游戏机器人的研究与应用

深奇智慧联合创始人　高 扬

AI时代的移动技术革新
Era of AI: Innovations in Mobile Technologies

NPC的驱动

低级

中级

高级

特高级

低级NPC

杂兵

- 事先编好行进路线，用事件驱动其出现或生效。
- “见人就打”

中级NPC

单机游戏群战中的配合型NPC

- 有一定事先编好的策略驱动
- 会一定的事先设定下的应变能力
- 场景极为单一，确定

高级NPC

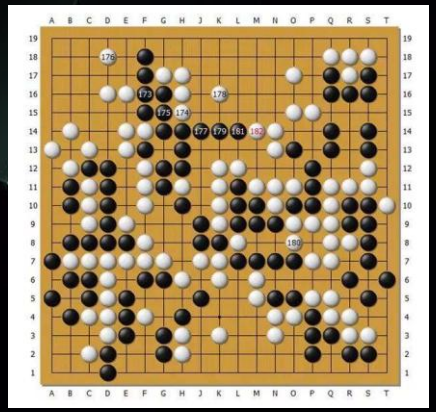
网游中的高级团战英雄

- 复杂的且变化多样的场景
- 动作丰富且复杂
- 评价模式相对复杂

特高级NPC

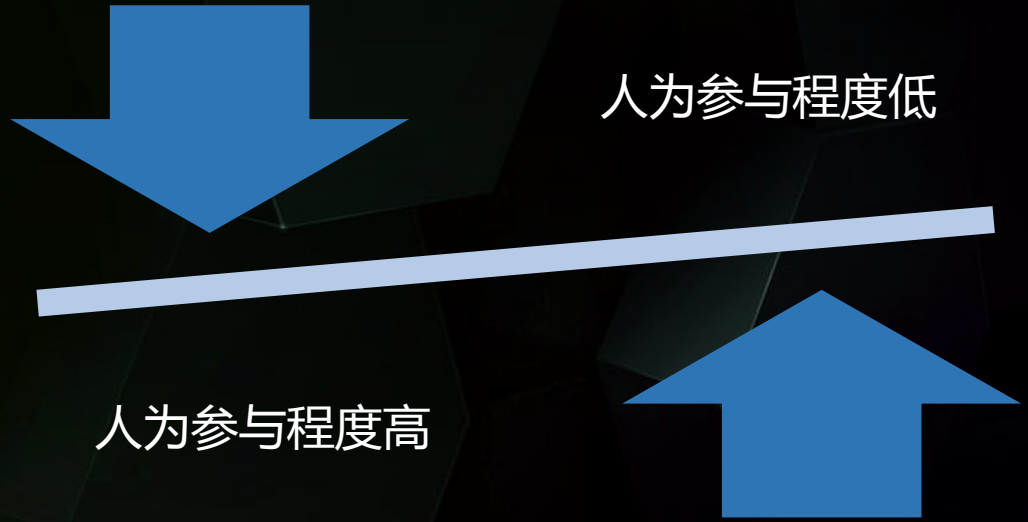
高级益智玩具类型

- 带有博弈心态的扑克
- 带有套路计算的麻将
- 带有长久盘面考虑功能的象棋或围棋等



