单位代码 **10006**

学 号 **16231246**

分类号

密 级

****

毕业设计(论文)

基于深度强化学习的音乐节奏游戏智能体的研究与实现

|  |  |
| --- | --- |
| 院（系）名称 | 计算机学院 |
| 专业名称 | 计算机科学与技术 |
| 学生姓名 | KurohaneNioko |
| 指导教师 | 童超 |

2020年6月

北京航空航天大学

基于深度强化学习的音乐节奏游戏智能体的研究与实现 秦子柠 北京航空航天大学

**本科生毕业设计（论文）任务书**

Ⅰ、毕业设计（论文）题目：

基于深度强化学习的音乐节奏游戏智能体的研究与实现

Ⅱ、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

本毕业设计使用原始资料是Schulman J, Wolski F和Dhariwal P等人发表的《Proximal policy optimization algorithms》以及Sutton等人著《Reinforcement Learning: An introduction Second Edition》以及俞凯等人翻译的中文版，从中参照了（深度）强化学习算法的基本原理、常规实现方法以及重要指标参数。技术要求则以能解决领域内的基准测试项目为标准，在实验中按照算法实验的一般原则进行，以确保实验结果的科学性。

Ⅲ、毕业设计（论文）工作内容：

本毕设的主要工作内容如下：

(1)本文在回顾深度强化学习的发展历程与关键算法及其相关应用的基础上，从当前深度强化学习中关于音频的训练较少的现状出发，参考了其他研究者在这个方面的成果，推断出强化学习的仿真环境对音视频协同数据的训练的支持不足的结论。

(2)在(1)的基础上，本文找到了一类对音视频协同数据的智能体训练的支持相对较优电子游戏——音乐节奏游戏，并基于当代的一款音乐节奏游戏构建了一个相对简单开源的强化学习环境NaiveBandori。

(3)在(2)中构建的环境下设计支持音视频数据输入的神经网络BandoriNet，基于当前的一种主流深度强化学习算法进行智能体的训练。以音频数据是否在训练中输入智能体进行了一系列的对照实验，并分析实验结果。

Ⅳ、主要参考资料：

[1] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. MIT press, 2018:1-25.

[2] Sutton R S, McAllester D A, Singh S P, et al. Policy gradient methods for reinforcement learning with function approximation[C]//Advances in neural information processing systems. 2000: 1057-1063.

[3] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.

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

[5] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]. 2014.

计算机 学院（系） 计算机科学与技术 专业类 162320 班

学生 ： KurohaneNioko

毕业设计（论文）时间： 2019 年 12 月 23 日至 2020 年 6 月 4 日

答辩时间： 2020 年 6 月 4 日

成 绩：

指导教师： 童超

兼职教师或答疑教师（并指出所负责部分）：

系（教研室） 主任（签字）：

注：任务书附在已完成的毕业设计（论文）的首页。

**本人声明**

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者：KurohaneNioko

签字：

时间：2020年 6 月

基于深度强化学习的音乐节奏游戏智能体的研究与实现

学 生：KurohaneNioko

指导老师： 童超

摘要

随着近年来深度强化学习在算法上的不断革新及其在实际决策问题上的应用具有的突出表现，强化学习作为机器学习中的重要组成部分已经越来越受到工业界和学术界的青睐。不同于监督学习和非监督学习，强化学习算法的数据输入往往需要算法运行中与环境进行交互而即时产生——这也导致强化学习的实验往往受制于交互环境本身的框架内。当前研究领域内较为通用的环境包括OpenAI-gym、MuJoCo等，也有借助游戏中抽象出的接口而构建的环境如围棋、DOTA2等，亦有针对工业场景下的定制的环境，如虚拟淘宝等。

针对这些环境中的强化学习训练仅有视频数据而缺乏音频数据这一与生产生活中有偏差的共性问题，在对当今主流的深度强化学习算法的研究与实现的基础上，本文的研究内容有以下几个方面：

(1) 根据音乐节奏游戏《Bang Dream! Girls Band Party》，构建了一个能支持强化学习训练的、能同时进行音视频数据交互的强化学习训练环境，同时构建了便于实验的环境的简化版本。

(2) 在(1)中实现的训练环境内，实现了当前主流强化学习算法——近端策略优化，针对某一特定的拓扑结构的神经网络构建的强化学习算法中的智能体，设计实验对比音视频多模态输入训练和单一视频输入的训练在收敛性能以及模型最佳表现等方面的异同。

(3) 多次不同随机数种子的实验结果表明，在节奏游戏环境中，两种训练方式训练的模型都有概率能收敛到能与人类玩家水平相近的模型，而不带音频输入的模型在收敛到较为理想的结果的概率上、模型大小上、训练速度上都有较为明显的优势；简单的模态混合策略并不能提升当前主流算法的表现，也无法有效地解决深度强化学习目前普遍存在的陷入局部最优的问题。

关键词：机器学习，深度强化学习，策略梯度算法，近端策略优化

Research and implementation of rhythm game agents based on Deep Reinforcement Learning

Author: KurohaneNioko

Tutor: Tong chao

**Abstract**

With the continuous innovation of deep reinforcement learning in algorithm and its outstanding performance in practical decision-making problems in recent years, reinforcement learning, as an important method of machine learning, has been more and more favored by industry and academia. Different from supervised learning and unsupervised learning, the data input of reinforcement learning algorithm often needs to interact with the environment in the operation of the algorithm, which leads to the experiment of reinforcement learning is often limited by environments. At present, general environments include openai gym, mujoco, etc. there are also environments built from video games, such as go, DOTA2 and customized environments for industrial scenarios as well, such as virtual Taobao.

In view of the common problem that the reinforcement learning training in these environments only has video data but lacks audio data, which has deviation from the reality, based on the research and implementation of the current mainstream deep reinforcement learning algorithms, the research contents of this paper are as follows:

(1) Based on the music rhythm game "Bang Dream! Girls Band Party", a reinforcement learning and training environment that can support reinforcement learning and training, and can interact audio and video data at the same time is constructed. A simplified vesion of the environment is provided as well for the convenience of experiments.

(2) In the training environment implemented in (1), a current mainstream reinforcement learning algorithm, Proximal policy optimization algorithms, is implemented. Aiming at the agents of reinforcement learning algorithm constructed by neural network with a specific topology structure, experiments are designed to compare the similarities and differences in convergence performance and model optimal performance between audio-video multimodal input training and sole video input training.

(3) The experimental results of several different random number seeds show that in the rhythm game environment, all of the models trained by the two training methods have the probability of catch up with human players, while the models without audio input have obvious advantages in the lager probability of converging to the ideal results, smaller model size and faster training speed. The simple modal fusion strategy can't neither improve the performance of current mainstream algorithms, nor solve the problem that deep reinforcement learning falls into local optimum effectively.

**Key words**: machine learning，deep reinforcement learning，policy gradient，Proximal policy optimization

目录

[1 绪论 4](#_Toc41893497)

[1.1 研究背景 4](#_Toc41893498)

[1.2 研究出发点与研究目的 4](#_Toc41893499)

[1.3 强化学习研究概况 6](#_Toc41893500)

[1.4 主要研究内容 7](#_Toc41893501)

[1.5 论文的组织结构 8](#_Toc41893502)

[2 相关技术调研 9](#_Toc41893503)

[2.1 强化学习的数学模型 9](#_Toc41893504)

[2.2 深度强化学习无模型算法的基本脉络 10](#_Toc41893505)

[2.2.1 基于值函数的深度强化学习算法 10](#_Toc41893506)

[2.2.2 基于策略梯度的深度强化学习算法 15](#_Toc41893507)

[2.3 多模态与强化学习的初涉 17](#_Toc41893508)

[3 基于音乐节奏游戏的强化学习环境构建 19](#_Toc41893509)

[3.1 强化学习环境构建的基本要素 19](#_Toc41893510)

[3.2 音乐节奏游戏基本规则的介绍 19](#_Toc41893511)

[3.3 NaiveBandori强化学习环境构建与实现 21](#_Toc41893512)

[3.3.1 环境实现的关键参数设定与计算方式 21](#_Toc41893513)

[3.3.2 针对实验的环境简化 22](#_Toc41893514)

[3.4 小结 25](#_Toc41893515)

[4 基于近端策略优化算法的深度强化学习智能体的设计与实现 26](#_Toc41893516)

[4.1 近端策略优化算法介绍 26](#_Toc41893517)

[4.2 基于AC算法框架的BandoriNet神经网络架构 27](#_Toc41893518)

[4.3 单视频输入模型与音视频混合输入模型对照实验 28](#_Toc41893519)

[4.3.1 两类智能体的初步表现 28](#_Toc41893520)

[4.3.2 局部最优与On-policy 30](#_Toc41893521)

[4.3.3 最佳收敛性能 33](#_Toc41893522)

[4.3.4 可能的改进方向 35](#_Toc41893523)

[结论 37](#_Toc41893524)

[致谢 38](#_Toc41893525)

[参考文献 39](#_Toc41893526)

# 绪论

## 研究背景

强化学习(Reinforcement Learning，RL)是机器学习中监督学习、半监督学习、非监督学习之外的一种重要方法[1-3]。与其他机器学习方法相比，RL的显著差异在于RL是智能体(进行学习的主体)通过试错的方式与环境进行交互，以期在随时间变化的环境及其状态中得到最优回报，进而学习得到在一定状态的环境下智能体应采取的行动，进而归纳得出一种在某种或某类环境下能使得回报最大化的策略。

近年来深度学习(Deep Learning，DL)的成功，表明了DL对环境中的不同状态，尤其是状态的核心特征，具有良好的感知能力。因此，研究者们很自然地想到将DL中至为关键的深度神经网络(Deep Neural Network，DNN)应用到RL中去。事实证明，这一做法在RL领域取得了切实的成就，深度强化学习(Deep Reinforcement Learning，Deep RL)领域就此诞生。在DL的助力下，RL在维数灾难的问题上有了一种可能的解决方案，这使得RL可以使用经简单预处理后的数据，甚至只使用原始信息而非人工处理后的特征进行学习，从而实现端到端(end-to-end)的学习[4]。这大大地扩展了RL的应用领域，也提升了RL在许多完全或不完全信息博弈领域中的表现。

深度强化学习有着广泛而深刻的应用。自从深度强化学习2013年在多款Atari游戏上取得了超越人类的成绩之后，深度强化学习在各种游戏的智能体设计、计算机系统的优化、能源与交通的资源调度、广告推荐与投资组合配置、基础科学研究等领域的应用也随之兴起[5-15]。相应的，开源社区也在不断地为强化学习研究提供便利，开发了许多经典的训练环境，小到围棋、象棋、桥牌、麻将等历史悠久的棋牌类、OpenAI gym中的简单机械装置与Atari2600游戏、MuJoCo中的机器人运动，大到便于自动驾驶和无人机仿真的AirSim，甚至在王者荣耀、DOTA2、星际争霸、等复杂的多人网络游戏中也出现了便于强化学习研究的接口[16-17]。

## 研究出发点与研究目的

在各路研究者将提出的深度强化学习的新算法用于实验中时，无论实验中使用的是怎样的环境，研究者往往只会使用图像或视频作为端到端RL训练的输入，而声音却往往被忽略。抛开棋牌游戏和大多数机器人应用的仿真环境这种设计中就不存在声音的环境不谈，对于强化学习环境中更接近如今的电子游戏的那些环境而言，不论是对于Atari2600上的电子游戏史上的早期游戏，还是对DOTA2这种当代网络游戏，游戏中的绝大多数情况下，声音的有无对于玩家游戏操作的决策几乎没有任何影响。同时，游戏设计者往往将游戏中的声音仅仅用于提升人类玩家综合游戏体验这一方面，而非游戏决策的关键。基于这样的设计初衷，声音信号对于游戏决策而言往往是信噪比很低的一个信源，自然，DeepMind关于DQN的研究者和基于DOTA2训练的OpenAI 5的研发者在训练相应智能体时出于压缩模型的搜索空间和提升样本使用率的需要，都忽略了音频数据[6,18]。

