MADDPG

Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments

Ryan Lowe*
McGill University
OpenAI

Yi Wu*
UC Berkeley

Aviv Tamar
UC Berkeley

Jean Harb
McGill University
OpenAI

Pieter Abbeel
UC Berkeley
OpenAI

Igor Mordatch
OpenAI

(Submitted on 7 Jun 2017 (v1), last revised 16 Jan 2018 (this version, v3))

7107018017 林祐陞

Schema

1 Introduction

- RL很常涉及Multi-agent的交互情況
- 傳統的DQN, policy gradient都不適用於MAS
- 主要問題 :
  - training過程中, 每個agent都在改變policy, 造成non-stationary
  - 對DQN來說, experience replay不可用
  - 對policy gradient來說, 環境不斷改變, 造成學習的variance進一步增大
- 新方法 : MADDPG

1 MADDPG

Policy-gradient
Q-learning

Actor-critic

+DQN

DDPG

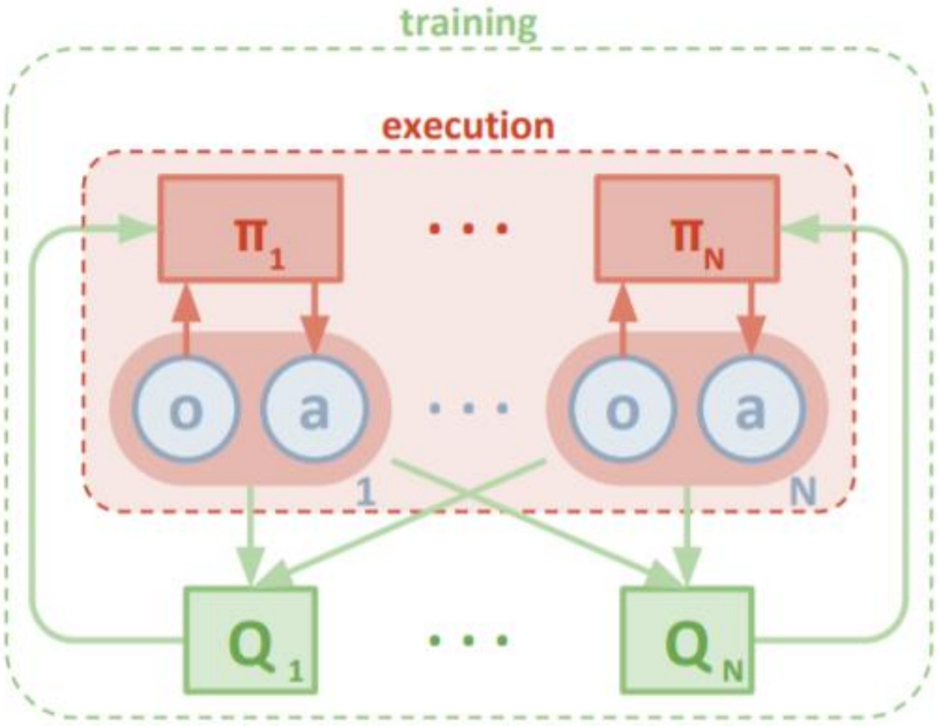
MADDPG

2 DDPG

- DDPG是Actor-critic和DQN算法的結合
- Deep : memory pool + 雙網路結構
- Deterministic : 使Actor不再輸出每個action的概率, 而是明確的一個action

3 MADDPG

- 每個Agent的訓練和DDPG類似
- 不同點在於Critic的input, 增加了額外信息
    - 例如 : 其他Agent的action
- 集中學習 + 分散執行
- 目標： 一個通用型的學習算法
    - 1. 執行中只使用local information
    - 2. 不需知道環境的可微分模型
    - 3. 不做通訊方法結構上的假設



3 MADDPG

- 若我們知道所有agent採取的action, 即便policy改變, 環境仍然穩定 s → s ’
- 放寬critic input的假設 , 用近似方式去計算其他agent的policy
- 使用這種方式集中學習

3

Multi-Agent Deep Deterministic Policy Gradient Algorithm

For completeness, we provide the MADDPG algorithm below.

Algorithm 1: Multi-Agent Deep Deterministic Policy Gradient for N agents

for episode = 1 to M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial state $\mathbf{x}$
  for $t = 1$ to max-episode-length **do**
    for each agent i, select action $a_i = \boldsymbol{\mu}_{\theta_i}(o_i) + \mathcal{N}_t$ w.r.t. the current policy and exploration
    Execute actions $a = (a_1, \ldots, a_N)$ and observe reward r and new state $\mathbf{x}'$
    Store $(\mathbf{x}, a, r, \mathbf{x}')$ in replay buffer $\mathcal{D}$
    $\mathbf{x} \leftarrow \mathbf{x}'$
    for agent $i = 1$ to N **do**
      Sample a random minibatch of S samples $(\mathbf{x}^j, a^j, r^j, \mathbf{x}'^j)$ from $\mathcal{D}$
      Set $y^j = r_i^j + \gamma Q_i^{\boldsymbol{\mu}'}(\mathbf{x}'^j, a'_1, \ldots, a'_N)|_{a'_k = \boldsymbol{\mu}'_k(o_k^j)}$
      Update critic by minimizing the loss $\mathcal{L}(\theta_i) = \frac{1}{S}\sum_j \left(y^j - Q_i^{\boldsymbol{\mu}}(\mathbf{x}^j, a_1^j, \ldots, a_N^j)\right)^2$