人工智能的差距

人工智能的差距



高级人工智能的套路

落地：

- 经典统计
- 神经网络
- 强化学习

高级人工智能的套路

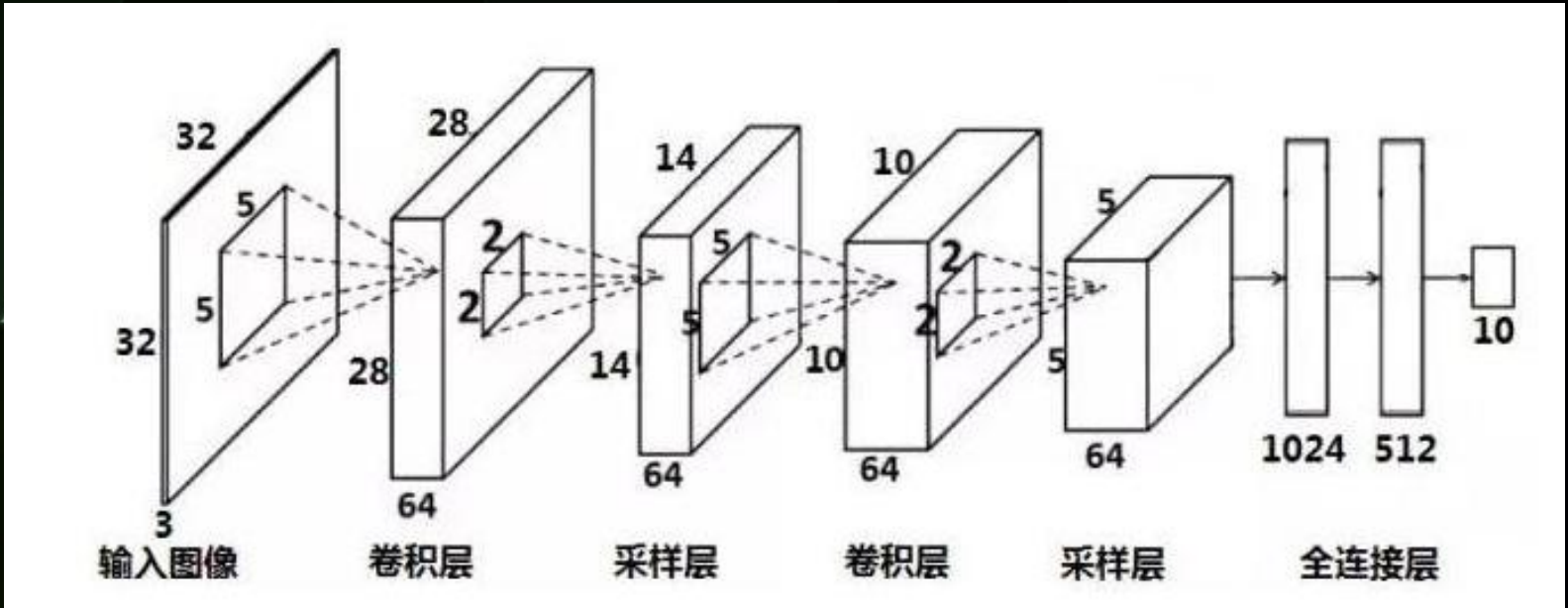
经典统计：

- 统计概率模型
- 排列组合模型
- 隐马尔可夫模型

数学期望

高级人工智能的套路

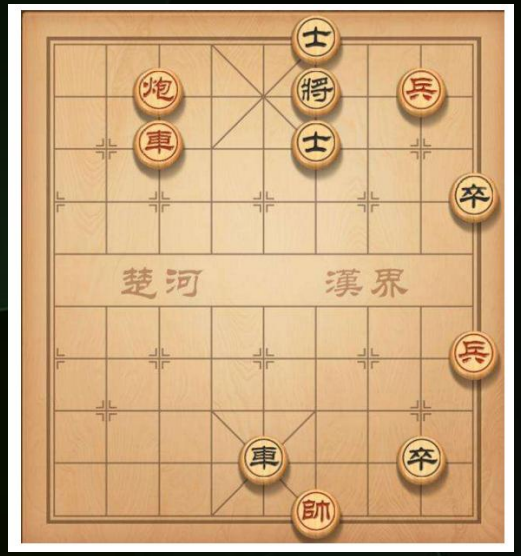
卷积网络：特征提取

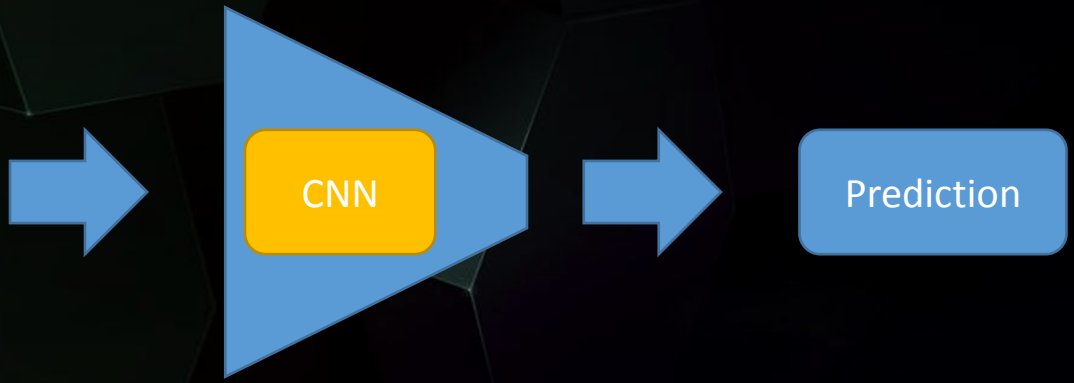


高级人工智能的套路

APICloud | IT大咖说 知识共享平台

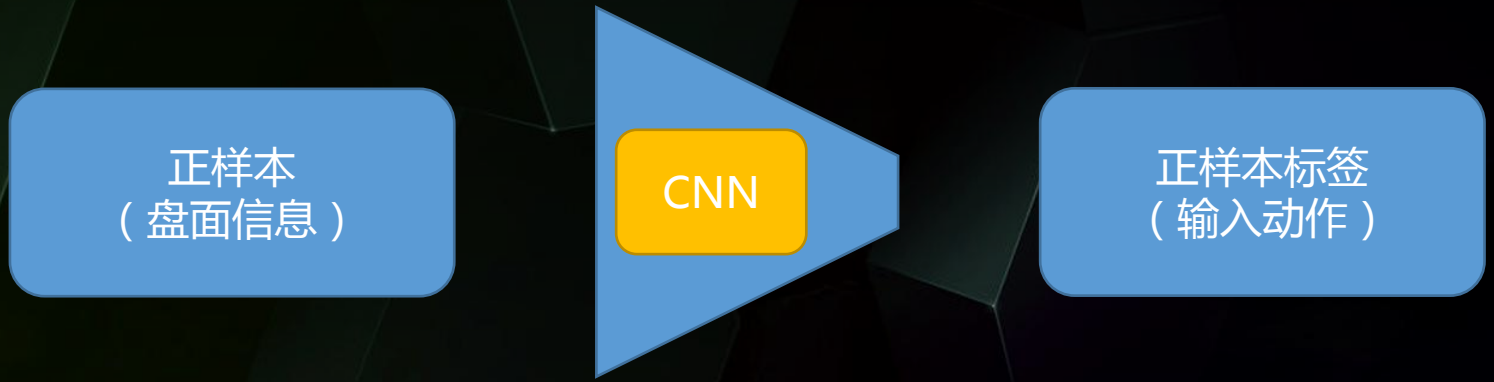
卷积网络





高级人工智能的套路

卷积网络



卷积网络的优点 / 缺点

收敛速度快

泛化能力好

应用场景广

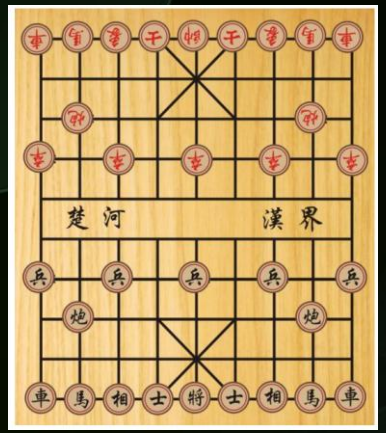