音频在当下的强化学习训练中的普遍缺位并不意味着它是绝对可以舍弃的存在。强化学习作为一种学习人类决策方式的人工智能方法，它的学习方式就目前的效果而言仍然有必要从人类决策行为的动机获得参考。诚然，视觉信息在人生活中接受的信息总量中占比超过80%，但听觉信息对人做决策时的影响往往同样无可替代。一方面，语音交流、收听广播以及听到雷声去收衣服等场景下的决策一般仅依赖于听觉信息；另一方面，在驾驶汽车时注意其他车辆的鸣笛、结合教师演示听讲座等诸多场景下，视听融合的信息在指导人的学习和决策活动中缺一不可。

基于音乐节奏游戏构建的环境有望填补强化学习在音视频多模态训练中环境缺失的空白。音乐节奏游戏是动作游戏的子分类，其核心在于挑战玩家的节奏感，主要分为基于舞蹈的游戏和基于音乐的游戏两大类。本文以下所指的音乐节奏游戏专指基于音乐的音乐节奏游戏。这一类音乐节奏游戏中一般都会播放音乐，并在屏幕上显示玩家需要按下的按钮。按钮的运动方式一般都是游戏设计者人工设计的，而且往往与音乐的节奏有很强的关联性。玩家需要在精确的时间按下按钮，而游戏会根据玩家按下按钮的动作的准确性以及与游戏音乐节拍的同步性，给玩家奖励不同的游戏分数。

既然音乐节奏游戏的动作决策对音乐节奏的依赖性比较高，那么音乐节奏游戏中的音视频数据对于人类玩家玩好这类游戏而言不可偏废。一方面玩家需要根据视频中按钮位置、颜色等信息来决策行动，另一方面玩家需要在多次游玩中熟悉相应的音乐节奏，这样才能在快节奏的音乐中或是高难度的游戏场景中提高反应速度以应对运动或变换速递极高的按钮序列，从而获得较高的游戏分数。

基于以上种种，我们有理由相信，在某些环境下，尤其是使用音乐节奏游戏进行强化学习智能体的训练中，在原有的视频数据中加入音频模态，有可能能够对智能体的训练产生一定的影响。而随着移动通信技术的迭代进步、日新月异的高性能处理器“飞入寻常百姓家”中的桌面端和移动端设备内，以及虚拟现实技术的逐渐普及，多模态数据在网络传输上的应用基础、在解码运算上的硬件基础以及相应的智能应用的普及程度上日趋成熟，多模态数据“分而治之”将逐步转变为音视频综合应用、协同处理；而多模态数据的深度强化学习的应用前景同样未来可期。

## 强化学习研究概况

强化学习的灵感一般认为与心理学中的行为主义理论有关。在其发展初期，博弈论、运筹学、控制论等学科中都涉及了现今强化学习的一些问题。强化学习经过几十年发展，机器学习视角下的强化学习研究者目前往往聚焦于包括但不限于以下的多个方面。

1、无模型(Model-Free)算法：智能体学习时不关心环境的演化规律，直接以最大化回报值为目标进行学习。其中又根据智能体学习时采用的基本思路，按照以拟合Q函数的计算为主或是以直接优化策略，大致分为两类无模型算法；

2、有模型(Model-Based)算法。这种算法下智能体以学习环境的演化规律为主，在理解环境的运作动力后，可以直接给出最优策略。这种算法往往应用于行动规则已被人类所知的环境中，如AlphaGo使用有模型算法控制对围棋棋局进行的蒙特卡洛搜索，避免智能体在理解围棋基本的落子规则上花费大量时间以提升学习的效率[18]；

3、探索(Explore)与利用(Exploit)的平衡。智能体在有限的训练次数内，既需要多利用已学到的策略来最大化回报值，又需要适当地偏离当前策略以发现更优的策略——二者不可偏废。在一定的场景下，平衡好二者对于提升智能体的训练效率和收敛时的性能而言意义重大；

4、分层强化学习(Hierarchy Reinforcement Learning)。该分支研究的灵感源于人类社会的组织结构，即不同阶层中的个体在决策时观察到的状态的抽象程度和行动方式往往有着不小的差别。具体而言，智能体面对复杂问题时，可能需要分层强化学习元策略(meta-policy)。元策略是目标是根据环境本身的变化决定实际采用的子策略的使用方式；

5、安全性与可复现性。从不同于多数研究者的角度对当前强化学习的成果进行检验与评价；

6、模仿学习(Imitation Learning)与逆强化学习(Inverse Reinforcement Learning)。模仿学习可以通过复制先验数据的行动以提升训练效率，而逆强化学习则以学习表示回报值的合适的函数而非最优策略为目标；

7、面向现实场景的研究。包括在现实场景中的训练和应用，以及克服仿真与现实场景的差异带来的问题等。这部分的内容在无人驾驶、无人机编队、机器人配送服务等前景广阔的实际应用领域中尤为重要；

本研究为基于近年来发展迅猛的无模型算法的相关研究。无模型算法的详细发展历程将在第二章的技术调研中进行重点阐述。

## 主要研究内容

图1.1 本研究架构图示

本研究主要程序的运行架构如图1.1所示。环境部分包括音视频播放渲染模块、音频状态观测、视频状态观测以及行动判定模块，其中音频状态观测、视频状态观测、行动判定模块直接与智能体进行数据的交互。智能体观测状态后选择行动，Actor交互获得奖励，Critic根据状态计算出状态的值。这样的交互达到一定次数时，根据一定的强化学习算法，对智能体的参数进行更新。迭代该过程多次直到之智能体的性能收敛为止。本研究主要通过控制音频输入的有无进行对照实验，进而通过分析得出相应的结论。

## 论文的组织结构

本文主要从当今深度强化学习算法的多模态训练和应用都较为不完善的现状出发，先在当前音乐节奏游戏的一般机制的基础上，实现了能用于强化学习训练的、与音视频多模态相性极佳的简单的音乐节奏游戏环境NaiveBandori，再在该环境中实现了当前主流的深度强化学习策略梯度算法，并在单一视频输入和多模态输入两方面进行了多组对照试验。

本论文一共分为四章，各章节的研究内容如下所述：

第一章为绪论内容，主要分析了深度强化学习在多模态环境下的应用前景与实际意义，并大致介绍了强化学习的发展现状，归纳概括了本研究的主要内容。

第二章为相关技术调研，其中主要梳理了深度强化学习中无模型算法的发展历程，包括两大类无模型算法的起源以及最新发展现状，重点介绍了深度强化学习的开山之作DQN和现在发展迅猛的策略梯度算法核心的AC框架。最后介绍了与本研究相关的几种多模态强化学习的一些实验成果。

第三章主要介绍了本文工作中构建音视频多模态深度强化学习训练环境的部分。为了便于不熟悉音乐节奏游戏的读者理解，其中加入了对这一类游戏的进一步介绍。

第四章首先根据第二章中实现的强化学习环境，根据AC框架初步设计了一个用于深度强化学习算法的神经网络框架。之后基于近端策略优化(Proximal Policy Optimization, PPO)算法实现了智能体的构建与训练，并在单一视频输入和多模态输入两方面进行了多组对照试验，之后对实验结果进行了初步分析。最后对今后本文所进行的研究提出了多个可能的改进方向。

第五章中总结了本论文的工作，得出结论。同时还对深度强化学习在多模态场景下的应用前景进行了展望。

# 相关技术调研

## 强化学习的数学模型

RL主要由智能体(Agent)、环境(Environment)、策略(Policy)、奖励信号(Reward Signal)、价值函数(Value Function)、环境模型(Model，非必需)共六部分构成[19]。其中，智能体即执行强化学习算法进行学习主体，也是执行决策的主体。环境则是智能体进行学习时，对实际状态进行观测时的范围。比如在使用RL实现的象棋对弈程序中，棋盘及其之上双方棋子的位置就是环境。策略可以理解为智能体行动的方法论，它定义了智能体在特定时间下、特定环境下采取的行动。简单来说，策略是环境状态到智能体所采取的行动的映射。奖励信号指的是智能体在学习中采取行动之后，环境给智能体的反馈。它反映了环境在当前状态下，智能体所采取该行动所获得的回报。RL中的价值指的是从当前状态下算起，直到学习结束的全过程中，智能体所能获得的回报总量。相比奖励信号仅考虑单次的回报，价值考虑的是从当前状态直到最终状态（学习结束）的全过程。价值函数是针对某一状态下的所有“状态-行动”到智能体所能获得的回报总量的映射。RL的目标正是试图学习一种策略，使得长期的回报总量最大;而估计回报总量时，使用的正是价值函数。RL中的环境模型指的是能模拟环境反应的模型，即根据智能体的行为与环境当前的状态，能推演出环境状态的变化，进而给出奖励信号。环境模型在无模型算法中不是必需的。

形式化描述RL过程一般使用马尔科夫决策过程(Markov Decision Process，MDP)[20]。MDP包括状态集合、动作集合、转换函数和奖励方程。策略为到的映射,记为。智能体在学习时，先通过观测环境，得到时刻中的状态。此时若智能体执行动作，根据状态转换函数可以得到状态转为的概率；同时智能体获得相应的奖励。定义策略下状态的值，表示状态在该策略下的可以获得的最佳预期收益。RL的目标即可形式化地描述为:寻找一个最优策略，对所有策略使得：

其中记为。另外，定义状态-动作价值函数（Q函数）为从状态开始，智能体执行动作，随后的执行的动作依照策略，整个过程中的预期收益。不难看出：

其中使用表示智能体使用贪婪策略（即对于每个状态均采取预期收益最高的动作）进行策略选择时所获得的预期收益。故最优策略

因此，RL问题被转化为在给定的MDP下，最大化Q函数的过程。

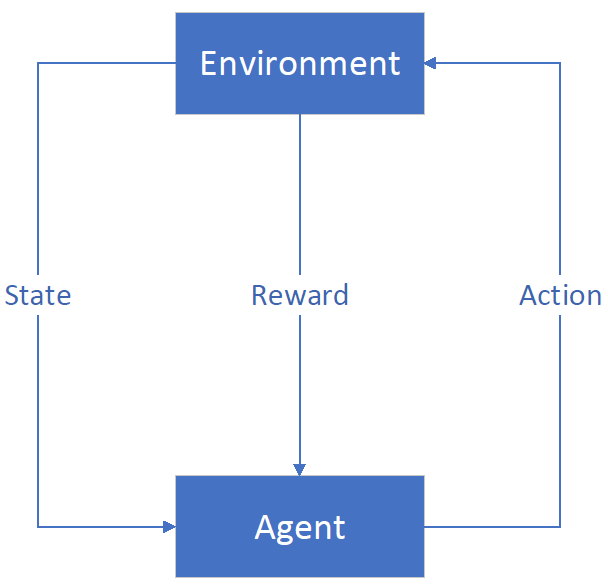


图2.1 无模型强化学习过程简图

## 深度强化学习无模型算法的基本脉络

将深度学习方法引入强化学习的尝试是强化学习领域内一次伟大的实践。在无模型的算法中，目前已经发展出基于值函数和基于策略梯度的两大类算法。

### 基于值函数的深度强化学习算法

基于值函数的DRL算法，指的是使用DNN来拟合值函数进行学习的算法。而实际中，则是通过DNN拟合Q函数并将其向最大值进行优化来拟合值函数。时序差分学习(Temporal-Difference learning)和Q学习(Q-learning)则分别是学习状态值函数和动作值函数的经典算法[21,22]。

