COPS RL reading grp

Aim: After a lot of thinking, we finally created this group, with a proper roadmap in our minds. You people have been invited, seeing the form responses and your activity in the talks. The aim of this group is to familiarize ourselves with the various subfields of RL for a few weeks so that we all are on the same page, and then decide upon a particular topic of mutual interest, and start collecting and working towards ideas, with the intention of getting a publication.

Useful Links:

Key points to remember:

1. Problems you may encounter in DeepRL: http://amid.fish/reproducing-deep-rl
2. Advice for short term machine learning research projects/papers: https://rockt.github.io/2018/08/29/msc-advice
3. OPENAI Spinning Up as a Deep RL Researcher: https://spinningup.openai.com/en/latest/spinningup/spinningup.html

Useful resources for a recap:

1. https://spinningup.openai.com/en/latest/spinningup/rl_intro.html
2. https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html
3. https://spinningup.openai.com/en/latest/spinningup/rl_intro3.html

Papers:

1. Deep RL: https://spinningup.openai.com/en/latest/spinningup/keypapers.html
2. Multiagent RL: https://github.com/LantaoYu/MARL-Papers#research-papers
3. Survey and Critique of MARL: https://arxiv.org/abs/1810.05587

1st Meet

Thursday: 28th Jan

Yash: Emergent Behaviours

Using Single Agent RL algos to train multiple agents in Multi Agent Settings and analyzing the behaviour of the agents during and after training

Playing Pong with DQN

Three modes: Competitive, Collaborative and Mixed

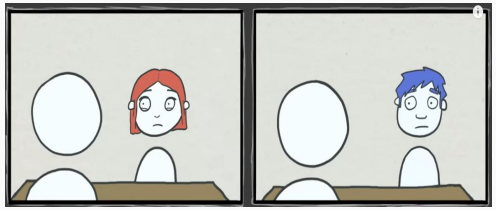
	Left player scores	Right player scores
Left player reward	+1	−1
Right player reward	−1	+1

	Left player scores	Right player scores
Left player reward	−1	−1
Right player reward	−1	−1

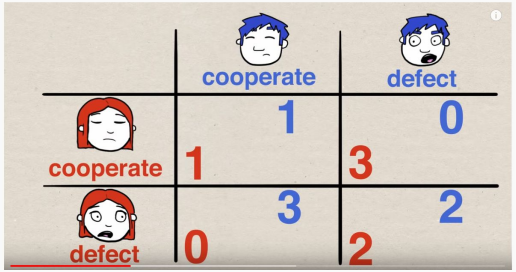
Analysis of:

1. Average paddle bounces per point
2. Average wall bounces per paddle bounce
3. Serving Time

Sequential Social Dilemmas



Markov Game Social Dilemmas



Prisoner's Dilemma - consists of 2 atomic actions - cooperate or defect. Both should cooperate with each other, but instead defect against each other.

Real World Dilemmas are different:

1. Temporally extended, hence need a cooperation and defection policy and not just a single action
2. Partially Observable
3. Have Many states

$M,\ \pi^C, \pi^D$

Fruit Gathering Game - Two agents trying to collect fruit which gives rewards. Can tag each other which doesn't give any reward but may help reduce the competition. Defecting Policy - Aggressive - Tagging. Cooperative Policy - No Tag.
When there was scarcity of resources and tagging reduced competition much more, aggressive policy. Otherwise cooperative. Increase Discount factor, more aggressive. Increase network size, more aggressive.