缺点：需要大量正样本及人类干预

高级人工智能的套路

卷积网络适用游戏类型：

适用于变化相对有限的游戏

适用于输入数据量偏小的游戏



卷积网络实现AI

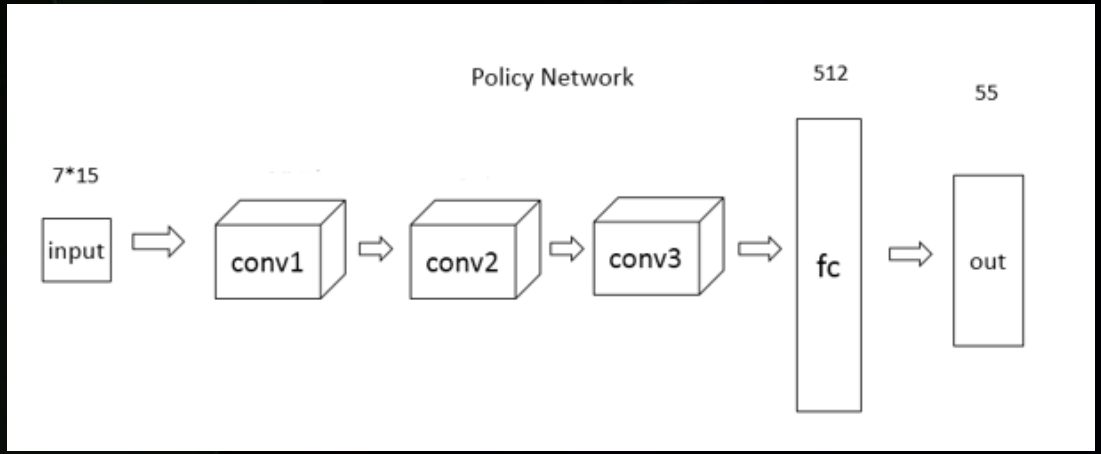
CNN方法

扫描各轮出牌和手牌，预测当前情况下本人出什么牌胜率最大

卷积网络实现AI

多层卷积网络

$$\begin{bmatrix} 0,2,0,0,0,0,0,0,0,0,0,2,4,1,1 \\ 0,0,0,0,0,2,0,0,0,0,0,0,0,0,0 \\ 0,0,0,2,0,0,0,0,0,0,0,0,0,0,0 \\ 0,0,0,0,0,0,0,0,0,2,0,0,0,0,0 \\ 0,0,0,2,1,1,1,1,1,0,3,0,0,0,0 \\ 0,0,0,2,0,0,0,0,0,0,0,0,0,0,0 \\ 0,0,0,0,0,0,0,0,0,2,0,0,0,0,0 \end{bmatrix}$$



卷积网络实现AI

向量说明

第一行：代表玩家现在的手牌状态；

第二行：代表玩家上轮出牌记录；

第三行：代表上家上轮出牌记录；

第四行：代表下家上轮出牌记录；

第五行：代表玩家的所有出牌记录；

第六行：代表上家的所有出牌记录；

第七行：代表下家的所有出牌记录；

卷积网络实现AI

损失函数

输出向量

- (0,0,0,0)：不出牌
- (1,0,0,0)：出一张
- (0,1,0,0)：出二张
- (0,0,1,0)：出三张
- (0,0,0,1)：出四张

出牌方式 y

真实出牌方式y

$$\text{Loss}= 1/n \sum i=1 \uparrow n \sum j=1 \uparrow 55 [y \downarrow i,j \log(y \downarrow i,j) +(1-y \downarrow i,j)\log(1-y \downarrow i,j)]$$

