Fakultas Informatika
School of Computing
Telkom University

CII3N3-Sistem Multi Agen

Semester Genap 2023/2024

Tjokorda Agung Budi Wirayuda (COK)

Pertemuan 13: Multi Q-Learning Variant

Cek Progress: Tugas Individu

- Kerjakan tugas individu berikut ini:
  - https://huggingface.co/learn/deep-rl-course/unit7/hands-on

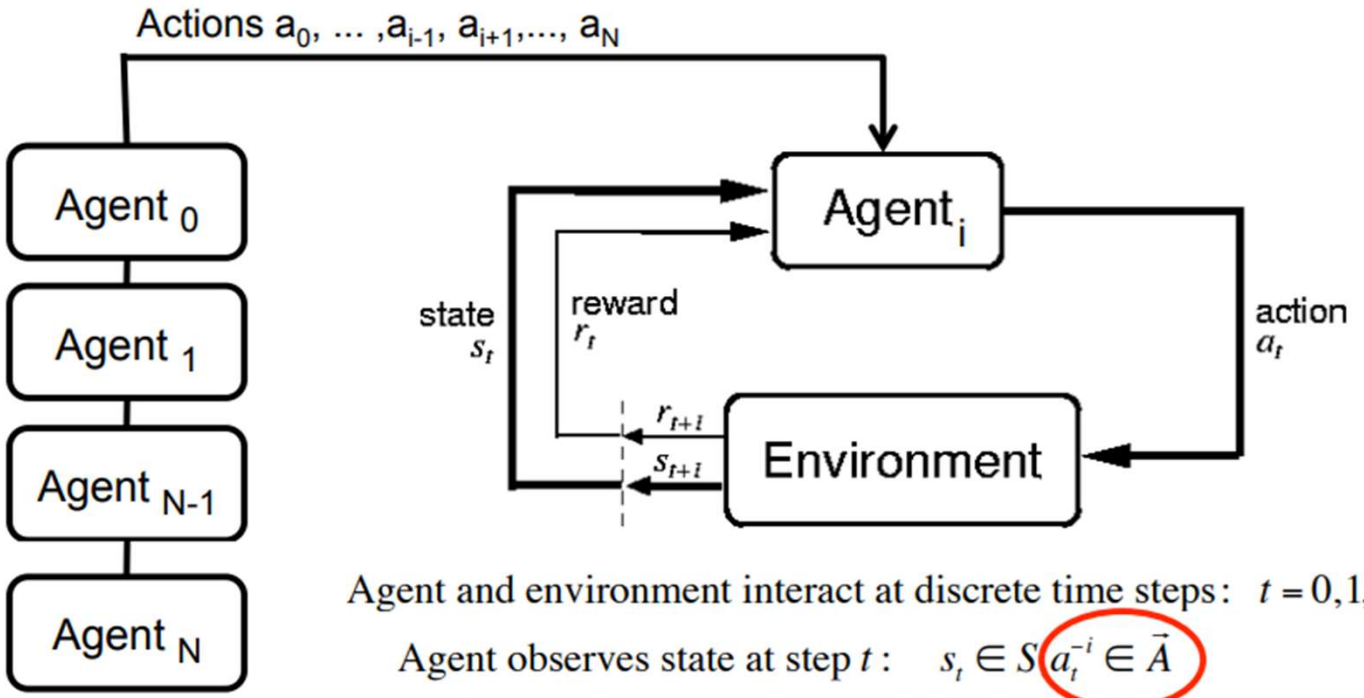
Joint-Action Learning

Cooperation by learning joint-action values

Joint-Action Learning

- Consider the case that we have 2 offenders in the soccer game instead of one:
  - The optimal policy depends on the joint action
  - For example, if robot A approaches the ball, the optimal action of robot B would be to do something else, e.g. going to the support position
- Solution: each agent learns a Q-Function of the joint action space: Q(s,)
- Observation or communication of actions performed by the team mates is required

The Agent-Environment Interface for Joint-Action learners



Agent and environment interact at discrete time steps: $t = 0, 1, 2, \ldots$

Agent observes state at step t: $s_t \in S$ $a_t^{-i} \in \vec{A}$

produces action at step t: $a_t \in A(s_t)$

gets resulting reward: $r_{t+1} \in \Re$

and resulting next state: s_{t+1}

Joint-Action Learners: Opponent Modeling

- Maintain an explicit model of the opponents/team-mates for each state
- Q-values are updated for all possible joint actions at a given state
- Also, here the critical assumption is that the opponent is stationary
- Opponent modeling by counting frequencies of the joint actions they executed in the past
- Each agent would update its Q-values that involves joint action using the Bellman update:

$$Q^j(s, a^j, a^{-j}) \leftarrow Q^j(s, a^j, a^{-j}) + \alpha\left(r^j + \gamma \max_{a'^j} Q^j(s', a'^j, a'^{-j}) - Q^j(s, a^j, a^{-j})\right)$$

Simple JoinQLearning Algorithm

JointQlearning(s, Q)

Repeat

Repeat for each agent i

Select and execute a^i

Epsilon greedy or Boltzmann Exploration

Observe s', r^i and $\boldsymbol{a}^{-i}$, where $\boldsymbol{a}^{-i} = \{a^1, \ldots, a^{i-1}, a^{i+1}, \ldots, a^N\}$

Update counts: $n(s, \boldsymbol{a}) \leftarrow n(s, \boldsymbol{a}) + 1$

Update counts: $n_t^i(s, a_j) \leftarrow 1 + n_{t-1}^i(s, a_j),\ \forall j$

Learning rate: $\alpha \leftarrow \dfrac{1}{n(s, \boldsymbol{a})}$

Update Q-value:

$$Q^i(s, a^i, \boldsymbol{a}^{-i}) \leftarrow Q^i(s, a^i, \boldsymbol{a}^{-i}) + \alpha\left(r^i + \gamma \max_{a'^i} Q^i(s', a'^i, \mu^i(s', a_1), \ldots, \mu^i(s', a_N)) - Q^i(s, a^i, \boldsymbol{a}^{-i})\right)$$

$s \leftarrow s'$

Until convergence of Q^i

Variant Q-Learning

- Minimax Q-Learning
- Nash Q-Learning
- Correlated Q-Learning

References

- https://gki.informatik.uni-freiburg.de/teaching/ws0809/map/mas_lect11b.pdf

Questions?

Fakultas Informatika
School of Computing
Telkom University

THANK YOU