Fruit Gathering Game

Nishant: Learning to Communicate in MARL

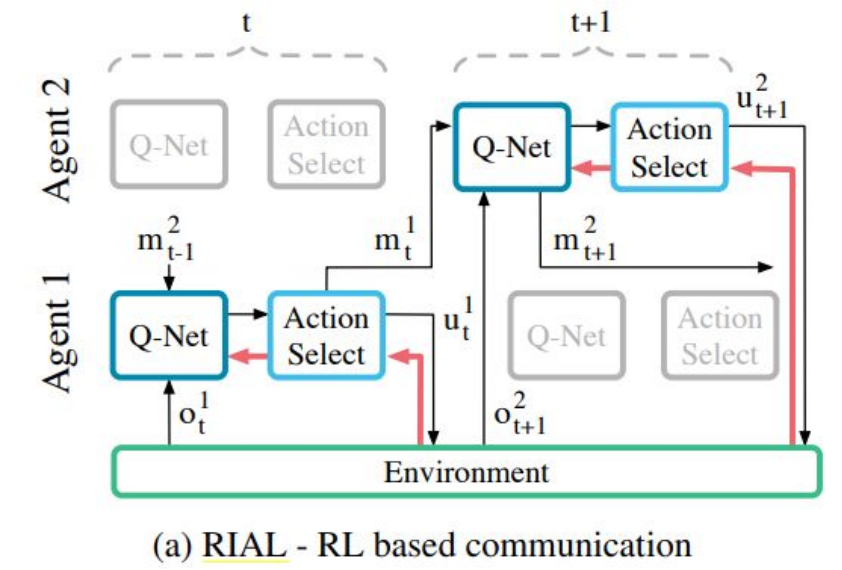
Communication Protocols

1. Direct messages: RIAL, DIAL, etc

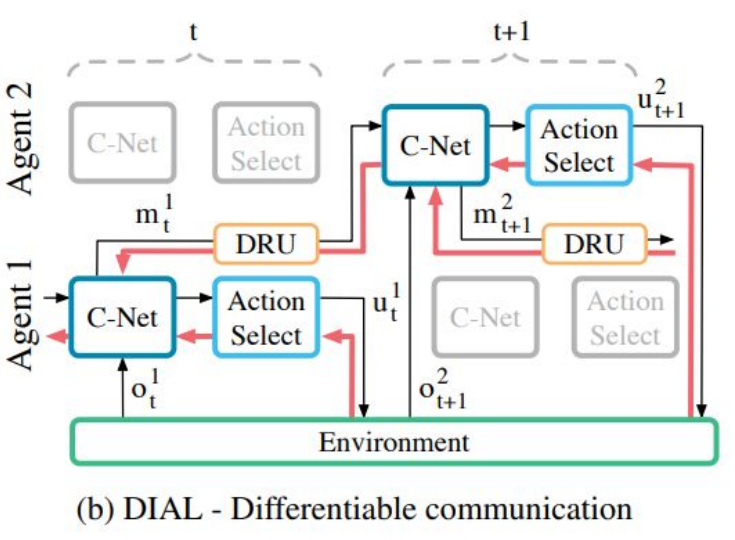
2. Via shared memory: Memory Driven - MADDPG