對應Q-network的更新

      Update actor using the sampled policy gradient:

$$\nabla_{\theta_i} J \approx \frac{1}{S}\sum_j \nabla_{\theta_i}\boldsymbol{\mu}_i(o_i^j)\nabla_{a_i} Q_i^{\boldsymbol{\mu}}(\mathbf{x}^j, a_1^j, \ldots, a_i, \ldots, a_N^j)\Big|_{a_i = \boldsymbol{\mu}_i(o_i^j)}$$

對應每個actor的更新

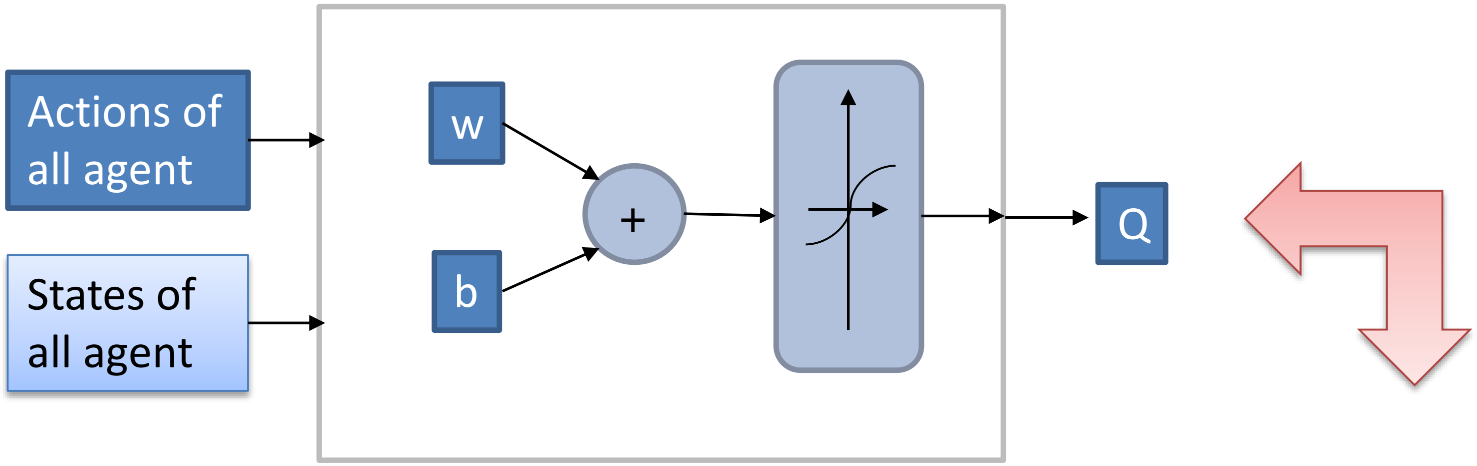
    end for
    Update target network parameters for each agent i:

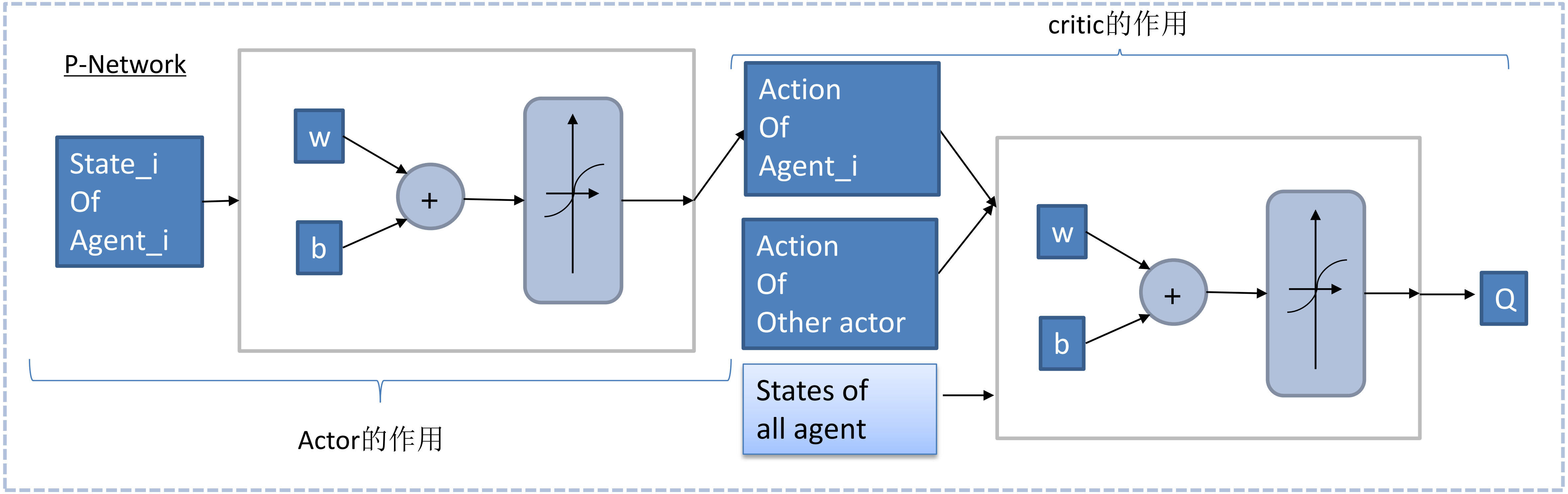
$$\theta'_i \leftarrow \tau\theta_i + (1 - \tau)\theta'_i$$