1. Q学习算法

Q学习算法由watkins等人于1989年提出并于1992给出严格数学证明的值函数近似算法[22]。其算法流程如图2.2。

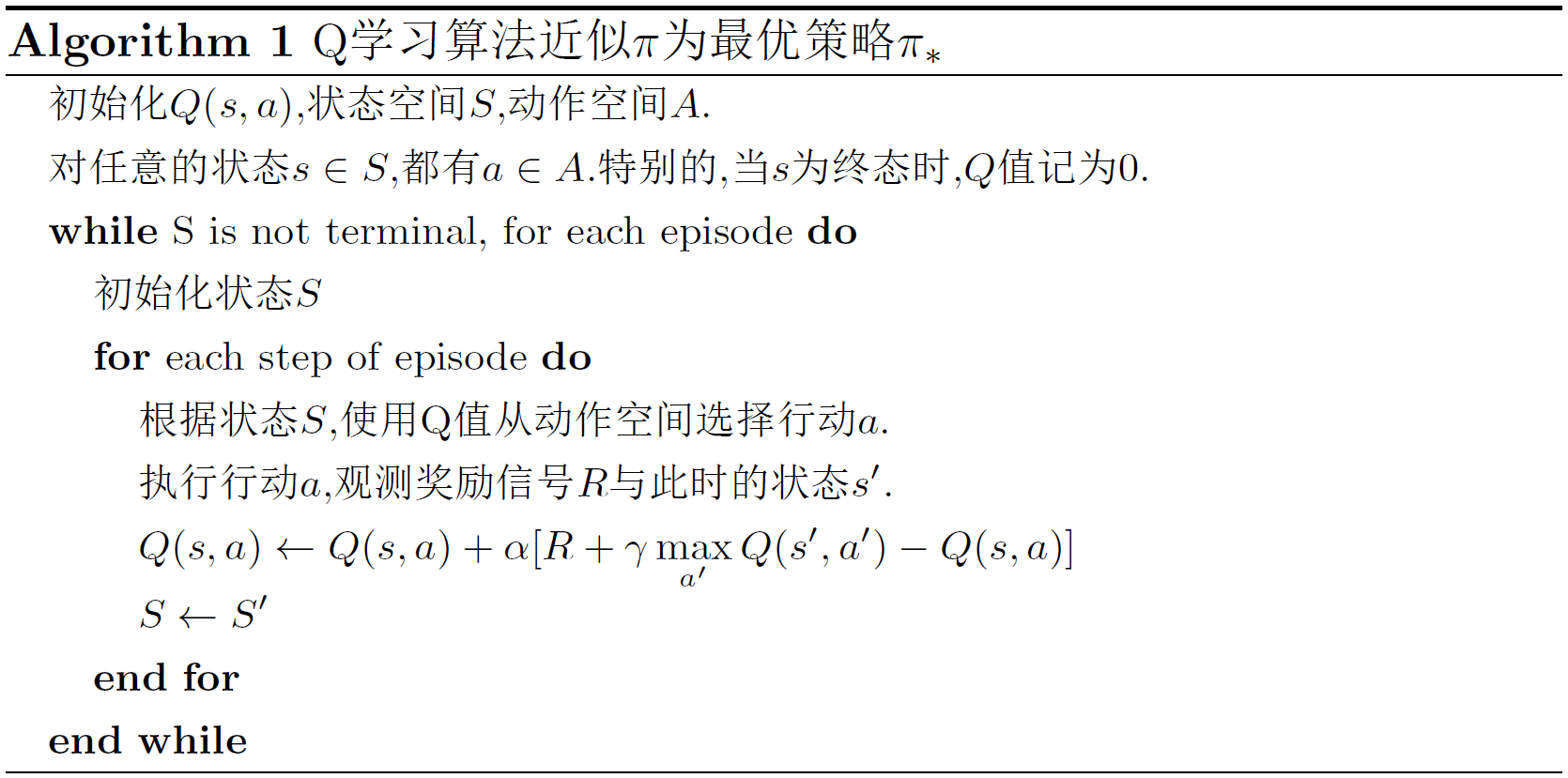


图2.2 Q学习算法流程

其中代表学习率，反应了未来状态下的行动带来的收益折算为当前状态下的收益的折旧因子。在若干轮迭代后，此时的Q函数即可近似为最优策略所对应的Q函数。

1. 深度Q网络(DQN)

研究者使用本节所述的Q学习算法中迭代计算Q函数时，往往因为状态空间或动作空间的中的维数灾难，导致求解Q函数的计算在时间和空间上的代价不能满足需求[23]。使用线性的函数拟合器，虽然可以使得计算资源的消耗在可以接受的范围内，但是在策略的效果上又有不少欠缺；而DQN诞生前就尝试使用DNN等非线性的函数拟合器来拟合动作值函数进行RL训练的研究者，则会陷入算法稳定性差、收敛困难的困境中[24]。DL与RL的结合一时停滞不前。

直到Mnih等人提出了DQN算法并将其应用于玩Atari2600游戏后，DL领域与RL领域的结合才产生了重大突破[18]。下面以玩Atari2600游戏为例，介绍DQN的神经网络拓扑结构及其训练方式。DQN以最新观测到的4幅游戏画面经灰度处理后作为输入。这些输入在经历以3层卷积层与两层全连接层构成的卷积神经网络的非线性变换后，直接输出当前状态下各个动作的Q值。图2.3为DQN的卷积神经网络拓扑结构的简图。此外，为了平衡探索与利用问题，智能体在选择行动时，可以按照-贪心(-greedy)算法进行选择。即按照的概率选择当前状态下最大Q值对应的行动，的概率选择其他行动。

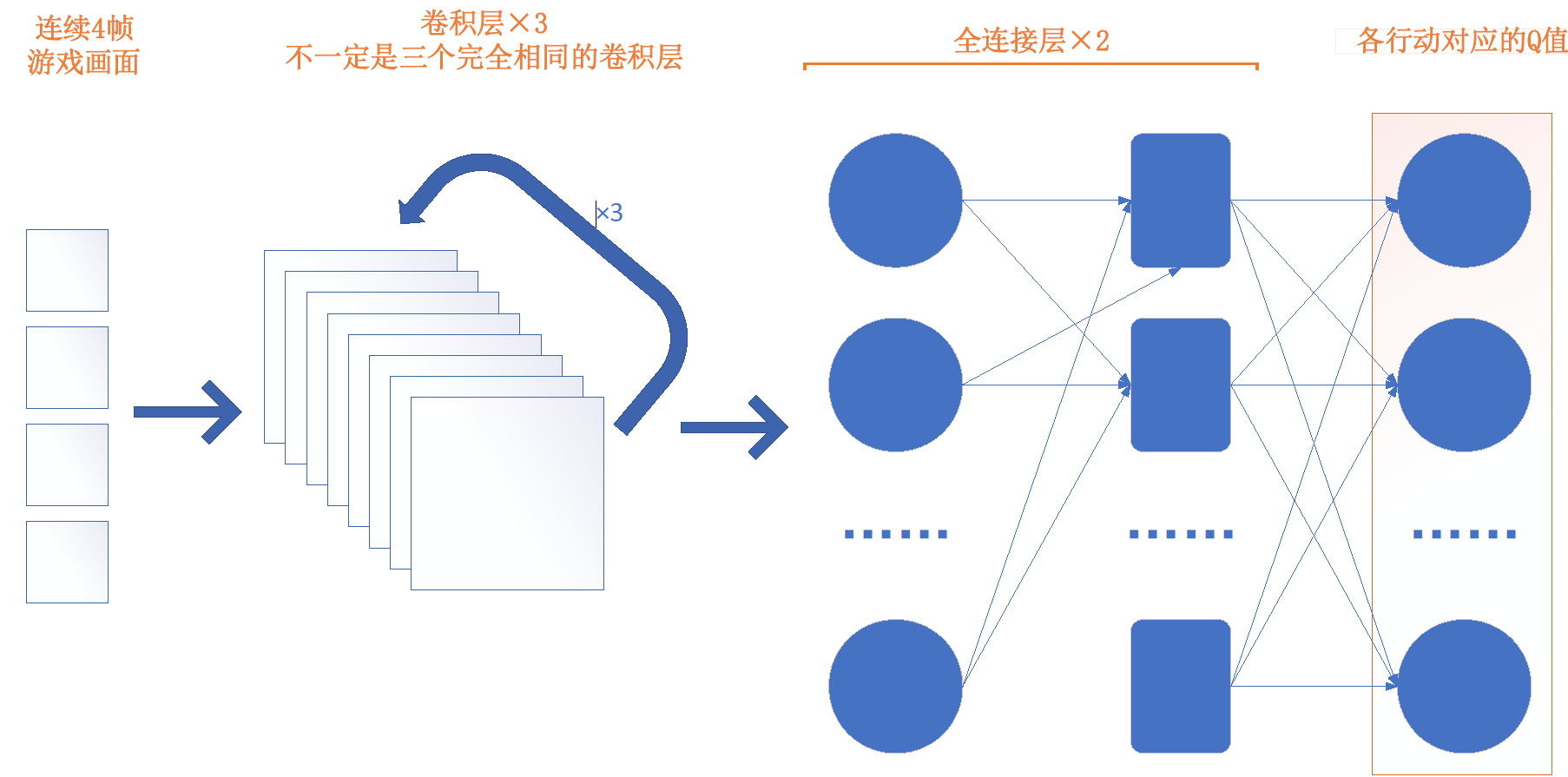


图2.3 DQN使用的神经网络的拓扑结构

为了解决本节所述的算法收敛困难的问题，DQN的训练过程，是在传统的Q学习算法上进行了一些有利于算法稳定性的改进得到的。

一是DQN使用了经验回放机制[25]。RL智能体在每次与环境交互时，将当前时刻的状态、采取的行动、下一时刻的状态、本次获得的奖励值作为一个经验四元组，存储在经验回放单元中。这个四元组记录了一次决策操作中获得的信息。随着智能体与环境交互次数的增加，经验回放单元中所记录的四元组数量同样会增加。每次训练神经网络时，会从经验回放单元中进行一次随机采样，选择出一个小批量的四元组样本，使用随机梯度下降算法更新神经网络的参数。这一做法能较好地保证训练样本之间的独立性，能满足神经网络训练对样本独立性的条件，从而提高了算法的稳定性。

二是DQN在计算Q值中使用了两个拓扑结构完全相同的卷积神经网络。其中，给出与环境交互的动作的网络，称为当前值网络。另一个网络称为目标值网络。在第轮迭代训练中，当前状态为，智能体的行动为时，下一个状态为，记为当前值网络的参数，则为参数为的当前值网络的输出；同理记为目标值网络的参数，则为参数为的目标值网络的输出。目标值网络的参数是不通过反向传播算法改变的[26]。