RIAL: Reinforced Inter-agent learning
DIAL: Differentiable Inter-agent learning

https://arxiv.org/pdf/1605.06676

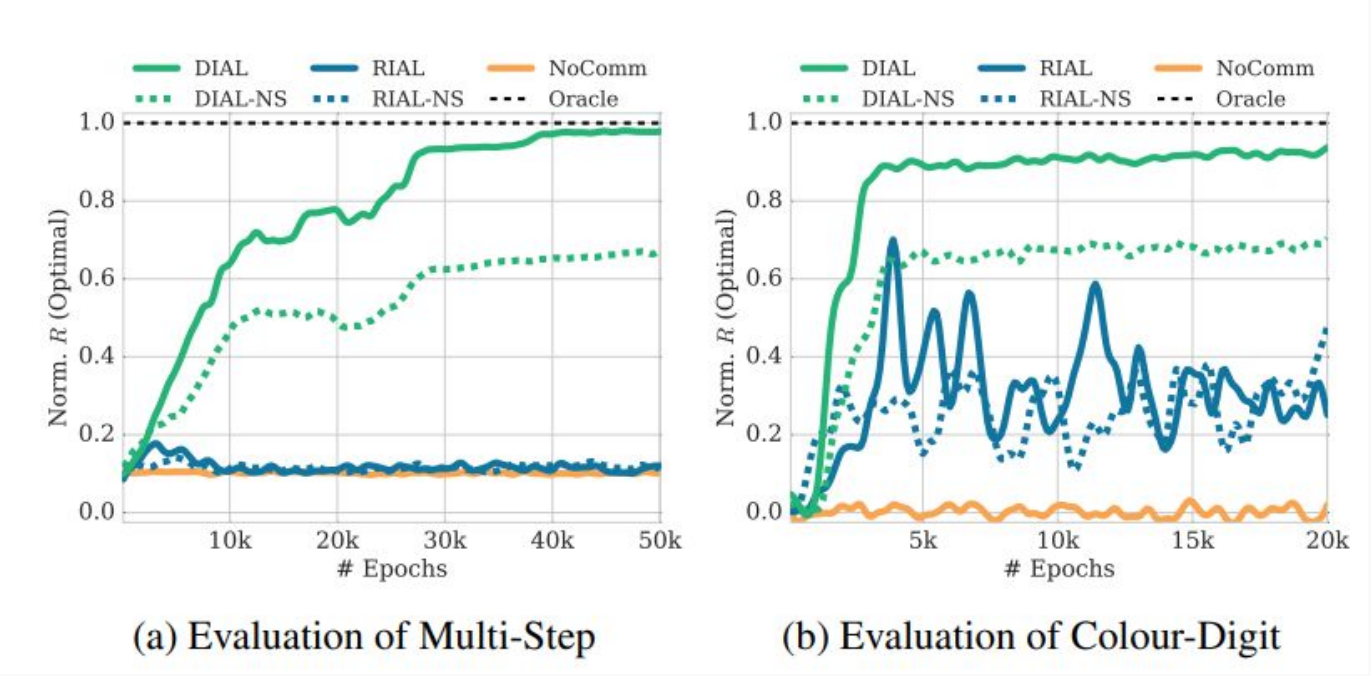


(a) RIAL - RL based communication



(b) DIAL - Differentiable communication

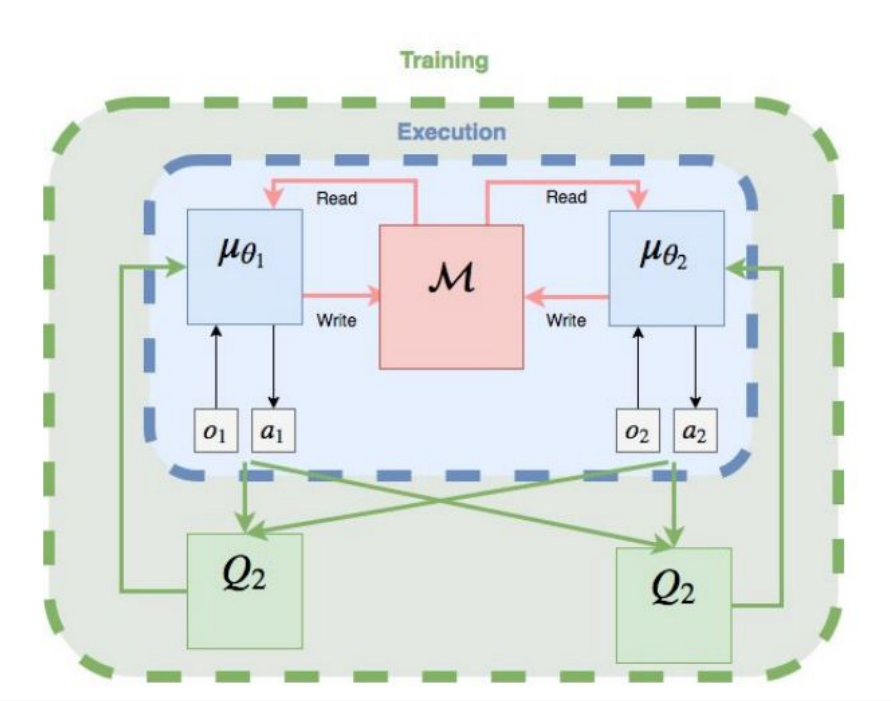
Contd..


