Imperfect-Information Problems

Team 19

Wen-Yu Lee, Yifu Wang, Qi Yu

Problem Description

...

Problem Description - Why do we need to explore the solution of imperfect-information game

- Perfect Information (pacman, go.....)
  - Nash equilibrium
- Reality: Imperfect Information (financial and energy trading, traffic control)
  - Large scope, expensive
  - Poker

Approaches

...

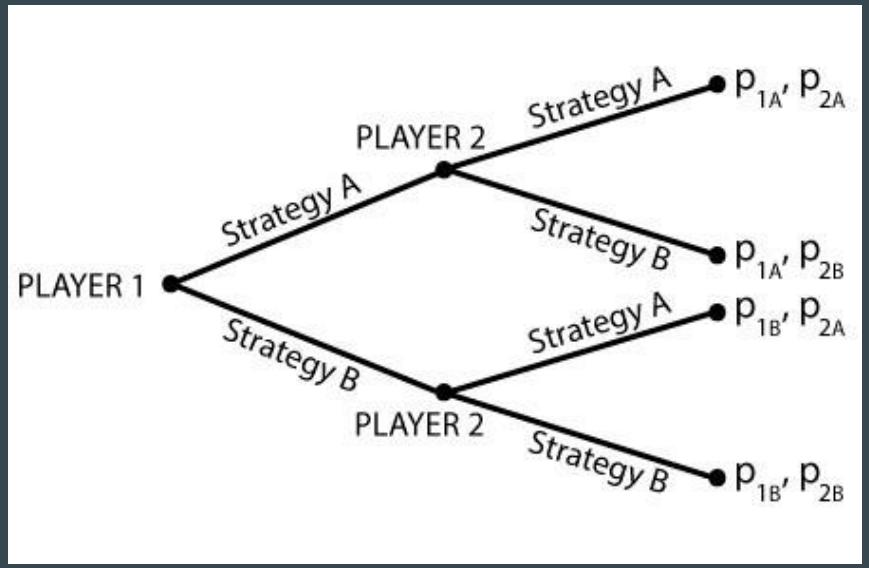
“Deep Reinforcement Learning from Self-Play in Imperfect-Information Games”
by J. Heinrich and D. Silver

Nash Equilibrium

- No players has incentive to change his/her strategy given what the other players are doing.

Fictitious Self-Play (FSP)

- Fictitious Play (FP): Choose the best response to their opponents' average strategies.
- Fictitious Self-Play (FSP): Mix between their best response and average strategies.



Neural Fictitious Self-Play (NFSP)

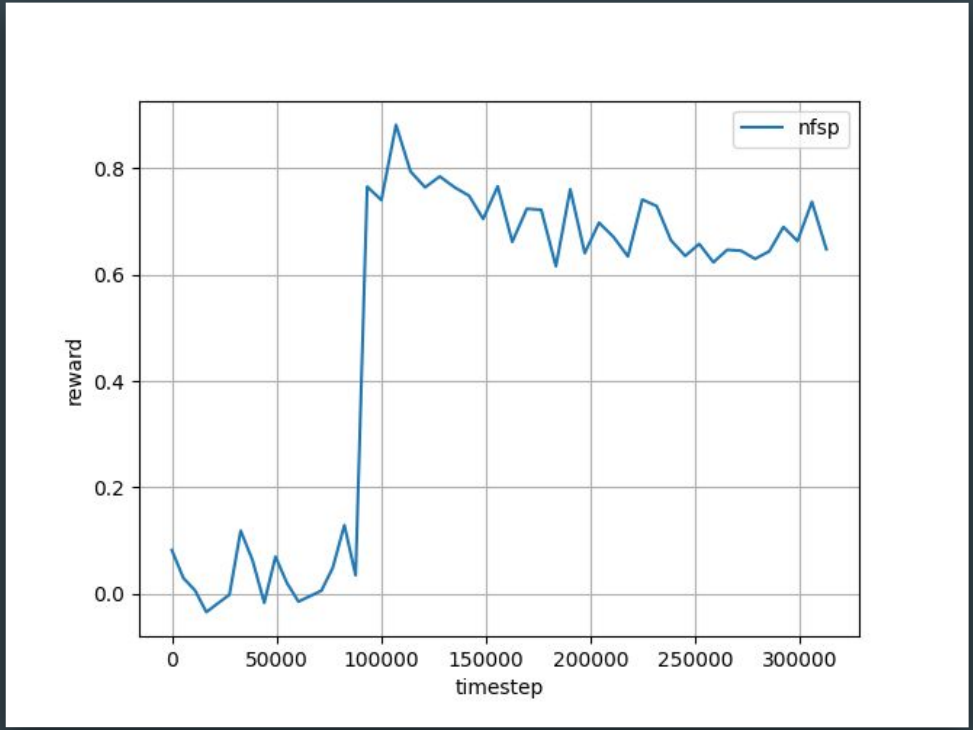
- Combination of Neural Network and Fictitious Self-Play

