度 80 的矩阵，

最终修改格式为 1 行 6400 列的矩阵以此为状态。

$$Reward = 0.2 * I(L > 1) + 2 * I(\mathrm{B} > 1) - 5 * I(\mathrm{L} < 1) \quad (4\text{-}1)$$

其中 reward 函数如公式 4-1 所示，其中 L 为角色是否存活：存活 reward 增加 0.2，若否则减少 5、若击中敌机则增加 2。

4.2 实验结果

本文对飞机大战，加入深度强化学习，并对模型进行训练，飞机有 3 个动作、状态数为 6041 个图像和位置信息所构成的 1 行 6041 列的矩阵。本文对该游戏进行了训练，实验结果请查看 demo。其中对模型进行了 5000 次训练，其每次训练的总回报率如下所示：

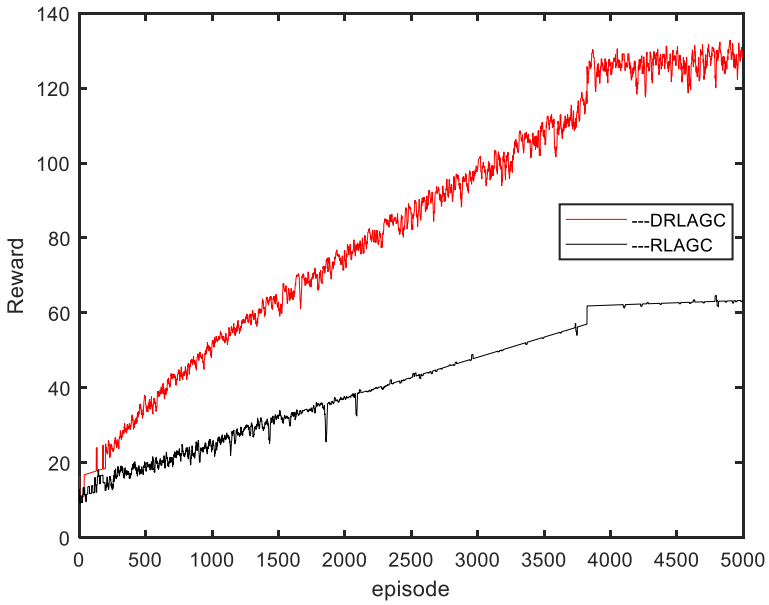

图 6 飞机大战实验结果图

4.3 结果分析

深度强化学习没有现成的样本，而是智能体在与环境的交互中收集相应的(状态、动作、奖赏)样本进行试错学习，从而不断地改善自身策略来获取最大的累积奖赏。由实验结果可以显示深度强化学习相比于深度学习，在相同训练次数的情况下，可以得到更多奖励，且随着训练次数的增加深度强化学习相比于深度学习得到奖励的优势愈加突出。得出强化学习在游戏中的可以更好的完成智能体的自动控制。

5 超级马里奥兄弟智能体强化学习控制

5.1 超级马里奥兄弟游戏模型

在《超级马里奥兄弟》这一游戏的 AI 界面中，马里奥有 6 个控制按键:上、下、左、右、跳跃和攻击。在每个时间步长中，智能体可从合法博弈动作集合中选择一个动作a_t将其传递给模拟器，模拟器相应修改其内部状态和游戏分数并返回当前场景的完整观察结果(包含该范围内敌人/道具/平台的位置和类型的数

组），而智能体通过游戏总分数的变化获得相应的奖励r_t。这款游戏的目标均是在每个阶段结束时，控制马里奥在不失去生命的基础上，通过收集道具、杀死敌人和快速完成关卡而获得尽可能高的分数。

其中马里奥有如图所示的 13 种不同的动作，并设置奖励为该状态下马里奥获得分数，选择向左+5，向右-5 每秒扣除 0.1。

$$Reward = 5 * \text{Right} - 5 * Left - 0.1 * \text{step} + \text{score} \quad (4\text{-}1)$$

图 7 超级马里奥兄弟实验模型图

5.2 实验结果

本文对超级马里奥兄弟，加入深度强化学习，并对模型进行训练，马里奥 13 个动作、状态数为 7056 个图像和位置信息所构成的 1 行 7057 列的矩阵。本文对该游戏分别训练了 100 次、8000 次、10000 次，实验结果请查看 demo。其中对模型进行了 5000 次训练，其每次训练的总回报率如下所示：