DIAL
RIAL
NoComm
DIAL-NS
RIAL-NS
Oracle
Norm. R (Optimal)
1.0
0.8
0.6
0.4
0.2
0.0
10k
20k
30k
40k
50k
Epochs
(a) Evaluation of Multi-Step
DIAL
RIAL
NoComm
DIAL-NS
RIAL-NS
Oracle
Norm. R (Optimal)
1.0
0.8
0.6
0.4
0.2
0.0
5k
10k
15k
20k
Epochs
(b) Evaluation of Colour-Digit

$$\nabla_{\theta_i} J(\boldsymbol{\mu}_{\theta_i}) = \mathbb{E}_{\mathbf{x},a,\mathbf{m}\sim\mathcal{D}}\left[\nabla_{\theta_i}\boldsymbol{\mu}_{\theta_i}(\boldsymbol{o}_i,\mathbf{m})\nabla_{a_i} Q^{\boldsymbol{\mu}_{\theta_i}}(\mathbf{x},a_1,\ldots,a_N)|_{a_i=\boldsymbol{\mu}_{\theta_i}(\boldsymbol{o}_i,\mathbf{m})}\right]$$

Memory-driven MADDPG

https://arxiv.org/pdf/1901.03887



These read and write operations are quite involved!

Read: $\mathbf{r}_i = \zeta_{\theta_i^\zeta}(\boldsymbol{o}_i, \mathbf{m}).$

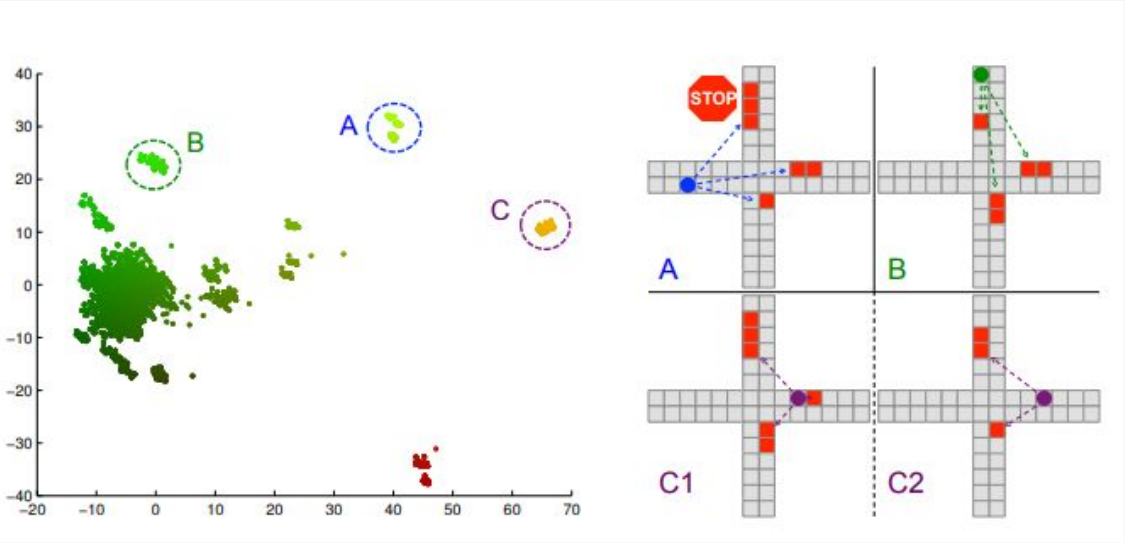
Write: $\mathbf{m}' = \xi_{\theta_i^\xi}(\boldsymbol{o}_i, \mathbf{m}).$

Action Selection:

$$a_i = \varphi^{act}_{\theta_i^a}(\mathbf{e}_i, \mathbf{r}_i, \mathbf{m}')$$

Simplest way of Communication Analysis: from the CommNet paper

https://arxiv.org/pdf/1605.07736.pdf



To better understand the meaning behind the communication vectors, we ran the simulation with only two cars and recorded their communication vectors and locations whenever one of them braked. Vectors belonging to the clusters A, B & C in Fig. 3(left) were consistently emitted when one of the cars was in a specific location, shown by the colored circles in Fig. 3(middle) (or pair of locations for cluster C). They also strongly correlated with the other car braking at the locations indicated in red, which happen to be relevant to avoiding collision.