神经网络的优化目标表示为

因此，使用均方误差作为损失函数时，误差函数关于当前值网络的参数的表示为

对参数求偏导，得到

并通过反向传播算法，按一定的学习率更新当前值网络的权重[26]。

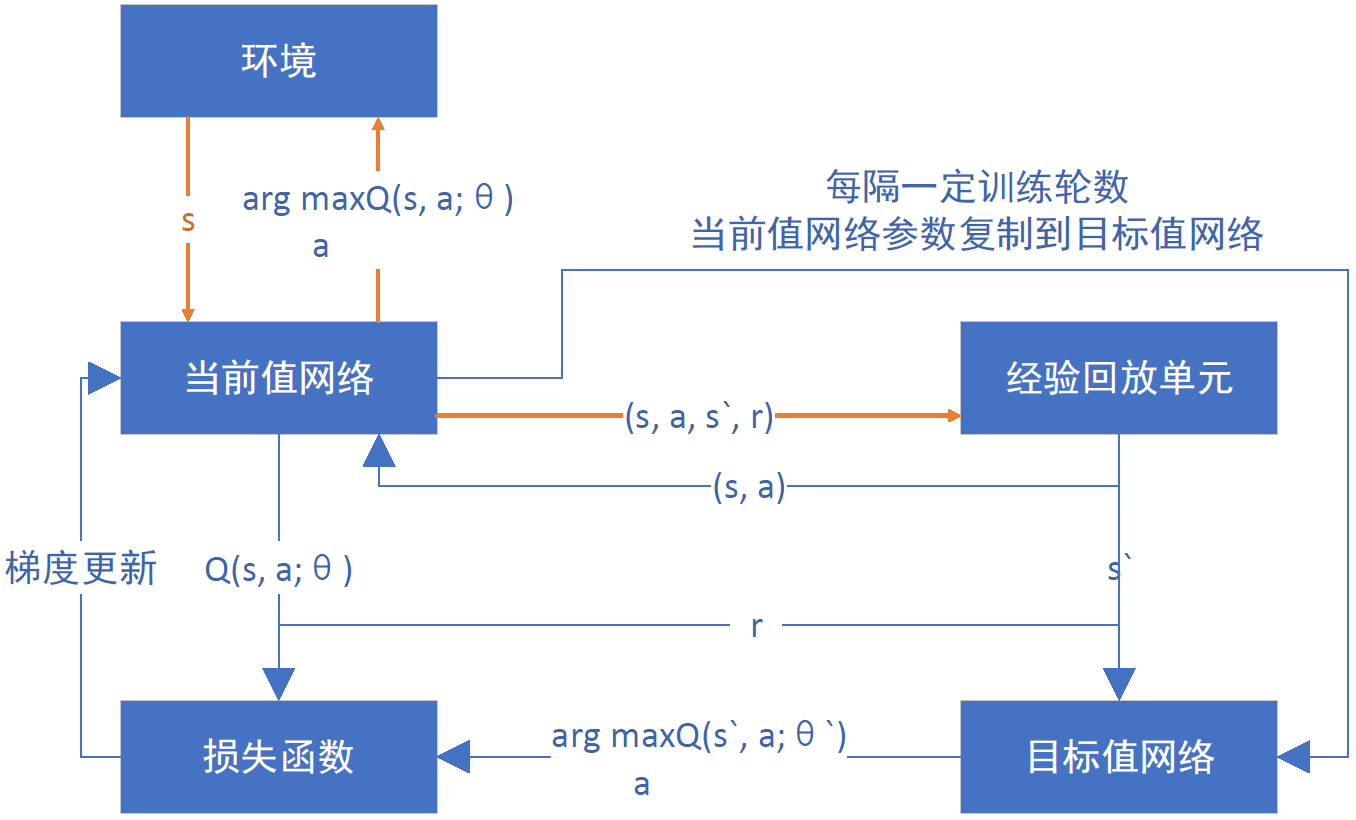


图2.4 DQN训练流程

总的来说，DQN训练时综合了目标值网络的输出与使用经验回放中的回报值作为真实值，当前值网络作为神经网络的预测值，将计算出的梯度更新到当前值网络中。在神经网络的训练迭代了一定次数后，将当前值网络中的参数复制并到目标值网络中，以替换目标值网络的参数。

可以看出，DQN的训练中在损失的计算上综合了目标值网络给出的预测值，而且目标值网络中的参数，在一定迭代次数内是保持不变的。这种通过含有旧参数的神经网络作为沟通了环境给出的回报值与当前值网络给出的预测回报值，降低了二者的直接相关性，降低了训练的难度，提高了算法的稳定性。训练流程如图2.4所示。

此外，DQN将通过将误差和奖励线性变换到较为合理的区间中，同时在一轮训练结束后，将神经网络中的梯度值压缩到合理的区间内，有效地缓解了梯度爆炸和梯度消失的问题，同样提高了算法的稳定性。实验表明，在49个不同的Atari2600游戏中，DQN在使用同样的模型参数，同样的训练方式，同样的网络结构的前提下，均能明显超越以往的所有算法的性能，达到了人类专家的水准。这展现了DQN良好的泛化性能，人们借此看到了通用型人工智能的一种可能的雏形[27]。

1. DQN的几种改进算法

DQN提出后，研究者们在DQN的基础上提出了多种改进算法。

优先经验回放(Prioritized Experience Replay，PER)：DQN中的经验回放，使用的是在经验回放单元中进行随机采样得到的。然而这种训练方式忽视了经验本身的分布特征。在DQN的基础上，Schaul等人将经验的特性纳入采样的考量，对更为重要的经验依概率多加采样进行训练[28]。结果表明，将优先经验回放机制应用到DQN中时，在41种Atari2600游戏中达到了论文发表时的最佳水准。

Hasselt等人根据DQN选取使下一状态中Q值最大对应的动作时会导致训练中Q值拟合时偏高的情况，结合Double Q-learning提出了Double DQN以缓解该问题[29,30]。

Dueling DQN将DQN的卷积神经网络提取出的特征分成两部分，分别拟合为Q函数中行动贡献的部分和Q函数中状态贡献的部分[31]。实践证明，这样拆分后的Q函数与原始的Q函数相比误差可以忽略不计，而这样的分开计算机制能加快收敛速度，尤其是当环境中出现许多相似值的行动的时候，Dueling DQN得到的策略更好。

Hausknecht等人针对不完全信息博弈问题中可观测的状态较少或是信噪比太低的情况提出了一种解决方案[32]。他们将DQN中第一个全连接层，替换为了长短期记忆网络单元(Long Short Term Memory，LSTM)，还减少了同时输入的画面数量至一幅，这样得到的网络称为深度循环Q网络(Deep Recurrent Q Network，DRQN)[33]。实验结果表明，可被智能体观察的状态的不完整程度较高的前提下，DRQN在性能上优于DQN，表明DRQN相比DQN更能适应在这种仅有部分状态可观测的复杂问题进行决策的情形。

### 基于策略梯度的深度强化学习算法

策略梯度算法在连续型的RL问题中有着广泛的应用。策略梯度，顾名思义，即将策略表示含有待优化的一组行动参数的一个映射，根据RL的训练过程中计算出的某一状态下智能体采取的行动的奖励，计算出全过程的回报总量，沿着梯度下降方向调整相应状态下的行动以最大化回报总量。重复这个过程即可逼近最优策略。这个方法已由Sutton等人于1999年给出了证明[34]。

策略有随机性和确定性之分。随机性策略指的是，在某一状态下，智能体依概率采取行动而；确定性策略则相反，对于某一状态，采取的行动都是固定的。类似的，策略梯度也分为随机性策略梯度(Stochastic Policy Gradient, SPG)与确定性策略梯度(Deterministic Policy Gradient， DPG)。SPG和DPG的计算公式分别由Sutton等人和Silver等人给出[34,35]。相比Q学习与DQN算法，策略梯度算法实现了从智能体获得的回报总量直接优化策略本身，是一种端到端的策略优化方法。而这种方法往往具有适用范围广，贴合现实世界，策略收敛效果更优的优势。

行动者-评论家(Actor-Critic, AC)算法框架：AC算法框架是基于策略梯度算法的一种应用，当前主流的策略梯度算法基本都是这个算法框架的一种实现。其主要架构如图2.5所示。

从图2.5中可知，AC框架中的智能体包含行动者(Actor)与评论家(Critic)。智能体与环境交互时，根据状态由行动者执行行动，根据评论家估计的奖励值使用SGD方法更新策略的参数；而评论家根据状态与行动使用TD等算法估计值函数Q。

这个过程本质上是行动者采取行动，由评论家度量该状态下行动者的行动的好坏，并给出具体的误差以便优化策略。如果说，行动者是学生，那么评论家就是会对学生的成绩给出具体指导意见的老师；AC算法框架同样体现了RL算法与人类学习过程中的相似之处。

DDPG算法：DDPG算法由Lillicrap等人于2015年提出[36]。该算法通过借鉴DQN算法的思想并将其应用到连续动作的决策空间中，在超过20种物理空间中模拟的问题有着健壮的表现。

DDPG算法是基于DPG和AC算法实现的。它以策略网络作为行动者，以Q网络作为评论家。训练中先根据当前环境的状态，由策略网络选择行为，并将环境原有状态、奖励、行为、环境的新状态作为元组存入经验回放单元中。进行一定轮次的训练后，从经验回放单元中随机采样一定数量的经验元组，并按照DQN的方法计算偏差。根据偏差，通过最小化均方误差的方法，更新评论家的梯度；再根据DPG的计算方式，更新行动者的梯度。

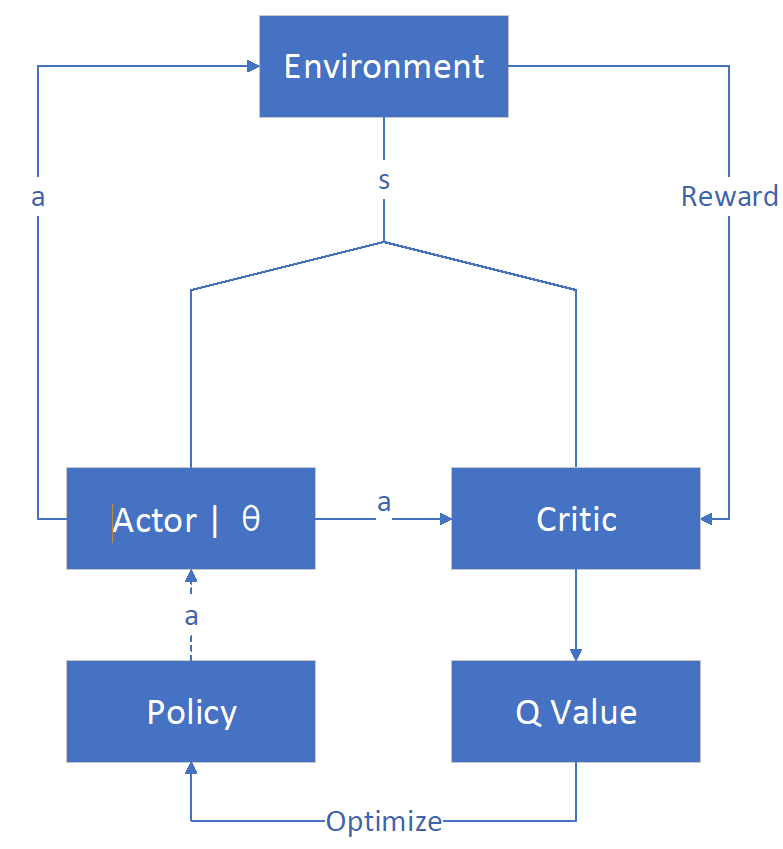


图2.5 AC算法框架

在更新梯度时，DDPG与DQN不同的是，DQN是每隔一定步长就进行参数的赋值，而DDPG是通过在原有参数基础上进行小步骤的迭代实现的。参数的更新方式按照：

其中为初始参数，为网络更新前的参数，为网络更新后的参数，为介于0与1之间的常量。此外，在进行决策时，为了平衡探索与利用的问题，DDPG在决策行动的选择上，引入了适当白噪声，有效地防止了过拟合现象。

通过DPG方法，DDPG算法解决了DQN在连续空间中的决策行动维数随着动作空间的精度增长而以指数增长的维数灾难；还能自行发掘出一些能和完全信息下的传统规划算法比肩的策略。此外，DDPG算法也是一种端到端的学习算法，但求得最优解所需的训练时间比DQN要短，这表明了DDPG算法的高效性。

A3C算法：DDPG算法及其改进均采用了经验回放机制，并通过随机采样的方式避免数据相关性带来的额外的有害特征。但同时也导致了耗费存储单元大和不能通过优化后的新策略从旧有经验进行学习，即只能采取离策略(Off-Policy)方式学习的缺点。

针对这些问题，Mnih等人提出了一个简单而轻量深度强化学习框架——A3C[37]。A3C算法给出的替代经验回放机制的方案，是在多核CPU上使用多线程技术建立多个智能体在多个环境不同的状态中异步地学习，但保证当前值网络和策略网络是共享的。多个智能体在与环境交互中产生的经验，会以异步的方式送到目标值网络和策略网络中去作为训练样本使用。

