CN Assignment 2

Name: Erwin Wu
Affiliation: Koike Lab
Student ID: 17M38147

Summary on Multi-Agent System

Multi-Agent System (MAS) is a system consists of numerous agents which can make decisions autonomously and solve problems collectively. We can often see its application in the field of Groups of autonomous vehicles or Wireless sensor networks.

Multi-agent systems are a possible way of expressing properties and architecture of agent arrangements. Generally speaking, it's an area of Distributed Artificial Intelligence that deals with how autonomous, distributed, and intelligent systems as a unit coordinate their specific knowledge, goals, capabilities, and plans to act in coordination or solve problems.

There are, for example, collaborative agent systems that are deliberately distributed in their architecture to perform more flexible and reliable tasks than a single, localized system. It can solve a problem out of the interaction of the actors, which a single unit would never be able to do. These units are each responsible for an activity, there is no need for overarching control, and together they find solutions through self-organized coordination.

Brief explanation of the types of agents:

- Collaborative Agents: Goals are achieved through collaboration and negotiation with other agents, mostly built as multi-agent systems. Cooperativity and autonomy are in the foreground, but they are often self-learning.
- Interface Agents: Communicate mostly with a human being as a system user. Also help on the internet to find offers and to negotiate specialized. These agents do not include Help wizards.
- Smart Agents: Have all the qualities and able to handle a wide variety of tasks.

In conclusion, the main purpose of MAS is to use multi agent models with different algorithms (or characteristic) to simulate a complex network e.g. human society.

While for my research in the field of Deep Learning, it is also possible to combine it with Multi-Agent System. As we all know, deep learning is based on deep neural networks. For most cases, the construction of the network's units (nodes and edges) will not change during the learning. That's because deep learning used to be supervised or unsupervised learning, while both need correct answers of the training result to modify and optimize its accuracy.

But nowadays, there is a kind of reinforcement learning in which an agent interacting with its environment is attempting to learn an optimal control policy. So, we can say deep reinforcement learning is using the method of dynamic programing. In detail, for each time step, the agent observes a state s, chooses an action α, receives a reward r, and transitions to a new state s'. Just like the Figure1. below:

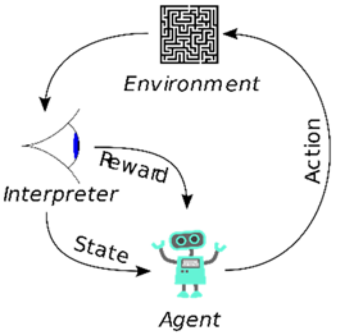

Figure1. Deep Reinforcement Learning[1]

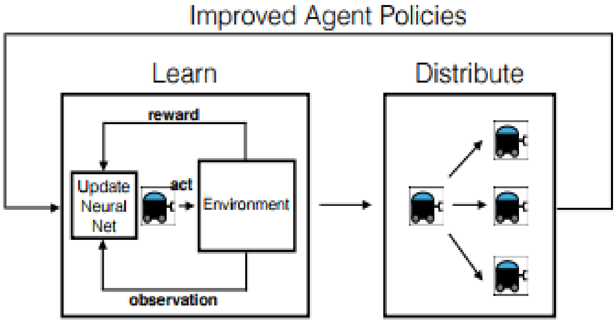

Figure2. Schematic of the multi-agent control training process[2]

In other word, since it is necessary for the agent to interacting with the environment, it is also possible to do that with multi-agent control. Actually, M.Egorov[2] from Stanford already tried to apply Multi-Agent System to Deep Reinforcement Learning. Hereby I introduce some problems they listed and how they managed to solve the problems.

The first challenge of applying multi-agent to reinforcement learning is problem representation. Specifically, the challenge is in defining the problem in such a way that an arbitrary number of agents can be represented without changing the architecture of the deep Q-Network (The deep neural network they used for reinforcement learning).

To solve this problem, they made a two-dimensional representation of the environment with discrete time and space, and limited the types of agent to two (allies and opponents), which assuming competing agents (see Figure2 for reference). These assumptions make it possible to represent the global system state as an image-like tensor, with each channel of the image containing agent and environment specific information (Fig.4):

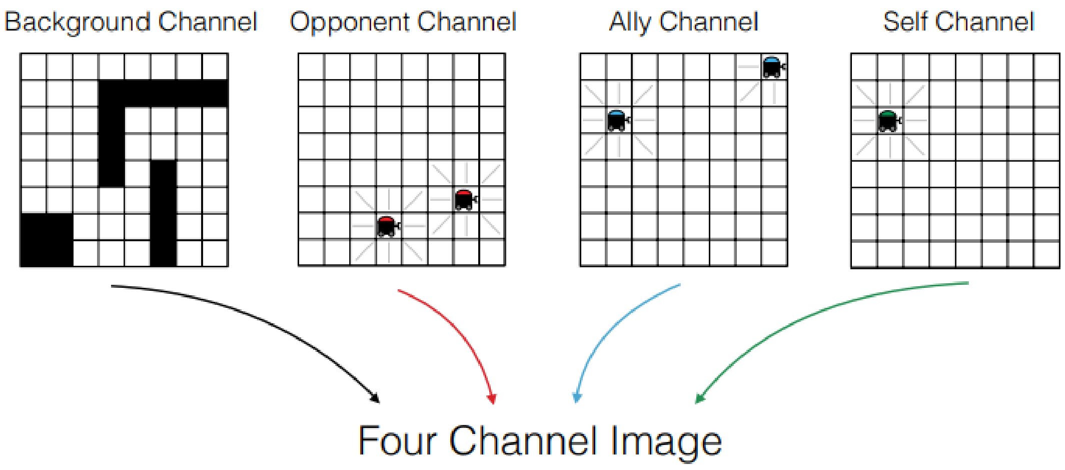

Figure 4. Four channel image-like representation of the multi-agent state used as input to the deep Q-Network.[2]

Another problem is multi-agent training, when multiple agents are interacting in an environment, their actions may directly impact the actions of other agents. To that end, agents must be able to reason about one another in order to act intelligently.

To incorporate multi-agent training, they train one agent at a time, and keep the policies of all the other agents fixed during this period. After a set number of iterations the policy learned by the training agent gets distributed to all the other agents of its type. Specifically, an agent distributes its policy to all its allies (just like Figure 2).

Last but not least, I also realized a problem for multi-agent reinforcement learning which was not mentioned by Egoroy[2], the agent ambiguity. In case where two (or more) ally agents are occupying the same position in the environment, the image-like stat representation for each agent might be identical, so their policies will be exactly the same. There are some solutions to break this symmetry like enforce the agents not to occupying the same place or make a stochastic policy for the agents, which was also mentioned by the teacher in the class.

This course about Multi-Agent Systems really benefits me a lot and I'll try to apply it to my research, of course after doing some experiments to solve the problem mentioned above.

[1] EGOROV, Maxim. Multi-Agent Deep Reinforcement Learning. 2016.

[2] Wikipedia contributors. Multi-Agent System. Jan 21, 2018. Available at: https://en.wikipedia.org/wiki/Multi-agent_system