More on communication:

R. Lowe, J. Foerster, Y.-L. Boureau, J. Pineau, Y. Dauphin, On the pitfalls of measuring emergent communication, in: 18th International Conference on Autonomous Agents and Multiagent Systems, 2019.

Discusses why some communication methods seem to work, but actually they dont.

Somnath : MultiAgent Cooperation

Formulating and using MaRL in Cooperative settings.

Experience Replay

The stochasticity of the system due to non optimal policy makes the experience meaningless in a later stage of training.

Two methods to fix it.

- Adaptive Importance Sampling
- Adding Fingerprints to the experience (ex :- training iteration number and exploration rate)

$$\prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}.$$

Parameter Sharing

Having individual Value function for each agent is noisy and not the most effective way of learning the Q function.

Hence we share the few initial layers for all the agents and then we use independent layers to learn specific rewards.

Let’s be optimistic

We often have negative rewards for stochastic noise in the policy, by which learning is hindered. Hence we have two methods

- Hysteretic Q Learning : - We simply use two different learning rates. α,β ($<\alpha$). Where β is used when we get a negative TD error, and α is used when we get a positive TD error(positive experience).
- Lenient Q Learning

$$l(s_t, a_t) = 1 - e^{-K * T_t(s_t, a_t)}$$

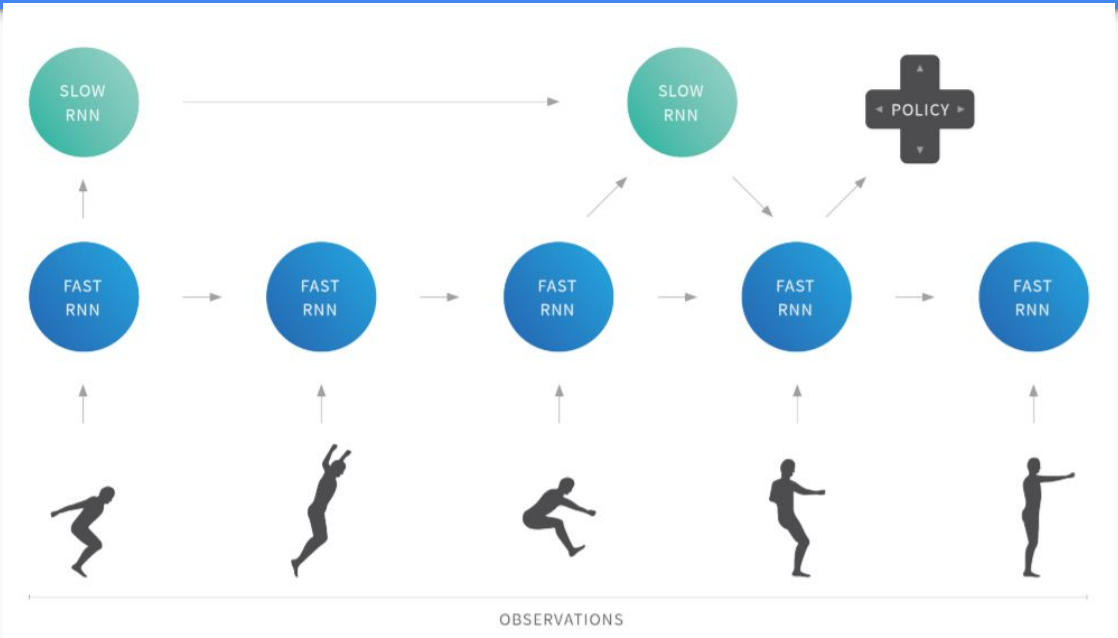
$$Q(s_t, a_t) = \begin{cases} Q(s_t, a_t) + \alpha\delta & \text{if } \delta > 0 \text{ or } x > l(s_t, a_t). \\ Q(s_t, a_t) & \text{if } \delta \leq 0 \text{ and } x \leq l(s_t, a_t). \end{cases}$$