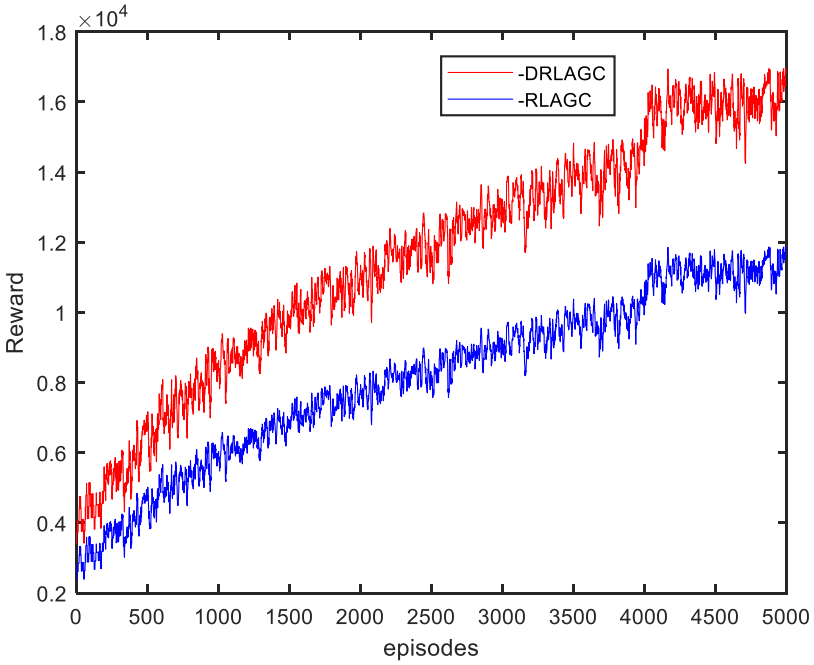

图 8 超级马里奥兄弟实验结果图

5.3 结果分析

由超级马里奥的例子说明了如何通过深度强化学习在该游戏环境下训练智能体。当智能体与环境交互时，通过大量的反复实验和迭代来更改策略。由该实验结果图显示，在相同训练次数的情况下，深度强化学习相比于深度学习可以获得更高的奖励分数，且随着训练次数的增加，深度强化学习相比于深度学习获得奖励的优势愈加明显，这同样说明深度强化学习可以在游戏中出色地完成智能体的自动控制。

6 结论

深度强化学习已经成功地应用于许多问题，由于在围棋和其他电子游戏上的出色表现而受到广泛关注。我们主要关注在游戏产业中如何利用深度强化学习技术，提升效率，改变生产方式。深度强化学习技术，是通过让智能体在游戏世界内探索的方式来训练模型提升水平，在合适的设计的基础上，往往能得到比较高水平的模型。本文通过强化学习设计了两个智能体来完成《超级马里奥兄弟》和《飞机大战》这两款游戏。通过将博弈环境抽象为状态向量，并使用 DQN 算法来处理复杂的游戏环境，实现了以上两款游戏的自动控制。在本文中我们只是列举了深度强化学习在游戏中的一些比较典型的例子，其中有的已经为产业带来价值，且其应用前景广阔。

致 谢

时光如白驹过隙，转瞬之间我的研究生生涯的第一学期已经接近尾声。从最初入学时的新奇与激动，到渐渐习惯，不过短短数星期而已。接着便是忙碌而充实的研究生生活。

学期初选课之际，很有幸能选到丁老师的机器学习这门课。丁老师深厚的专业功底和知识体系使我上课时总能听到老师对一些问题的独到的见解，激发了我对机器学习这一领域得浓厚兴趣，从而在该课题下进行浅薄的研究。在此，我谨向丁老师表示我最诚挚的感谢。祝愿丁老师身体健康，桃李满天下！

感谢实验室同学对我的帮助和鼓励，是你们使我能很快地适应研究生的生活。

最后，感谢我的家人对我的关心和包容，成为我前进路上的动力。感谢岁月，于世间欢愉与悲苦，许以宽宏。

研究生生涯已拉开序幕，我将带着老师、家人、朋友的希冀，努力书写自己的人生篇章！

参考文献

[1]Tsay J J, Chen C C, Hsu J J. Evolving intelligent mario controller by reinforcement learning[C]//2011 International Conference on Technologies and Applications of Artificial Intelligence. IEEE, 2011: 266-272.

[2]Geoffrey E.Hinton and Simon Osindero. A fast learning algorithm for deep belief nets[J].In Neural Computation,2006.

[3]Coates,A.Demystifying Unsupervised Feature Learning[D].PhD thesis,Stanford University,Palo Alto,USA,2012.

[4] Bengio,Y.Learning Deep Architectures for AI[J].Found.trends Mach.Learn.2(2009),1-127.

[5]Yu,D.Seltzer,M.l.,Li,J.,Huang,J.-T.,and Seide,F.Feature learning in deep neural networks-a study on speech recogniyion tasks[J].CoRR abs/1301.3605(2013).

[6]Ranzato,M.,lan Boureau,Y.,and Cun,Y.L.Sparse feature learning for deep belief networks[J].In Advance in Neural Information Processing Systems 20,J.Platt,D.Koller,Y.Singer,and S.Roweis,Eds.Curran Associates,Inc.,2008,pp.1185-1192.

[7]Arnold,L.,Rebecchi,S.,Chevallier,S.,and Paugam-Moisy,H.An Introduction to Deep Learning[M]. In Advances in Computational Intelligence and Machine Learning,ESANN’2011(2011),pp.477-488.

[8]Volodymyr Mnih,Koray Kavukcuoglu,and David Silver.Playing Atari with Deep Reinforcement Learning[J].In ICML,2014.

[9]Richard Sutton and Andrew Barto. Rcinforcement Learning:An Introduction[M]. MIT Press,1998.

[10]Sutton R S, Barto A G. Introduction to reinforcement learning[M]. Cambridge: MIT press, 1998.

[11]Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. arXiv preprint arXiv:1312.5602, 2013.

[12]Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. nature, 2015, 518(7540): 529-533.

[13]Klein S. CS229 Final Report Deep Q-Learning to Play Mario[J].

[14]Chen K. Deep reinforcement learning for flappy bird[J]. 2015.