不完全信息下强化学习的研究与应用

Research and Application of Reinforcement Learning under Incomplete Information

信息与数据科学　张万鹏　　指导教师：阮吉寿

2019/5/24

一、背景

强化学习是机器学习方法中的一类算法，在给定环境下模拟各种行为和动作，接收环境传递的激励和惩罚反馈，自行学习如何行动才能使长期收益最大化。

在传统的强化学习问题中，环境信息会明确给出，如围棋这种双人博弈游戏，由于棋盘盘面有限，且规则简单清晰，对手的当前状态和下一步状态之间的转移概率分布可以得到准确的表示。通过得知这些信息，能够清晰建立准确的环境模型，进而做到通过具体的数学分析来求得最优解。

具体地，一场博弈中，如果玩家完美掌握了对手的策略、特征、回报函数等信息，称玩家掌握「**完全信息**」，并称这样的博弈为「**完全信息博弈**」，反之称为「**不完全信息博弈**」。

在不完全信息下，由于难以重建环境模型，首先需要建立对手的资源概率分布，才能进一步建立对手的策略概率分布，导致问题的解空间非常复杂，传统的强化学习方法难以克服这一问题，因此需要做出一些改进。

二、核心思想

强化学习的核心观点是要使期望收益值最大化，所以其基本算法思路就是通过定义「**价值函数**」，然后尝试去优化总期望收益价值。为了实现这一目标，最简单的方法是把问题抽象为「**Markov 决策过程**」，基于「**状态转移概率分布**」进行理论分析，在其基础上提出了 Bellman 方程关系：

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s') \right]$$

通过解该方程，理论上可以得到一个最优解，但该算法效率极低，且要求能够定义出状态转移概率分布 $p(s', r|s, a)$ ，因此问题背景必须是「**完全信息博弈**」，导致这样的算法实用价值不高。后面则将基于该核心思想进行改进和优化。

三、改进

改进一：Monte Carlo 模拟

为了能够使强化学习适用于「**不完全信息博弈**」，首先需要克服无法确定状态转移概率分布 $p(s', r|s, a)$ ，即无法计算 Bellman 方程中 $\sum_{s',r} p(s', r|s, a)[r + \gamma v(s')]$ 的问题。

考虑引入 Monte Carlo 模拟方法，可以构造出环境模拟器来模拟采样，从而得到近似回报值

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$$

来替代 Bellman 方程中的 $\sum_{s',r} p(s', r|s, a)[r + \gamma v(s')]$ ，避免使用状态转移概率分布来直接计算期望的复杂做法。通过引入 Monte Carlo 模拟这一改进，确保了强化学习方法也能处理不完全信息博弈问题。

改进二：Bootstrap 估计

在上面的 Monte Carlo 算法中，由于需要等待一个完整的回合结束，得到每个时间点的奖励值 R_t，才能返回到之前各时间点，求得用于估计总收益的回报值

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$$

这一缺点限制了算法的使用场景，其高延迟使得算法无法实时地进行「**在线学习**」。为了解决这一问题，基于统计学中 Bootstrap 自助法的重抽样思想，使用估计值

$$\widehat{Q}(S_{t+1}, A_{t+1}) \approx R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \cdots$$

来取代「**当前时间点**」以后的所有奖励值

这样便能有

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \\ &= R_{t+1} + \gamma \left[R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \cdots \right] \\ &\approx R_{t+1} + \gamma \widehat{Q}(S_{t+1}, A_{t+1}) \end{aligned}$$

从而能够根据当前时刻 t 下的信息进行一定程度的 “再利用”，使用估计值来替换 G_t，提高采样效率，确保能够实时地在线估计收益价值：

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \widehat{Q}(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right]$$

改进三：ε-贪心学习

在强化学习的策略选择中，基本思路是要去采取能使期望收益最大化的策略行为，这样的策略属于「**贪心策略**」，即

$$\pi(a|s) = \begin{cases} 0 & , a \neq \max\limits_{a} q_\pi(s,a) \\ 1 & , a = \max\limits_{a} q_\pi(s,a) \end{cases}$$