这种架构的优势在于以下三方面。一是可以借助多核CPU进行多种探索与利用策略，无需在这方面的平衡问题投入额外的计算资源；二是多个智能体始终采用不同的策略在变化情况不一的环境中学习产生的经验满足了训练使用经验的随机性的需求，可以采用Sarsa等On-Policy算法；三是通过网络共享减轻了对大量GPU计算资源的高度依赖，降低了A3C的训练和使用的门槛。

实验结果表明，A3C算法即使不采用多个GPU，也不采用分布式方法，在耗时上比之前基于大量GPU的算法更短，使用的计算资源也更少，却能在Arari2600游戏的效果上优于当时已有的算法。

## 多模态与强化学习的初涉

在强化学习的各种应用中，综合处理音视频多模态数据训练智能体的研究由于训练场景或环境构建相对困难、多模态数据分布差异较为显著而不利于强化学习智能体进行端到端训练等问题没能很好地解决而导致相关研究相对较少。

环境建构方面，Simon Brodeur等人于2017年开源了HoME——一种通过3D建模用于模拟多种环境下的居家环境[38]。而该项目已经有3年未更新且只支持2020年不再维护的Python 2.6及以上版本；Manolis Savva等人开源了名为MINOS的复杂环境下室内导航模拟器，在该平台上的多种强化学习算法均未能达到实验预期[39]。该平台中的多模态信源主要是源于位置各异的各类光学传感器，并非专指音视频。

实际应用方面，Jiaping Zhang等人提出了一种多模态的分层强化学习框架，在任务驱动的对话上下文中结合了图像数据和对话（文字）数据，联合学习多模态对话的特征表示以及分层对话策略，提高了对话任务的成功率和效率；此外还提出了状态适应技术将上下文对话特征集成状态表示中[40]。该框架在图像猜测游戏和问答游戏中的结果明显优于DQN的表现。

大阪大学的Ahmed Hussain Qureshi等人于2016年提出了多模态深度Q网络，分别用两个卷积神经网络处理图像的灰度帧和图像序列的深度帧，再通过两个线性层进行特征融合[41]。图2.6展示了使用了该算法的让机器人的在实际场景中的训练。机器人通过在公共场所中与访客交互收集数据，每日数据作为一个episode进行训练，14日后机器人成功与访客握手的概率超过了90%，表明该机器人学到了基本的社交技能。



图2.6 多模态深度Q网络训练机器人学会握手

# 基于音乐节奏游戏的强化学习环境构建

强化学习相较于其他的机器学习方法，最大的特色就是需要一个可供算法训练的智能体进行动态交互的环境，而一般不能是现成的数据集。深度学习框架在短短的十几年内，结合开源社区的庞大力量，将原本较为复杂的神经网络计算图、计算任务的调度、硬件协同的接口、权重更新过程的并行等实现层面上一个个重大议题全部简化成了几个接口，大幅降低了深度学习技术学习与使用的基本门槛；而强化学习中环境构建部分则更像一个较为传统的软件开发方面的内容，环境的工作方式需要研究者进行编码实现。

随着计算机仿真技术以及处理器的多核与超线程技术的不断进步，元件制程的不断提升，几十年前难以实现的强化学习仿真问题对于现在的设备而言，硬件资源已不再是瓶颈。相比强化学习早期的研究方式，现在的PC已经具备了构建并运行强化学习环境仿真的能力，且能通过快于现实速度的仿真提升单位时间内智能体与环境的交互次数，从而积累大量样本以对抗强化学习固有的样本效率低下的问题。本章将从强化学习环境构建的基本要素开始，逐步介绍本文研究中所使用的强化学习环境NaiveBandori的基本构建思路与部分重要的参数设置，为第三章的算法实验做足准备。

## 强化学习环境构建的基本要素

根据图2.1中描述的强化学习的一般性流程，环境需要具备的功能包括：

1、对所研究问题对应的实际情况进行模拟仿真。为了保证运行效率，以快于现实时间粒度的速度进行仿真往往很有必要；

2、能在运行中接收智能体执行的行动，并根据当前状态和行动计算后续状态以及该行动产生的奖励；

3、智能体进行训练所需的观测环境所得的状态，也应由环境产生并传递给智能体。

4、即时交互功能。在智能体训练完成后，需要进行即时的交互以验证算法或智能体的实际效果；

5、其他功能，如重置环境初态（随机或固定方式均可）便于多次反复训练等。

## 音乐节奏游戏基本规则的介绍

《Bang Dream! Girls Band Party》是一款典型的移动端下落式音乐节奏游戏。这类游戏的基本流程大致分为节奏设置和玩家进行游戏两部分。

节奏设置：游戏制作者根据某个特定的游戏曲目的节奏信息，在曲目播放到特定的时间位置时，设置游戏中按钮（或称“音符”，以体现其与节奏直接的密切关联）的大小、颜色、位置、种类等。从时间尺度上来看，相当于根据节奏人为设置了一系列的音符序列（或称“谱面”，与“音符”这一概念相对应）。有的游戏对同一曲目还有不同的难度设置，难易度一般以对玩家的反应速度的考验程度做区分。难度越高的曲目谱面组合，曲目的每分钟节拍数(Beats per minute, BPM)、单位时间内的音符数量、种类范围、音符位置变化的显著程度等项目往往会比相对简单的曲目谱面组合要高得多。由于谱面与曲目的制作均由人工完成，且二者有明确的对应关系，故音乐节奏游戏的随机性相对较小，这一点与多数其他游戏有显著区别。

玩家进行游戏：玩家在以一特定的曲目谱面组合进行游戏时，音符会根据游戏制作者设定好的方式进行运动。玩家需要在特定时间内，对特定的音符执行相应操作（多为点击、拖拽等），若操作正确玩家即可获得游戏分数，反之会失去游戏分数。有的节奏游戏会在玩家游玩中统计某次游戏中失去的游戏分数的总量，当这个总量达到一定值时会强制结束本次游戏。

如果音符的运动方式大致遵循从游戏画面上方向游戏画面下方运动的方式（或从原理玩家的一方运动向靠近玩家的一方），那么这一类音乐节奏游戏就被称为下落式音乐节奏游戏。游戏制作者为了便于玩家上手，越来越多的音乐节奏游戏都倾向于采用这种方式。



图3.1 音乐节奏游戏《Bang Dream! Girls Band Party》的游戏截图

以图3.1为例，画面左上方为玩家本局内已得到的游戏分数，右上角为本局内玩家失去游戏分数的一个统计度量。当音乐播放的进度到达一定值时，不同种类的音符也会从上到下运动到下方横线附近，玩家此时对这些音符进行相应的操作即可根据操作执行的时间点和节拍设定的时间点的对应程度获得相应的游戏分数。玩家若想提升某个曲目谱面组合下的游戏水平，则需要以相同的设置反复游戏，在游戏中加深对音乐节奏的熟悉程度，提升反应速度，尽可能地对所有音符都进行正确的操作。

虽然玩家可以关闭游戏声音，仅依靠肉眼观察音符的位置进行游戏，但由于音符出现的规律与音乐节奏相关联，且高难度下对反应速度要求更为苛刻，经验表明，不知道相应的音乐节奏时，玩家想取得较高的游戏分数会比较困难。

## NaiveBandori强化学习环境构建与实现

如图3.2所示，本研究已通过Pygame框架实现了本章2.2节部分《Bang Dream! Girls Band Party》的游戏中的有助于智能体决策的音视频方面的交互接口。游戏的具体机制中的各项参数参考了bestdori网站对该游戏环境的JavaScript-Pixijs实现[44]。本节中将对环境的实现细节中的关键内容进行进一步的阐述。

### 环境实现的关键参数设定与计算方式

为了便于智能体的训练，相比原版游戏，这里仅保留了音符以及提示音符运动方向的7个轨道，而对游戏背景图片、音符点击后即消失时产生的特效、获得分数时的提示等内容全部取消前端的渲染，内置到后台计算中。这样能在增加智能体观测的状态的信噪比的同时，也能提升游戏画面更新的性能。

计算某一时刻音符的位置时，主要根据游戏画面高度和音符速度这两个参数进行计算。游戏画面的长宽比固定为16:9。下文所述的时刻均指环境中音频播放的某一位置对应的时刻。以游戏画面左上角为坐标原点，横向向右为增大方向，纵向向下为增大方向。对判定时刻为的音符，在时刻时的计算方式如下。记。

首先根据音符速度（1~11之间的实数）计算音符从出现在游戏画面上到音符判定时刻所需要的时间。若，则按公式3.1~3.4计算音符的坐标。

其中为音符的图像素材的原始高度，可视为常量。若计算出的坐标满足音符仍在游戏画面范围内，则进行渲染。若计算出的坐标使得音符不在游戏画面范围内或则不显示音符。

坐标计算时，满足轨道约束方向即可，不再赘述。

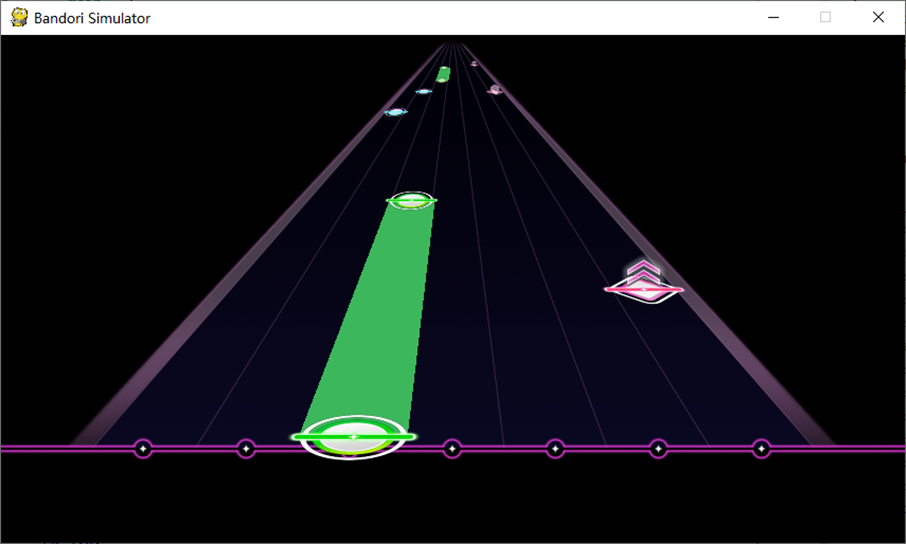


图3.2 《Bang Dream! Girls Band Party》游戏画面仿真

### 针对实验的环境简化

为了加快模型的收敛以提高实验的效率，用于本文的深度强化学习研究的NaiveBandori的环境构建则在此基础上进一步做了简化调整。具体在以下五方面：

1、原版游戏中使用了7个轨道的各种音符（包括图3.2所示的绿色长条，这意味着需要通过这个音符的判定，玩家需要在绿条经过下方横线期间在对应轨道上完成长按）。NaiveBandori仅使用中间的轨道，输出的状态中的视频部分也仅输出中间轨道附近以节省算力。此外音符不存在长按（图3.2中左起第三个轨道中的白色圆心与绿色圆环以及绿色横线组成的音符按钮），只存在单点音符（图3.2中左起第一个轨道中的白色圆心与蓝绿色圆环组成的音符按钮）和滑动音符（图3.2中最右轨道中出现的带箭头的红色音符按钮）。即，单点音符需要智能体进行一个点击操作；而滑动音符需要智能体进行一个滑动操作。相应的，奖励判定中也仅以这两种操作为准。

2、原版游戏中根据玩家操作与谱面设定二者的时间差设置了5个档次的得分。为了避免因为奖励值设置得不合理导致训练的模型表现不佳的情况，NaiveBandori的判定得分的机制简化成了对于每个音符在60fps内的3帧内完成正确点击则加10分（奖励为+10），否则扣除10分（奖励-10）。另外，为了促进智能体学习到没有音符时不应进行点击操作，奖励中同样设置了少量惩罚（-1）针对无可点击音符时的点击操作。