Let’s be realistic

The previous two methods are over optimistic value functions because we learn the positive experiences, This is may not be good in the long Run hence we use **Double Q Learning.**

This makes the Q updates more stable.

Hierarchical Approach



Hierarchical approach in time domain.

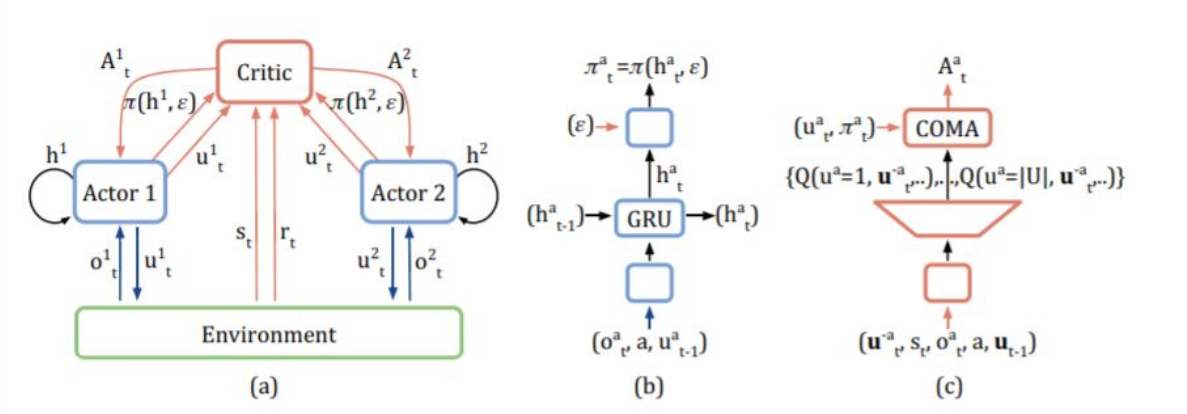
contd.

Hierarchy in observation.

MADDPG and COMA

MADDPG : - This has a single critic model and Multi Actor models

COMA : - This has a single critic model which has counterfactual rewards for each Agent. With multiple actor models.

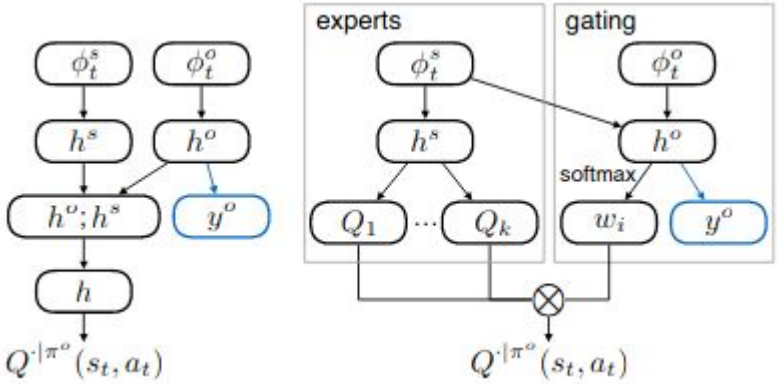


Ayush: Agents Modeling Agents

Deep Reinforcement Opponent Network

- Learning policy representation of the opponents collectively
- Integrated with Q Learning using
  - Simple concat
  - Mixture of Experts

$$Q^{\pi|\pi^o}(s_t, a_t) = \sum_{o_t} \pi_t^o(o_t|s_t) \sum_{s_{t+1}} \mathcal{T}(s_t, a_t, o_t, s_{t+1})$$
$$\left[\mathcal{R}(s_t, a_t, o_t, s_{t+1}) + \gamma \mathbb{E}_{a_{t+1}}\left[Q^{\pi|\pi^o}(s_{t+1}, a_{t+1})\right]\right].$$