过于贪心会导致算法非常容易收敛到局部最优解，为了解决这一问题，设定一个较小的探索率 ε，让算法以 $1-\varepsilon$ 的概率进行常规学习，ε 的概率进行自由探索，从而较好地避免局部收敛的情况。

$$\pi(a|s) = \begin{cases} \frac{\varepsilon}{|\mathcal{A}(s)|} & , a \neq \max\limits_{a} q_\pi(s,a) \\ 1-\varepsilon+\frac{\varepsilon}{|\mathcal{A}(s)|} & , a = \max\limits_{a} q_\pi(s,a) \end{cases}$$

四、优化

优化一：结合梯度下降法

在前面的算法中，$s \in \mathcal{S}, a \in \mathcal{A}$ 所在的变量空间都很大，对变量进行逐个更新的效率很低，使得算法实用性并不高，为了优化算法的效率，考虑结合机器学习中的梯度下降法，将算法进行参数化改造，首先得到参数化的 loss function：

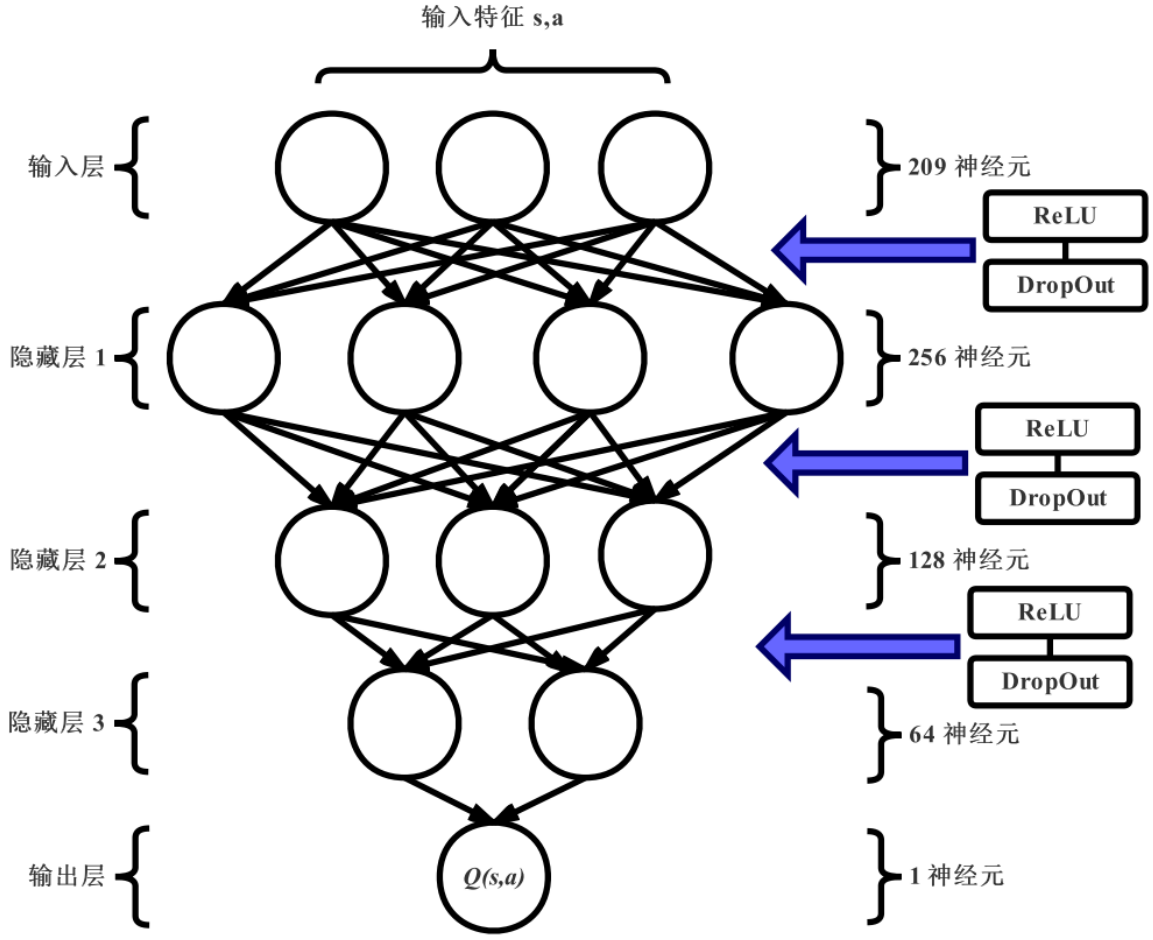
$$J(\boldsymbol{w}) = \frac{1}{2}\left[R_{t+1} + \gamma \widehat{Q}(S_{t+1}, a, \boldsymbol{w}) - \widehat{Q}(S_t, A_t, \boldsymbol{w})\right]^2$$