3、谱面不采用《Bang Dream! Girls Band Party》游戏中带有的。音符到达游戏画面下方横线的时间为生成的音调的时长的中间点对应的音频时长位置，但音符的运动方式与《Bang Dream! Girls Band Party》游戏机制保持一致（便于后续可能的迁移学习）。出于模型泛化能力的考虑，这里选择随机生成环境初始值（包括谱面和音乐）。每次环境被重置或被初始化，其谱面和音乐均随机生成。谱面生成时在音频总时长中随机取多个点，并保证任意两点对应的时刻差值大于200毫秒。每个点对应的音符从单点音符和滑动音符二者中随机选取。

4、音频同样不采用《Bang Dream! Girls Band Party》游戏中带有的，而是按照生成的谱面随机生成。音量默认值为10000。总时长是随机的，范围为8至12秒。其中每次生成的音调时长均为200毫秒以确保音调之间不存在重叠，音调之间的空隙使用静音填充。音乐的生成则从标准音阶中的7个音（由低到高C4~B4）随机选取。谱面中的音符与音调的对应关系为：单点音符则从C4~F4中等概率随机选取，滑动音符则从F4~B4中等概率随机选取。这样生成的音乐为一些完全无声的片段与7种音调中的一种且时长约为100毫秒的随机初始相位的正弦波交错拼接而成。为了增加环境的随机性以促进智能体学习到的策略具有更好的泛化性能，最后还需要对音频全程加入强度为音量的2%的白噪声以增强音频状态的随机性。

5、为了提升样本效率，原版中分数扣除量累计达到一定值后强制结束本局游戏的机制得以保留，促进智能体更多地在新一局的新环境内多多采样训练。本局游戏提前结束的条件是：每局游戏开始时预设生命值10，当预设的初始生命值加上本局游戏中获得的奖励值之和小于0时游戏强制结束。即，在没能获得任何正奖励的前提下，也能容许至多一次的对音符的错误操作。

基于这五方面的调整，NaiveBandori能在减少仿真环境系统资源占用的同时，进一步增加了环境的随机成分，有利于提高训练速度，也有利于智能体能在其中获得较为充分的探索机会。

另外，与OpenAI gym实现的Atari2600游戏环境在视频状态观测上不一致的是，OpenAI gym在输出视频作为状态观测值时，使用的是多张单通道的灰度图，而NaiveBandori输出的视频流为多张三通道彩色图。

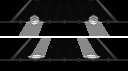


图3.3 灰度图下NaiveBandori视频流中的两帧的同一视频区域部分

在多数情况下，灰度图已经能满足观测状态的输入，这往往是因为游戏画面中的颜色仅仅起到美术效果的作用，而对游戏决策的影响不大；或者所需的颜色特征可以让智能体通过学习其形状进行弥补甚至替代。如DQN使用的就是连续4帧边长为84像素的正方形游戏画面经灰度处理后的图像。

但NaiveBandori环境中音符的颜色与智能体需要执行的正确行动是强关联的，而在低分辨率下音符的形状又非常接近。如图3.3所示，某局游戏中两个不同时刻的游戏画面的下半部分分别是两个滑动音符和两个单点音符通过游戏画面下方横线的情形。由于音符都具有中间颜色较亮，周遭颜色较暗的特点，且如果为了节省内存空间而使用低分辨率进行训练，则让卷积神经网络学习分辨单点音符和滑动音符可能较为困难。故状态观测的视频流选用彩色图而非灰度图。

音频状态观测为NaiveBandori所独有。由于Pygame框架本身是为游戏制作而打造的，故将其用于强化学习时的状态观测时会受制于当前的Pygame版本使用的SDL的多媒体底层框架。为了便于在有声卡或无声卡的各式平台上进行训练，在SDL框架下通过录音进行音频状态的观测的通用实现十分困难。此外，使用录音接口采集音频本身存在着大量而随机的时延，且音频播放无法以高于现实速度进行仿真，这不利于基于大量采样的训练，故音频采集通过对音频流二进制数据切片完成。若智能体决策的时间间隔为3帧，则截取的音频数据为最近3帧的音频序列。

## 小结

综合本章讨论的内容，本研究中使用到的环境的主要参数如下。

音符速度为9.0，游戏画面高度为90像素，音频与谱面生成部分按照第三章第三节中所述的默认值。

智能体的视频输入：前端渲染的帧率维持在60帧每秒左右，故大约16.67毫秒就要渲染一帧游戏画面。视频状态观测区域为中间轨道中下方50%的区域（约为宽23像素高45像素的矩形区域）。每三帧彩色的视频状态观测区域的图片为一组智能体可接收的视频输入。

智能体的音频输入：每生成一组视频输入时，以当前环境仿真中的音频应播放到的位置为止，之前的48毫秒的音乐采样数据，与3个视频帧的时长基本对应。

本文第四章将按照本章中设定的NaiveBandori环境参数来初始化环境以进行智能体的训练。

# 基于近端策略优化算法的深度强化学习智能体的设计与实现

## 近端策略优化算法介绍

近端策略优化(Proximal Policy Optimization，PPO)算法是策略梯度算法的一个重大改进，它于2017年被DeepMind和OpenAI几乎同时发表[42,43]。当前的PPO版本由于在多种环境中有着收敛稳定、效果好、计算量较算法思想相近的同类算法少得多的优点，现已成为OpenAI的研究中重要的算法基准。

记智能体的策略为，为策略的参数。策略梯度算法中一局游戏的运行记录发生的概率为

期望回报为

对采样中的所有进行归一化（线性映射到，尤其是回报值全部为正时避免算法陷入局部最优），再引入折旧因子（反映运行记录中越早的奖励值其可信度越低）后，将每个运行记录下的回报值记为，有

公式4.3使用的仅为环境给出的奖励值的简单算数和，没有体现出与状态和行动的联系。这里结合Q学习中的Q函数和状态的价值函数（AC算法框架中用Critic网络拟合价值函数），得到优势函数，用于替代中的用于求和的各项：

策略梯度算法的推导过程使用的是On-policy方法，即每次策略迭代更新到时，旧的策略的运行记录将不能用于新策略。这是为了避免学习到的策略无法收敛的重要机制，而这就导致了传统的策略梯度算法收敛缓慢、样本效率低下的缺陷。PPO算法提出了重要性采样机制用于弥补这一缺陷：通过对遵循某一概率分布的样本估计函数时，可以通过在与概率分布差异不大的概率分布上进行采样来完成。即

故期望回报的梯度

其中中的为按照第个运动轨迹进行的采样。上式已应用了。

为了确保策略更新的步长不至于过大（一方面避免误差严重累计，另一方面也为了确保重要性采样的适用条件），还要对根据上式计算出的损失量进行裁剪：即据此计算出的误差不能高于，也不能低于，超过这个范围的损失量将被裁剪到相应的区间端点。其中，一般小于0.2。记据此计算出的损失量为。

结合传统AC算法中使用的价值函数和信息熵造成的损失量，PPO算法更新策略时最终使用的损失函数为

其中和均为PPO算法中给定的超参数。由于以上多种机制确保了策略更新的步长较为平滑，故实际算法实现时往往可以通过对同一组样本学习至多两次而不至于过度偏离策略梯度算法的数学基础。

## 基于AC算法框架的BandoriNet神经网络架构

根据第二章中所述的环境中状态观测和接收动作输入的各项参数。神经网络的架构需要同时接受视频输入和音频输入。

如图4.1所示，如果智能体是需要同时接受音频和视频的，当设置智能体每3帧进行一次决策时（每秒60帧），每次的输入为3张彩色帧和48毫秒长度的音频，其中三帧彩色图进行3次三维卷积后经过全连接层输出到256维；而音频首先要通过快速傅里叶变换从时域数据转换为频域数据，再转为频域与分贝强度的数据并进行归一化，之后通过三层全连接层后再经过全连接层输出到256维。音视频两个256维拼接成512维后经过一个512的全连接层再输出。其中Actor和Critic的区别仅在输出层，Actor的输出的维数与智能体可执行的行动的数量一致，且需要经过Softmax函数映射到概率上来表示执行各个行动的概率；Critic只需要单个输出来拟合当前状态的价值函数。

不需要输入音频时，出于对照实验的需要，只需要去掉网络中处理音频部分的网络，再把输出层的前一层改为256维即可。



图4.1 BandoriNet网络拓扑图

## 单视频输入模型与音视频混合输入模型对照实验

以本章第二节所述的BandoriNet，参考了开源项目“天授”的PPO算法实现，在多个不同的随机数种子下进行了多次对照实验[44]。出于说明方便的考虑，本条中将使用音频和视频作为训练输入的智能体简称为A-Bandori智能体，仅使用游戏画面作为训练输入的智能体简称为G-Bandori智能体。

下文中所用曲线图的横轴均为训练轮数，纵轴的含义的对应分别为：rew=每局游戏的总回报值的平均，loss=反向传播时的总误差，clip=PPO算法对重要性采样计算的误差裁剪后的结果，ent=信息熵带来的损失，vf=Critic拟合的贡献的误差。

### 两类智能体的初步表现

实验结果表明，在一些特定的随机数种子的设定下，基于BandoriNet的PPO智能体均能收敛到效果较好的结果，即对于一局游戏中的多数音符，都能在指定时间内执行正确的操作。

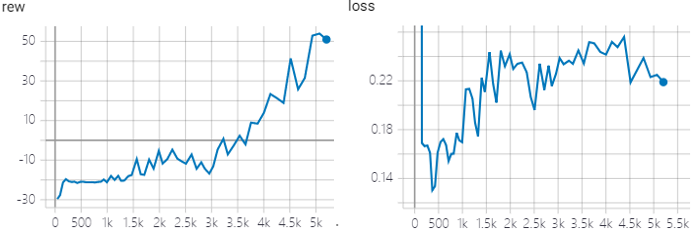


图4.2 A-Bandori智能体的一次训练结果（横坐标为训练轮数）

图4.2所示为A-Bandori智能体的一次训练结果，由于在此次训练中终止训练条件设置为测试中的平均回报值超过50即停止，故这次训练自此终止了。但是从Loss图中可以看出，Loss还未趋于稳定，终止训练时的Loss甚至不是已知范围的最小值。这个A-Bandori智能体此时应处于欠拟合状态，仍然存在通过训练提升性能的空间。

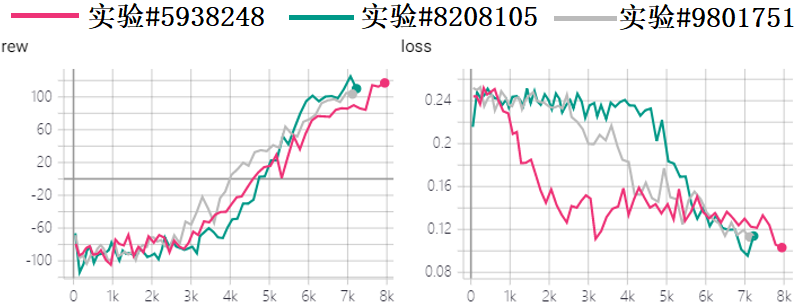


图4.3 G-Bandori智能体的三次训练结果（横坐标为训练轮数）

图4.3所示为G-Bandori智能体以三个不同的随机数种子初始化后的训练结果（其中红色曲线对应随机数种子5938248，蓝色曲线对应随机数种子8208105，灰色曲线对应随机数种子9801751）。鉴于本款内A-Bandori智能体训练中存在结束过早的问题，其后的训练中把终止训练的回报值门槛上调到了100以上。但我们仍然能发现这三个G-Bandori智能体的Loss还没有趋于稳定，训练仍然不够充分，仍处于欠拟合状态。

从图4.2和图4.3可以看出，基于BandoriNet的PPO算法构建的智能体，能对NaiveBandori环境提出的决策问题给出较为理想的方案。由于初始条件存在差异，无法对A-Bandori智能体和G-Bandori智能体之间在收敛速度（收敛所需训练轮数）的优劣下十分明确的结论。但在多次对照实验的训练中，G-Bandori智能体的收敛到理想情况的比例较高，据此推测G-Bandori智能体比A-Bandori智能体更容易收敛到理想情况。加上G-Bandori智能体无需音频输入，还有参数少，推断更快的优点。