4*13=52

0,0,0

小王、大王、不出牌

卷积网络实现AI

胜率：

- 和人对抗
- 地主身份：50%左右的胜率
- 农民身份：40%

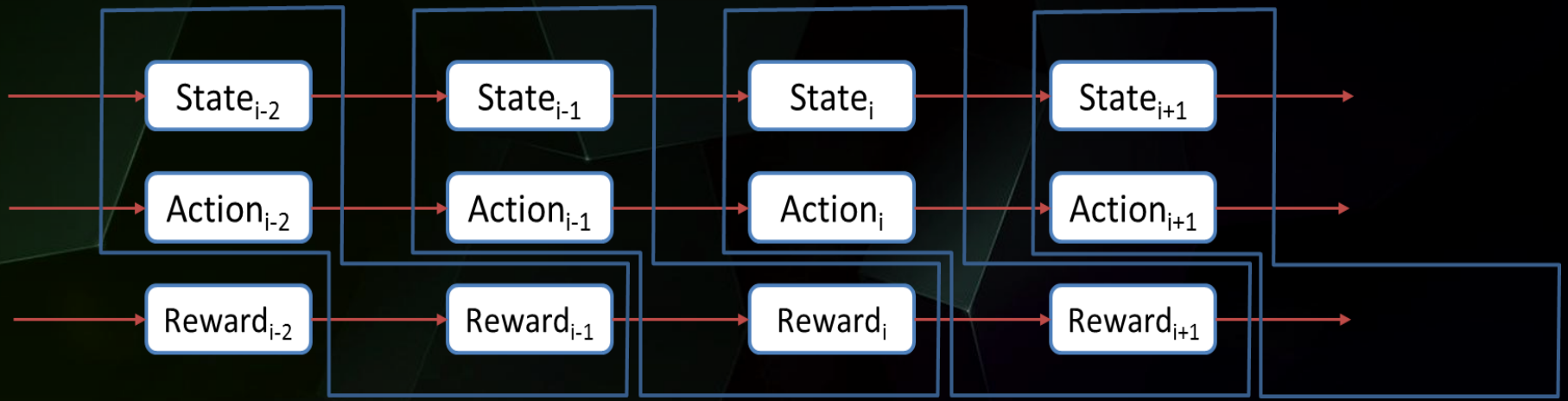
DQN实现AI原理

DQN（Deep Q-Network）

强化学习

DQN实现AI原理

马尔可夫决策过程



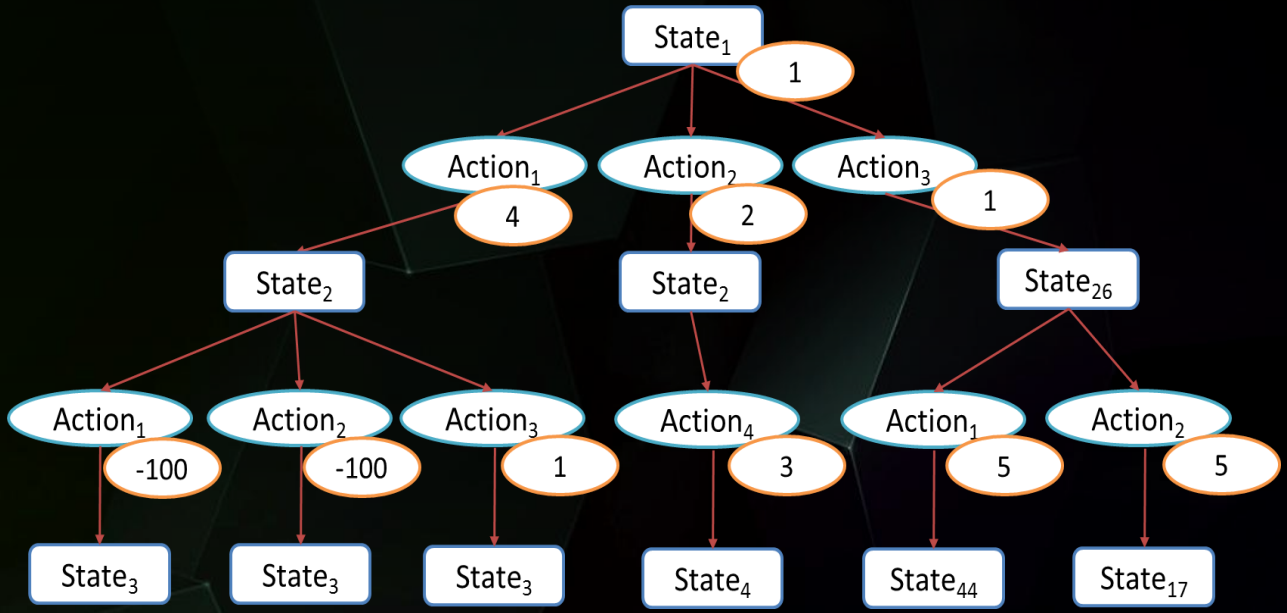
DQN实现AI原理

马尔可夫决策过程

	ACTION1	ACTION2	ACTION3	ACTION4	ACTION5	ACTION6	……
STATE1							
STATE2							
STATE3							
STATE4							
……							

DQN实现AI原理

动态决策（Dynamic Programming）



$$Q(s_t,a_t) \neg\ Q(s_t,a_t) + a'\ [r_{t+1} + gMAXQ(s_{t+1},a_{t+1}) - Q(s_t,a_t)]$$