进而可以使用梯度下降法来进行高效迭代收敛：

$$\begin{aligned}
\boldsymbol{w}_{t+1} =& \boldsymbol{w}_t - \alpha \nabla J(\boldsymbol{w}) \\
=& \boldsymbol{w}_t + \alpha \left[R_{t+1} - \gamma \widehat{Q}\left(S_{t+1}, a, \boldsymbol{w}_t\right) - \widehat{Q}\left(S_t, A_t, \boldsymbol{w}_t\right)\right] \\
& \times \left[\gamma \nabla \widehat{Q}\left(S_{t+1}, a, \boldsymbol{w}_t\right) - \nabla \widehat{Q}\left(S_t, A_t, \boldsymbol{w}_t\right)\right]
\end{aligned}$$

优化二：使用 Neural Network

在梯度下降法的基础上，通过反向梯度传播的方法，引入深度神经网络，进一步提升算法的收敛效率。



优化三：双神经网络抵消偏差

上面提出的各种优化和改进中，存在大量的估计和近似，这会给算法带来一定的偏差，但是巧妙的是，两个存在这种偏差的估值网络交替使用，恰好能够抵消这一偏差。

可以证明，构造两个神经网络 Q_1, Q_2 来进行价值估计：

$$Q_2(S, \arg\max_a Q_1(a)) = Q_2(S, A^*)$$

这样得到的估计值

$$\mathbb{E}\left[Q_2(S, \arg\max_a Q_1(a))\right] = \mathbb{E}\left[Q_2(s, A^*)\right] = q(s, A^*)$$