  end for
end for

3

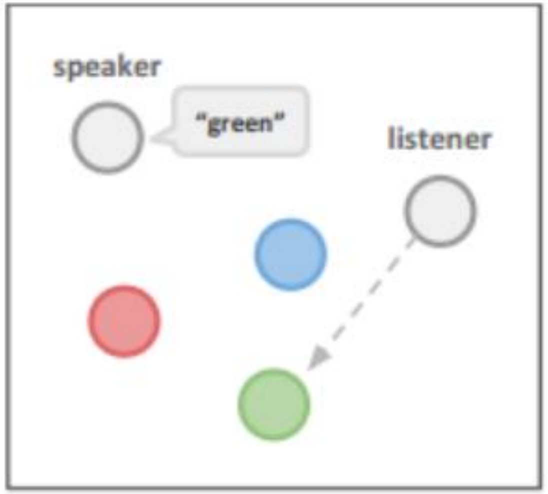
Q-Network


Actions of all agent
States of all agent
w
b
+
Q


P-Network
critic的作用
State_i Of Agent_i
w
b
+
Action Of Agent_i
Action Of Other actor
States of all agent
w
b
+
Q
Actor的作用

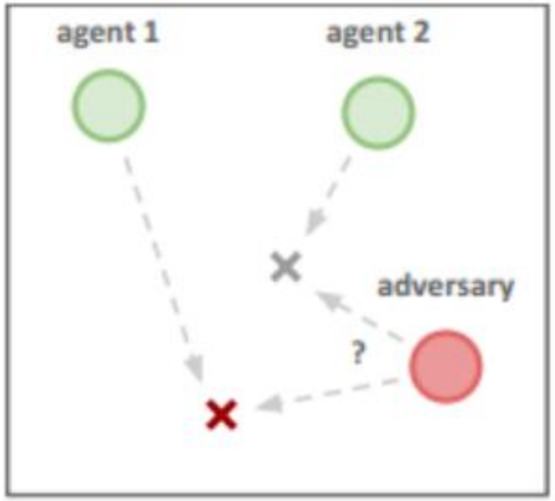
4 Experiments

➢ 合作通訊

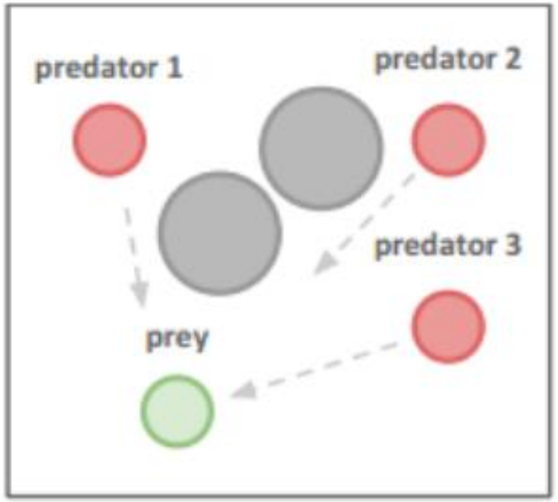


➢ 合作導航

➢ 欺騙



➢ 捕食



5 Summary

- 實驗結果表明MADDPG比傳統RL方式更能適應複雜的環境設置
- 缺點 : critic的input space會隨著agent數量線性成長
- 解決方案 : 實戰中, 僅使用鄰居的action來緩解
- 應用在股票市場

6 Reference

- Deep reinforcement learning For Multi-Agent Systems : A Review of Challenges, Solutions and Applications (Review)

- Continuous Control with Deep Reinforcement Learning (DDPG)

- Counterfactual Multi-Agent Policy Gradients (COMA)

- Cooperative Multi-Agent Control Using Deep Reinforcement Learning (Gupta 3 method)