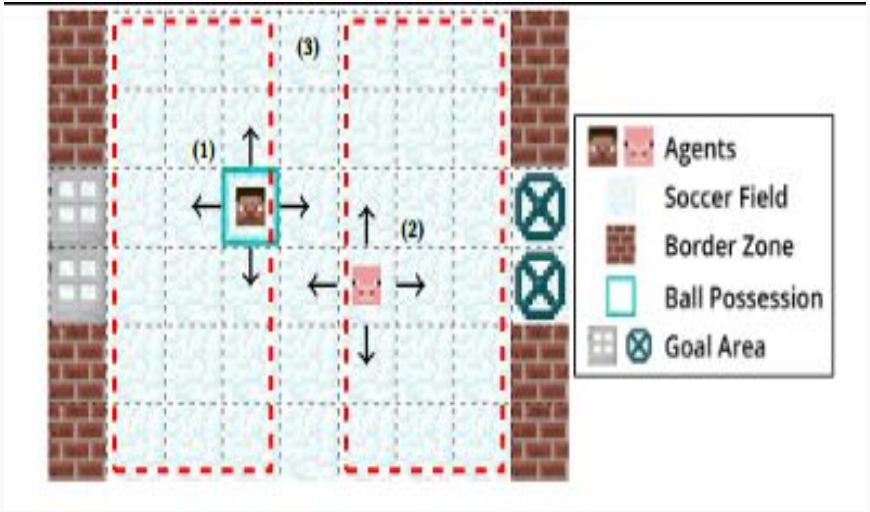


https://arxiv.org/pdf/1609.05559.pdf

Deep Policy Inference Q-Network for Multi-Agent Systems

- Learns a hidden representation for opponents policy parameters from raw pixel data and uses it to predict agent's Q values
- Uses adaptive loss in training, the loss is based on auxiliary tasks like learning own Q values or opponents policy parameters at different stages of training.

https://arxiv.org/pdf/1712.07893.pdf



Shravan: Multi agent RL (basics)

Single Agent case

- We already know the value and the Q function and how they are updated.
- Exploration is necessary for convergence:

$$h(x,u) = \frac{e^{Q(x,u)/\tau}}{\sum_{\bar{u}} e^{Q(x,\bar{u})/\tau}}$$

- Corresponds to Boltzmann exploration where **h** is the policy and **$\tau > 0$** is the temperature which controls exploration.

Multiagent: Extension of single agent

- Things remain same as the single agent case but now the reward, Q and value functions depend on joint actions (called as stochastic game)

$$R_i^{\boldsymbol{h}}(x) = \mathrm{E}\left\{ \sum_{k=0}^{\infty} \gamma^k r_{i,k+1} \,\middle|\, x_0 = x, \boldsymbol{h} \right\}$$

$$Q_i^{\boldsymbol{h}}(x, \boldsymbol{u}) = \mathrm{E}\left\{ \sum_{k=0}^{\infty} \gamma^k r_{i,k+1} \,\middle|\, x_0 = x, \boldsymbol{u}_0 = \boldsymbol{u}, \boldsymbol{h} \right\}.$$

- From a single agent perspective the environment is not stationary.

Static games and Nash Equilibrium

- A static game is a stochastic game with no state signal or dynamics. Can be zero sum, general sum.
- Here the policy losses the state argument and transforms into a strategy.
- In a static game all the agents tend to attain nash equilibrium.
- Nash equilibria is a joint strategy $[\sigma^*_1 ,...,\sigma^*_n]^T$ such that each individual strategy is the best response to other.
- In this scenario no agent can benefit from changing its strategy as long as all the other agents keep their strategy constant.
- All static games have at least one Nash equilibria.