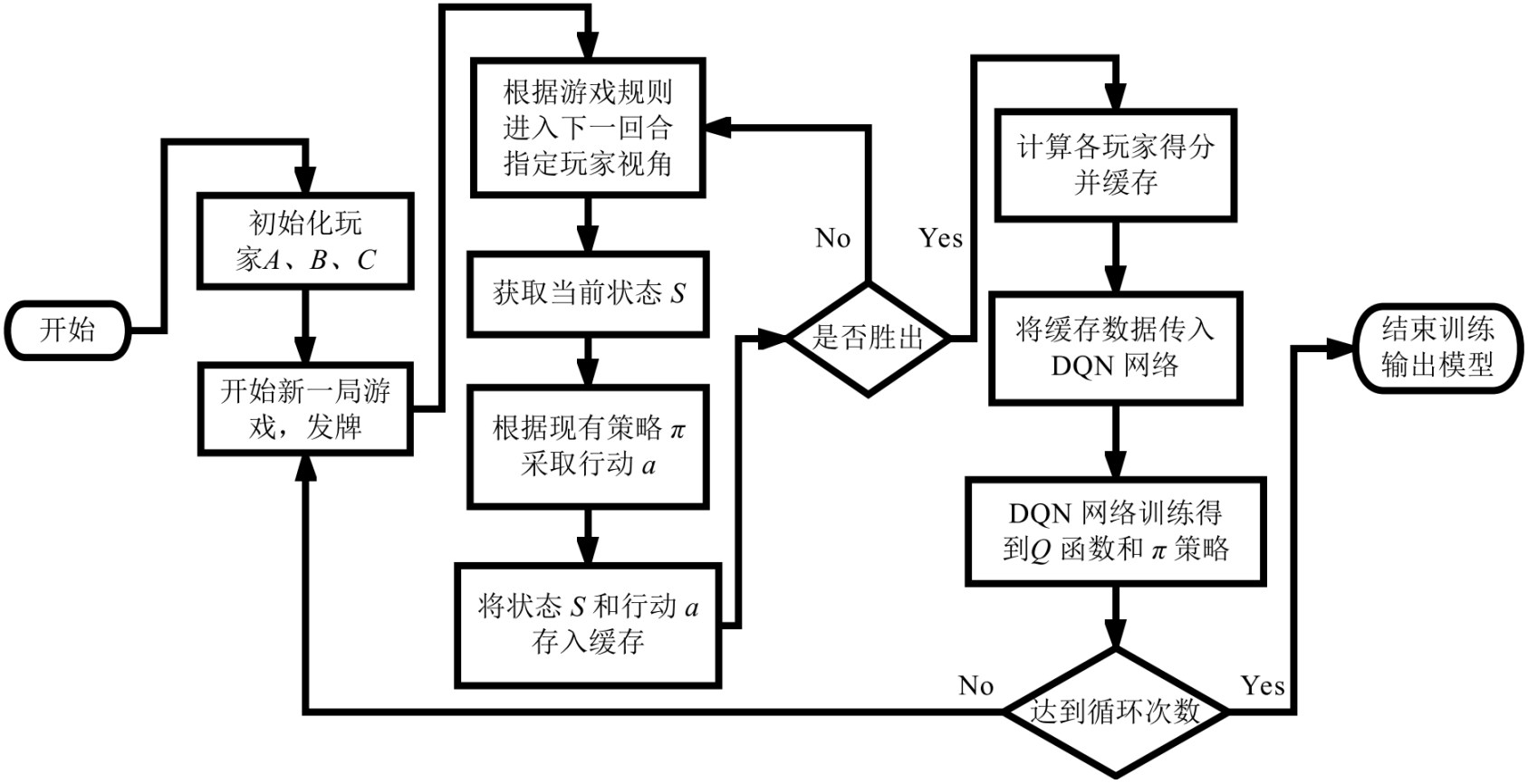
是真实值的无偏估计。

基于这一定理，可以通过构建两个神经网络交替使用来抵消偏差值，使评估指标为真实值的无偏估计，从而可以确保在该评估体系下能够训练和学习出恰当的博弈策略。

五、仿真实验

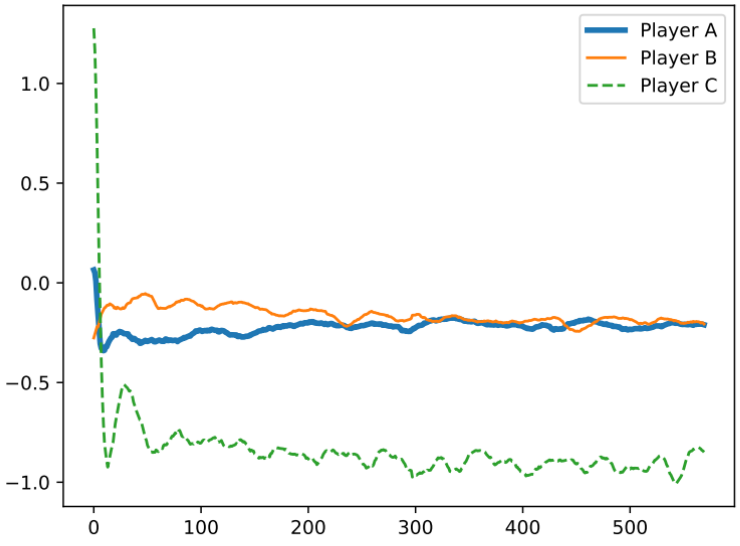
为了验证所提出的算法能否较为适应地处理现实中的不完全信息博弈问题，本文基于一个纸牌游戏进行了仿真测试实验。