Algorithm 1 Neural Fictitious Self-Play (NFSP) with fitted Q-learning

Initialize game Γ and execute an agent via RUNAGENT for each player in the game
function RUNAGENT(Γ)
  Initialize replay memories $\mathcal{M}_{RL}$ (circular buffer) and $\mathcal{M}_{SL}$ (reservoir)
  Initialize average-policy network $\Pi(s, a \mid \theta^{\Pi})$ with random parameters θ^{Π}
  Initialize action-value network $Q(s, a \mid \theta^{Q})$ with random parameters θ^{Q}
  Initialize target network parameters $\theta^{Q'} \leftarrow \theta^{Q}$
  Initialize anticipatory parameter η
  for each episode **do**
    Set policy $\sigma \leftarrow \begin{cases} \epsilon\text{-greedy}\,(Q), & \text{with probability } \eta \\ \Pi, & \text{with probability } 1-\eta \end{cases}$
    Observe initial information state s_1 and reward r_1
    for $t = 1, T$ **do**
      Sample action a_t from policy σ
      Execute action a_t in game and observe reward r_{t+1} and next information state s_{t+1}
      Store transition $(s_t, a_t, r_{t+1}, s_{t+1})$ in reinforcement learning memory $\mathcal{M}_{RL}$
      if agent follows best response policy $\sigma = \epsilon$-greedy (Q) **then**
        Store behaviour tuple (s_t, a_t) in supervised learning memory $\mathcal{M}_{SL}$
      end if
      Update θ^{Π} with stochastic gradient descent on loss
        $\mathcal{L}(\theta^{\Pi}) = \mathbb{E}_{(s,a)\sim\mathcal{M}_{SL}}\left[-\log \Pi(s, a \mid \theta^{\Pi})\right]$
      Update θ^{Q} with stochastic gradient descent on loss
        $$\mathcal{L}\left(\theta^{Q}\right) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{M}_{RL}}\left[\left(r + \max_{a'} Q(s', a' \mid \theta^{Q'}) - Q(s, a \mid \theta^{Q})\right)^2\right]$$
      Periodically update target network parameters $\theta^{Q'} \leftarrow \theta^{Q}$
    end for
  end for
end function