DQN实现AI原理

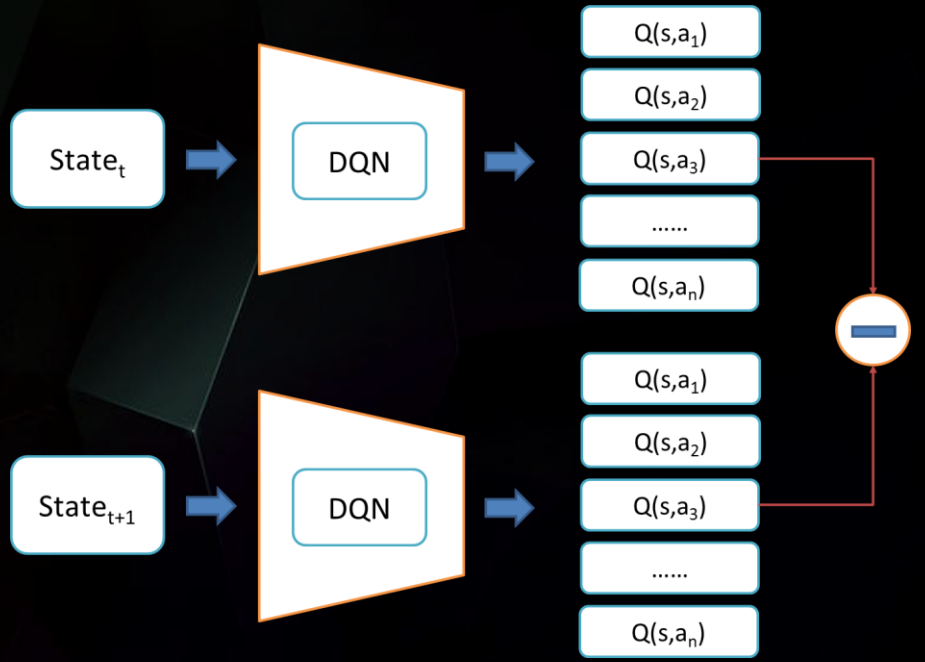
DQN损失函数比较特殊

根据贝尔曼方程

$$Q_t^*(S_t, A_t) = E[r_{t+1} + \max_{a_{t+1}} Q_{t+1}^*(S_{t+1}, A_{t+1}, a_{t+1}) \mid S_t, A_t]$$

$$E[r_t + \max_{a_{t+1}} Q_{t+1}^*(S_{t+1}, A_{t+1}, a_{t+1}) - Q_t(S_t, A_t; \theta)]^2$$

$$Loss(w) = E[r + \gamma \max Q(s', a', w) - Q(s, a, w)]^2$$



DQN实现AI原理

- 目前从训练结果来看DQN的胜率远超CNN
- DQN的落地也就意味着未来用它去做其它人工智能连续决策方面的事情有了保障
- 下棋、麻将、自动驾驶等

强化学习延展

DQN

开山之作

Double DQN

反馈稳定

Dueling DQN

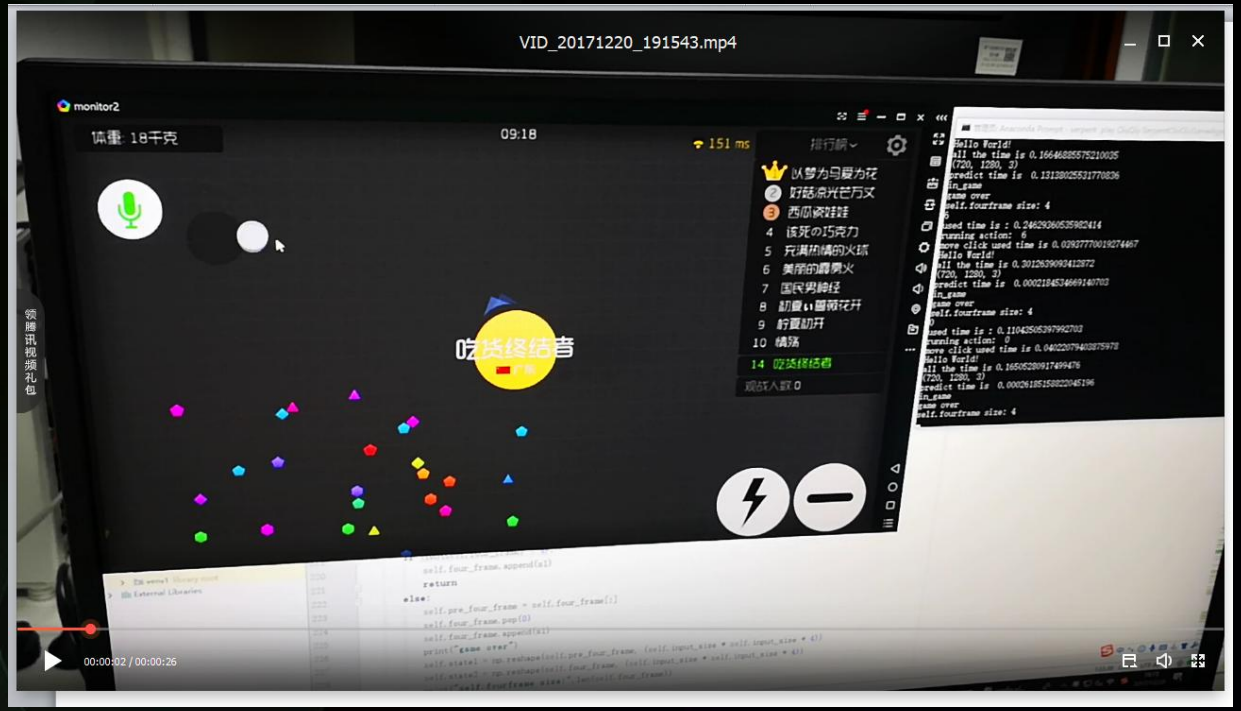
评价合理

DDPG

连续输出

A3C

并行训练



谢谢观看
THANKS