（1）仿真实验流程

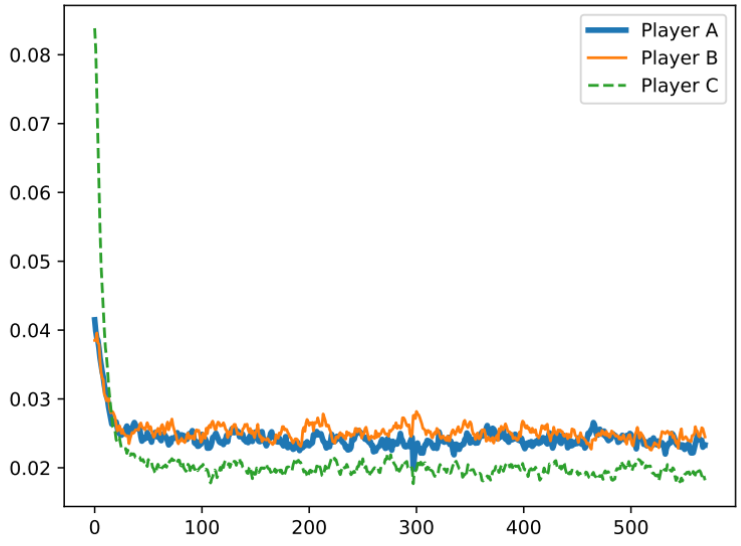

开始
初始化玩家A、B、C
开始新一局游戏，发牌
根据游戏规则进入下一回合指定玩家视角
获取当前状态 S
根据现有策略 π 采取行动 a
将状态 S 和行动 a 存入缓存
是否胜出
No
Yes
计算各玩家得分并缓存
将缓存数据传入DQN 网络
DQN 网络训练得到Q 函数和 π 策略
达到循环次数
No
Yes
结束训练输出模型