### 局部最优与On-policy

不同于前款所述的实验结果，在其他的许多随机数种子下，智能体往往无法收敛到使得总回报为正的状态。

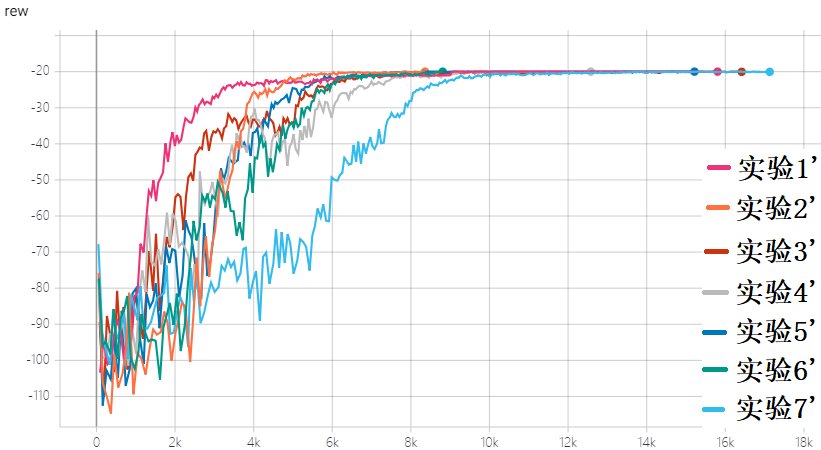


图4.4 两种Bandori智能体的多次训练结果（横坐标为训练轮数）

从图4.4可以看出，不管哪种智能体，许多情况下或早或晚总回报值收敛到-20，且此时对应的Loss均低于0.1，且数千个训练轮数内保持相对稳定。这说明这些Bandori智能体收敛到了一个局部最优值。-20这个数的来源与环境给的奖励值有关。经过查证，这些智能体在测试时都是在错过了开头的两个音符的操作时间窗口后（而不是弄混了点击与滑动操作）生命值从10变为负值，由环境强制终止了游戏。

甚至在前款所述的结果较好的G-Bandori智能体所使用的随机数种子下，对应的A-Bandori智能体的表现却不尽人意。

收敛到局部最优回报值-20的原因分析：考虑到环境对冗余的点击操作有奖励值为-1的惩罚，而环境每次运行中音符在时间上的密度不够大——换言之多数情况下智能体的正确行动应该是什么都不做。基于这一事实，算法在采样时，很可能对这一类的“状态-行动”的组合的采样过度了，而对正确点击并获得正的奖励值的“状态-行动”组合的采样不足。还有可能智能体与环境交互程度不够充分，或是没能在训练早期产生足够多的正奖励值的样本，加上PPO作为一种On-policy算法相对Off-policy算法更稳定，这也导致了算法陷入局部最优后难以从中跳出来。以上种种原因都有可能导致智能体在决策上过度地走向了“避害”——即使是在需要点击或是滑动操作的时候，也选择了什么都不做。最终智能体在测试中错过了开头两个音符的操作的时间窗口，总回报值-20，游戏强制结束。

同一随机数种子下A-Bandori智能体陷入局部最优但G-Bandori智能体能顺利收敛到较为理想的值的结果，很可能与多模态神经网络BandoriNet的结构的不合理之处有关。图4.6中A-Bandori智能体训练中的前1000次迭代中全都出现了Loss先急剧上升再急剧下降到相对正常的数值的现象。推测这部分的Loss在计算中遇到了无穷大的数值，而这些严重偏离期望数值的训练很可能会让神经网络中的权重也发生严重偏离。这种情况下的随机数种子的设置很可能对初始化神经网络的权重起不到应有的作用。

根据本章第一条中对PPO算法的介绍，Loss的来源有Critic网络、裁剪后的重要性采样、信息熵三方面。根据图4.7和图4.8可以得出，使得Loss如此变化的主要贡献源是裁剪后的重要性采样造成的误差。

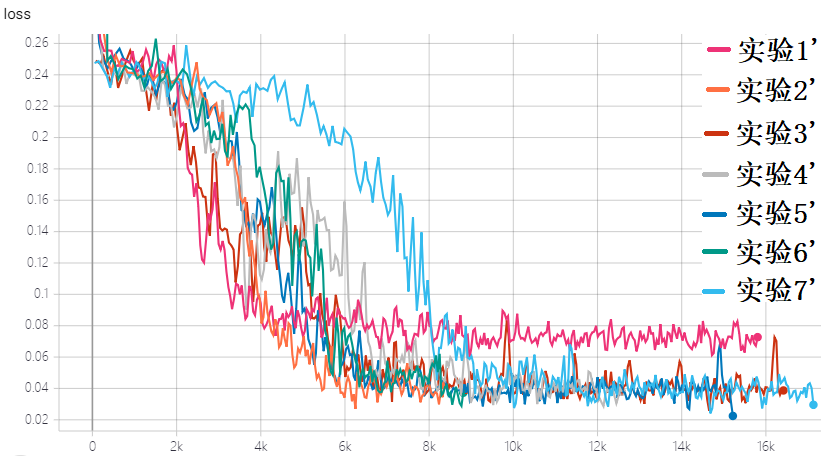


图4.5 两种Bandori智能体的多次训练结果对应的Loss（横坐标为训练轮数）

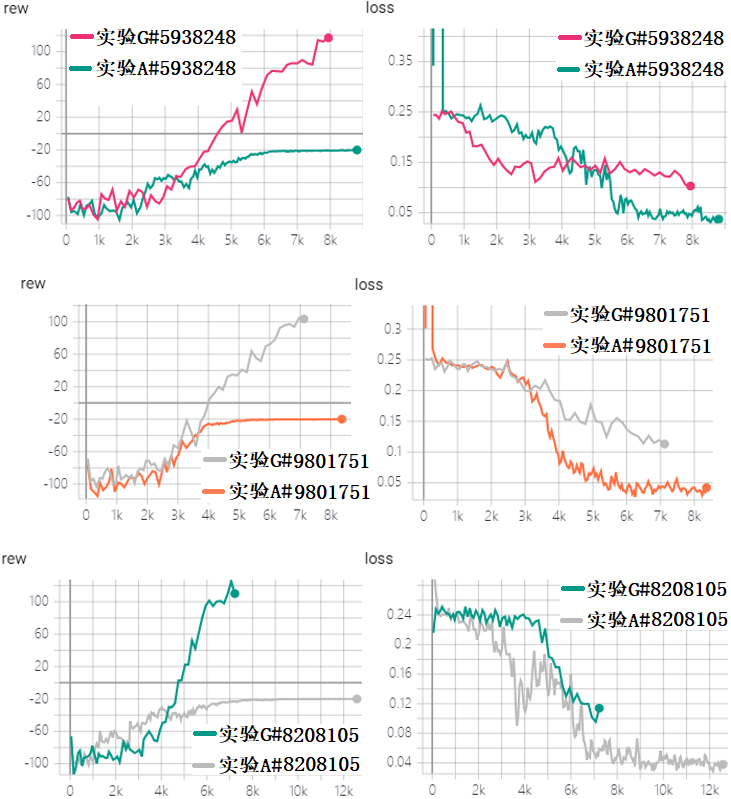


图4.6 随机数种子分别为5938248、9801751和8208105的两种智能体的训练结果



图4.7 三个陷入局部最优的A-Bandori智能体的信息熵误差与Critic误差

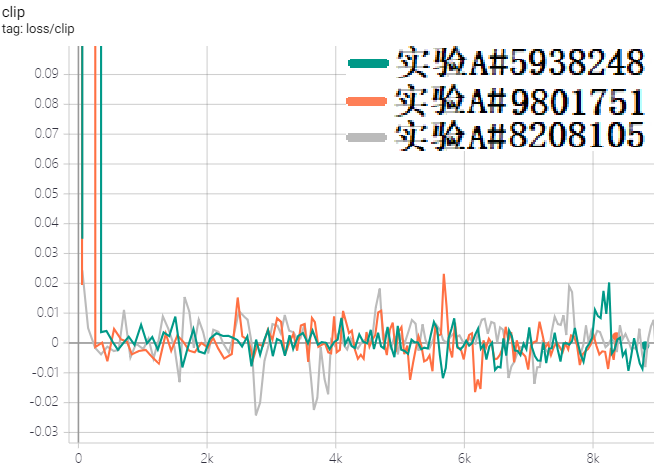


图4.8 三个陷入局部最优的A-Bandori智能体的裁剪后的重要性采样误差

从总回报的初始值中也可以看出BandoriNet的不合理之处。对比前款图4.2中收敛的A-Bandori智能体的回报值变化曲线，初始值仅为-30；而本款内的A-Bandori智能体的回报值变化曲线的初始值均低于-60。据此推测，回报初始值较大的智能体，其神经网络的初始化结果很可能十分理想，算法很容易就将神经网络权重朝着使总回报为正的方向进行更新，最终表现良好：这一事实很可能表明BandoriNet对良好的初始化的依赖程度较高。而对初始化依赖程度高恰恰说明了BandoriNet与PPO在处理多模态决策问题存在不合适之处。

### 最佳收敛性能

为了对比A-Bandori智能体和G-Bandori智能体的在最优条件下的性能，本款中对本条中前款所提及的收敛结果较为理想但仍然欠拟合的A-Bandori智能体和G-Bandori智能体各一个载入PPO算法中，但不设置随机数种子分别继续进行训练。结果如图4.9与图4.10所示，其中蓝色曲线和红色曲线分别代表A-Bandori智能体和G-Bandori智能体的最终收敛时的实验结果。



图4.9 A-Bandori智能体和G-Bandori智能体的在最优条件下收敛时的损失值

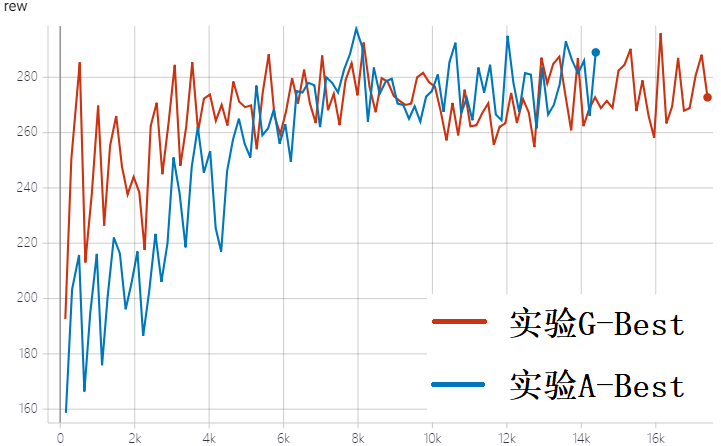


图4.10 A-Bandori智能体和G-Bandori智能体的在最优条件下收敛时的回报值

从图4.9可以中看到，二者的Loss在经过充分训练后都趋于稳定，尤其是信息熵贡献的误差相比其他几项而言以及相当小了。此时的智能体的表现也十分优异。虽然NaiveBandori生成的一局游戏中，音频长度和音符数量都具有一定的随机性，游戏之间的理论最高分不尽相同，但经过测试，绝大多数情况下二者都能达到一局游戏中的理论最高分值。据此推测，这两个智能体在跨过了前款所提到的局部最优的结果后，由于PPO的On-policy性质，这使得智能体在进一步的学习中不容易丢失已有成果，反而能在此基础上进一步优化，最终达到一个近乎正反馈的效果，使得智能体的表现提升迅猛。