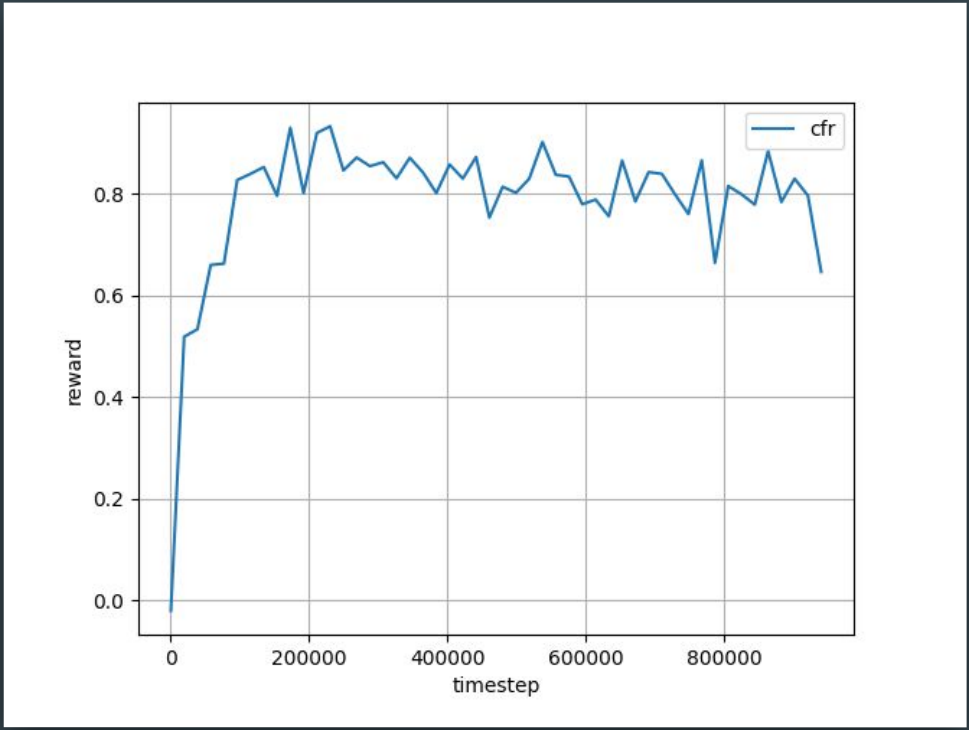
Results

...



NFSP Result

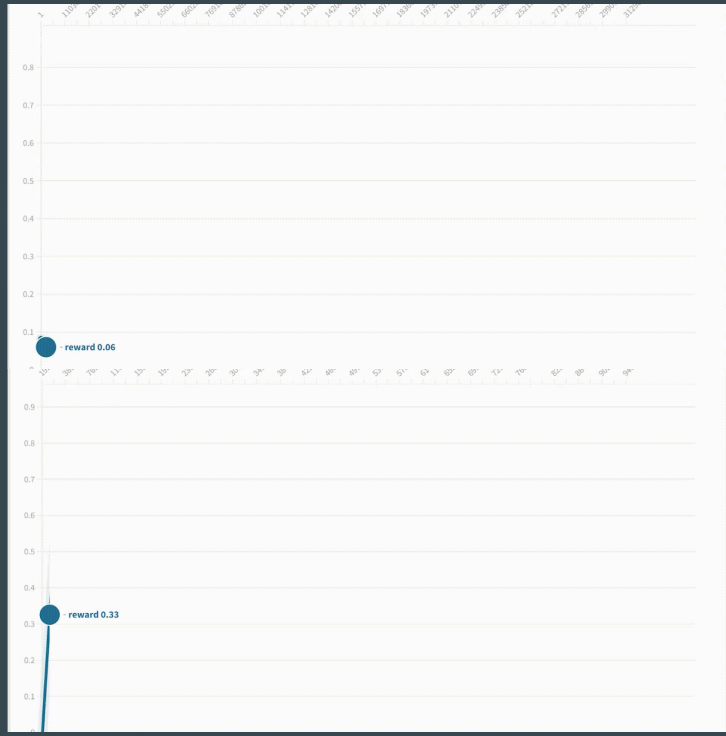
- The NFSP agents learn by playing against themselves, without any explicitly dened prior knowledge. It uses DQN to replay experiences with Q-Learning.



CFR Result

- CFR is a RL algorithm that analyses the effect of each move played by the agent and attempts to minimize the loss.

NFSP VS CRF



Lesson Learned

...

Lesson Learned

- Nash Equilibrium
- Normal form and Extensive form
- Fictitious Self-Play
- Neural Fictitious Self-Play

Future Work

...

Large-Scale Imperfect-Info Games

- Monte Carlo Neural Fictitious Self-Play (MC-NFSP)
- Asynchronous Neural Fictitious Self-Play (ANFSP)
- Other Models .....

Q&A

...