（2）仿真实验结果

收敛性

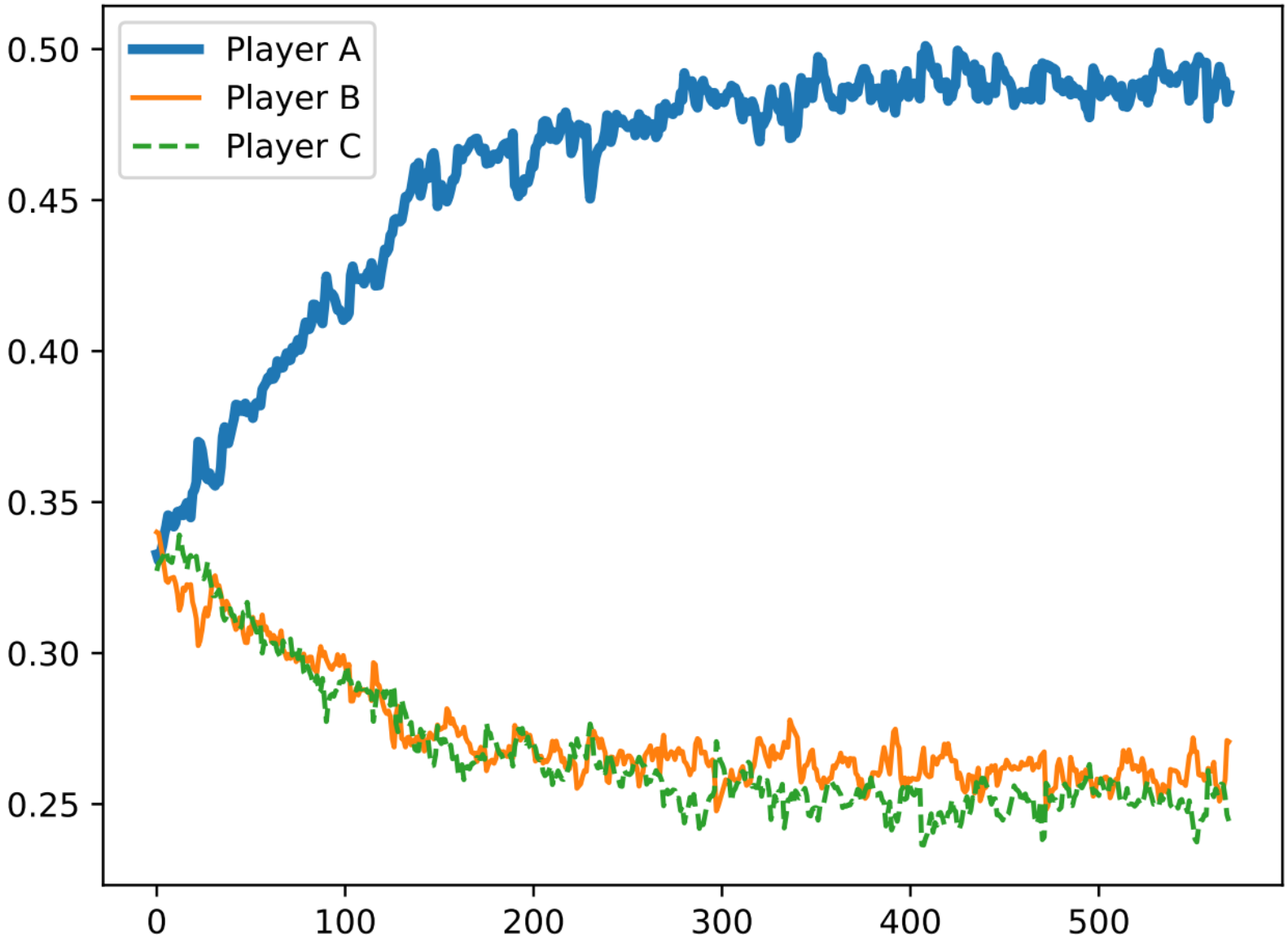


(a) Q 函数训练均值



(b) Q-Learning 训练误差值

实时训练胜率


Player A
Player B
Player C
0.50
0.45
0.40
0.35
0.30
0.25
0
100
200
300
400
500

（3）独立测试结果

主模型 A		辅助模型 B		辅助模型 C	
$1-\varepsilon_A$	胜率	$1-\varepsilon_B$	胜率	$1-\varepsilon_C$	胜率
0.99	0.662	0.99	0.165	0.99	0.173
0.99	0.685	0.90	0.158	0.90	0.157
0.95	0.621	0.80	0.191	0.80	0.188
0.95	0.646	0.70	0.173	0.70	0.181
0.90	0.558	0.60	0.221	0.60	0.221
0.90	0.583	0.50	0.207	0.50	0.210

六、总结与展望

论文提出了能够解决多人博弈问题的“自适应 Deep Q-Learning 算法”。该方法将传统的理论强化学习方法与新兴的深度学习技术相结合，其创新点在于使用了双神经网络来抵消偏差，以及分离神经网络进行交互博弈，来进行自适应学习。

目前，对于简单的完全信息博弈问题，已经提出了许多有着不错成果的强化学习理论，但对于更实际且更复杂的不完全信息博弈问题，还尚未得到充分的研究，因此，针对不完全信息博弈问题的强化学习将会是未来的一个重要研究方向，有待进一步地充分研究。

谢谢

感谢各位老师在百忙之中抽出时间参与论文审阅与答辩，谢谢!