### 可能的改进方向

基于前文的分析，这里拟提出几种可能能够改善智能体的性能的方案。

1、在训练的不同阶段根据情况分别使用Off-policy和On-policy

智能体在训练初期可能面临探索不足的问题，对此可以考虑采用Off-policy的算法进行初步训练。借助优先经验回放的思想，在使用Off-policy算法探索“状态-动作”较为充分之后，对更为重要的样本进行优先采样训练。如果可能的话，在训练取得一定成果后（如成功跨越了局部最优的常见门槛），可以考虑更换训练方式为On-policy加大对当前策略的利用程度，以此加快收敛到最终数值的过程。但设计机制使得Off-policy平稳过渡到On-policy将是一个难点。

2、引入注意力机制

近年来，注意力机制因为其思路简洁而对算法性能提升明显的优点，在图像分割已经自然语言处理等领域，尤其是涉及到语义分割的问题上得到了推广应用。而如果对BandoriNet的按照音频部分和视频部分的划分使用注意力机制，可能也能带来性能上的提升。此外，注意力机制还能帮助研究者对神经网络的结构进行进一步的分析。

3、借鉴多模态机器学习中的方法

BandoriNet处理多模态数据的方法其实还是将音视频数据分开进行了，没有充分运用到多模态机器学习领域内已有的算法成果，如基于子空间的聚类、基于语义匹配的方法等等。

4、音频预处理方式的选择与组合

本文仅仅尝试了使用快速傅里叶变换来处理音频的方式，但音频预处理的方法除此之外包括但不限于预加重（Pre-emphasis）、加窗（Windowing）、梅尔滤波器（Mel Filter Bank）、MFCC等等。这需要研究者基于音频本身的特征进一步分析实验以确定音频预处理的方式。

5、BandoriNet拓扑结构的调整

音频作为序列数据，一方面同样可以对其进行卷积操作以提取特征；另一方面也可以使用循环神经网络来提取序列特征；还可以将音视频按帧进行模态融合提取特征后，再使用循环神经网络来处理。

6、生成经验作为预训练

如果仅仅想加快新模型的训练，那么可以借助目前已经训练好的模型与环境的交互产生的“经验”。这种方式产生的经验对应的奖励几乎全都是非负值。如果在新模型的训练中加入这类“经验”，那么新模型在训练中很可能可以更好地克服探索不足的带来的缺陷。

# 结论

深度强化学习近年来成果斐然，自DQN在多种Atari2600游戏上达到人类专家级水平起，到AlphaGo和AlphaZero在围棋水平上达到了人类从未达到过的高度，再到如今深度强化学习在多种交叉领域内大放异彩——尽管深度强化学习如今还有各种各样的痛点，但它的魅力已经吸引了越来越多的研究者。本文在回顾深度强化学习的发展历程与关键算法及其相关应用的基础上，从当前深度强化学习中关于音频的训练较少的现状出发，参考了其他研究者在这个方面的成果，推断出强化学习的仿真环境对音视频协同数据的训练的支持不足的结论。

在此基础上，本文找到了一类对音视频协同数据的智能体训练的支持相对较优电子游戏——音乐节奏游戏，并基于当代的一款音乐节奏游戏构建了一个相对简单开源的强化学习环境NaiveBandori。并在该环境下设计支持音视频数据输入的神经网络BandoriNet，基于当前的一种主流深度强化学习算法PPO进行了智能体的训练。以音频数据是否在训练中输入智能体进行了一系列的对照实验，最终得到了以下几个结论：

一、基于BandoriNet的PPO算法训练的智能体，无论是否有音频输入，均有可能收敛到较为理想的状态，即能在多数游戏中获得接近游戏自身的最高分的奖励分数；

二、基于BandoriNet的PPO算法构建的智能体、无论是否有音频输入，均有较大可能性收敛到总回报值为负的局部最优解，智能体的最终表现对初始化的依赖程度较高；

三、BandoriNet中提取音频数据的子模型，在某些情况下会对智能体收敛到较为理想的状态起到负面作用。即存在部分情况，接收音频输入的智能体的性能相对较差；

四、融合音视频进行的深度强化学习训练，要达到较好的性能，还需要在改进网络结构、数据预处理方式、多模态机器学习方法应用等方面进行大量的探索与实验。

随着通信技术的迭代更新以及设备智能化水平的提升，多模态数据在人们的生产生活中日益联系紧密。与此同时，多模态数据也在日益增强人们的生活体验。在智能化、集成化的潮流之中，深度强化学习的应用正方兴未艾。深度强化学习在多模态环境下的应用，定能为实现人类美好生活的愿景贡献力量。

# 致谢

2020年对所有人乃至人类历史而言注定是不平凡的一年。疫情深刻地改变了我们的生活，改变了世界的样貌。在这样一个特殊的时期进行的“居家毕业设计”，足以称得上宝贵而难忘的经历了。

在这几个月的居家隔离的毕业设计工作中，我要感谢我的父母。自从我读大学以来，从未有如此机会与父母相伴这么久的时间。而在这几个月的时间内，父母在物质上和精神上对我完成毕业设计的支持难以用言辞来形容。

同时，我还要感谢童超老师对我的悉心指导。虽然疫情使我们天各一方，但特殊时期内童老师仍然保持了他勤勉严谨的作风，通过网络对我们的毕业设计工作进行了悉心地指导。童老师总是反复确认我们确实没有问题了才会结束视频会议，绝不把问题留到第二天。虽然我在毕业设计的最后关头出现了比较严重的技术性意外，但是在童老师的关心、支持与鼓励下，我成功走出了困境完成了毕业论文。

其次，感谢Sutton等老前辈在强化学习领域数十年如一日的耕耘，没有他们我的毕业设计绝对无法达到这样的完成度。

再次，感谢我的室友们。虽然我们选择的专业不尽相同、爱好人各有别，但是我们在求同存异上付出的努力，使得我们的友谊经受住了种种考验，时至今日依旧牢不可破。

另外，感谢马克思主义思想在我多次思想上出现懈怠、或是畏难情绪涌上心头时帮助我数次打牢思想基础，支撑着我在解决技术性意外中保持强健的精神。

最后，感谢过去的、现在的世界人民。在历史的长河中，人民用生命创造出的巨大物质财富和精神财富代代相传，塑造了这个世界，也塑造了我自己。历史与人民把时代的接力棒传到我们手中，我们有责任也有能力，将人类文明的火种传承下去。

# 参考文献

[1] Norvig P R, Intelligence S A. A modern approach[M]. Prentice Hall, 2002:33-49

[2] Mohri M, Rostamizadeh A, Talwalkar A. Foundations of machine learning[M]. MIT press, 2018:9-41.

[3] Barlow H B. Unsupervised learning[J]. Neural computation, 1989, 1(3): 295-311.

[4] 刘建伟, 高峰, 罗雄麟. 基于值函数和策略梯度的深度强化学习综述[J]. 计算机学报, 2019 (2019 年 06): 1406-1438.

[5] Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of go without human knowledge[J]. Nature, 2017, 550(7676): 354-359.

[6] Berner C, Brockman G, Chan B, et al. Dota 2 with Large Scale Deep Reinforcement Learning[J]. arXiv preprint arXiv:1912.06680, 2019.

[7] Zhang C, Patras P, Haddadi H. Deep learning in mobile and wireless networking: A survey[J]. IEEE Communications Surveys & Tutorials, 2019, 21(3): 2224-2287.

[8] Mukwevho M A, Celik T. Toward a smart cloud: A review of fault-tolerance methods in cloud systems[J]. IEEE Transactions on Services Computing, 2018.

[9] Tang D, Li X, Gao J, et al. Subgoal discovery for hierarchical dialogue policy learning[J]. arXiv preprint arXiv:1804.07855, 2018.

[10] Glavic M, Fonteneau R, Ernst D. Reinforcement learning for electric power system decision and control: Past considerations and perspectives[J]. IFAC-PapersOnLine, 2017, 50(1): 6918-6927.

[11] Liu X Y, Ding Z, Borst S, et al. Deep reinforcement learning for intelligent transportation systems[J]. arXiv preprint arXiv:1812.00979, 2018.

[12] Zhang S, Yao L, Sun A, et al. Deep learning based recommender system: A survey and new perspectives[J]. ACM Computing Surveys (CSUR), 2019, 52(1): 1-38.

[13] Liang Z, Chen H, Zhu J, et al. Adversarial deep reinforcement learning in portfolio management[J]. arXiv preprint arXiv:1808.09940, 2018.

[14] Pan Y, Farahmand A, White M, et al. Reinforcement learning with function-valued action spaces for partial differential equation control[J]. arXiv preprint arXiv:1806.06931, 2018.

[15] Popova M, Isayev O, Tropsha A. Deep reinforcement learning for de novo drug design[J]. Science advances, 2018, 4(7): eaap7885.

[16] OpenAI.Gym: A toolkit for developing and comparing reinforcement learning algorithms[CP/OL]. https://gym.openai.com/,2020-05-16.

[17] Microsoft AI & Research.Airsim: Open source simulator for autonomous vehicles built on Unreal Engine / Unity[CP/OL].https://microsoft.github.io/AirSim/,2020-05-22.

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

[19] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. MIT press, 2018:1-25

[20] Wiering M, Van Otterlo M. Reinforcement learning[J]. Adaptation, learning, and optimization, 2012, 12: 3.

[21] Sutton R S. Learning to predict by the methods of temporal differences[J]. Machine learning, 1988, 3(1): 9-44.

[22] Watkins C J C H, Dayan P. Q-learning[J]. Machine learning, 1992, 8(3-4): 279-292.

[23] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述[J]. 计算机学报, 2018, 41(1): 1-27.

[24] Tsitsiklis J N, Van Roy B. An analysis of temporal-difference learning with function approximationTechnical[J]. Report LIDS-P-2322). Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Tech. Rep., 1996.

[25] Mahadevan S, Connell J. Automatic programming of behavior-based robots using reinforcement learning[J]. Artificial intelligence, 1992, 55(2-3): 311-365.

[26] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. nature, 1986, 323(6088): 533-536.

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

[28] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay[J]. arXiv preprint arXiv:1511.05952, 2015.

[29] Hasselt H V. Double Q-learning[C]//Advances in neural information processing systems. 2010: 2613-2621.

[30] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning[C]//Thirtieth AAAI conference on artificial intelligence. 2016.

[31] Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning[J]. arXiv preprint arXiv:1511.06581, 2015.

[32] Hausknecht M, Stone P. Deep recurrent q-learning for partially observable mdps[C]//2015 AAAI Fall Symposium Series. 2015.

[33] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[34] Sutton R S, McAllester D A, Singh S P, et al. Policy gradient methods for reinforcement learning with function approximation[C]//Advances in neural information processing systems. 2000: 1057-1063.

[35] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]. 2014.

[36] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.

[37] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//International conference on machine learning. 2016: 1928-1937.

[38] Brodeur S, Perez E, Anand A, et al. Home: A household multimodal environment[J]. arXiv preprint arXiv:1711.11017, 2017.

[39] Savva M, Chang A X, Dosovitskiy A, et al. MINOS: Multimodal indoor simulator for navigation in complex environments[J]. arXiv preprint arXiv:1712.03931, 2017.

[40] Zhang J, Zhao T, Yu Z. Multimodal hierarchical reinforcement learning policy for task-oriented visual dialog[J]. arXiv preprint arXiv:1805.03257, 2018.

[41] Qureshi A H, Nakamura Y, Yoshikawa Y, et al. Robot gains social intelligence through multimodal deep reinforcement learning[C]//2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids). IEEE, 2016: 745-751.

[42] Heess N, TB D, Sriram S, et al. Emergence of locomotion behaviours in rich environments[J]. arXiv preprint arXiv:1707.02286, 2017.

[43] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.

[44] Burrito.Bestdori[CP/OL].https://bestdori.com,2020-05-21.

[45] Tsinghua Machine Learning Group.Tianshou (天授): An elegant, flexible, and superfast PyTorch deep Reinforcement Learning platform[CP/OL]. https://github.com/thu-ml/tianshou,2020-05-21.