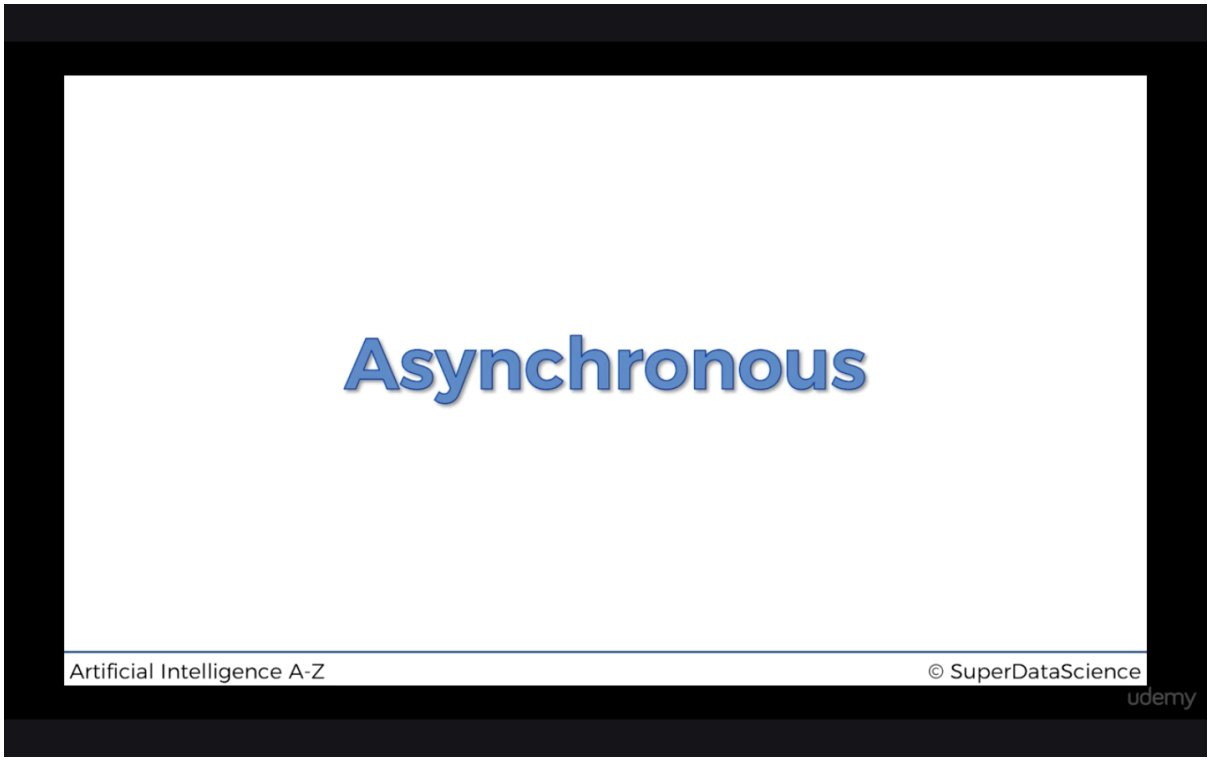
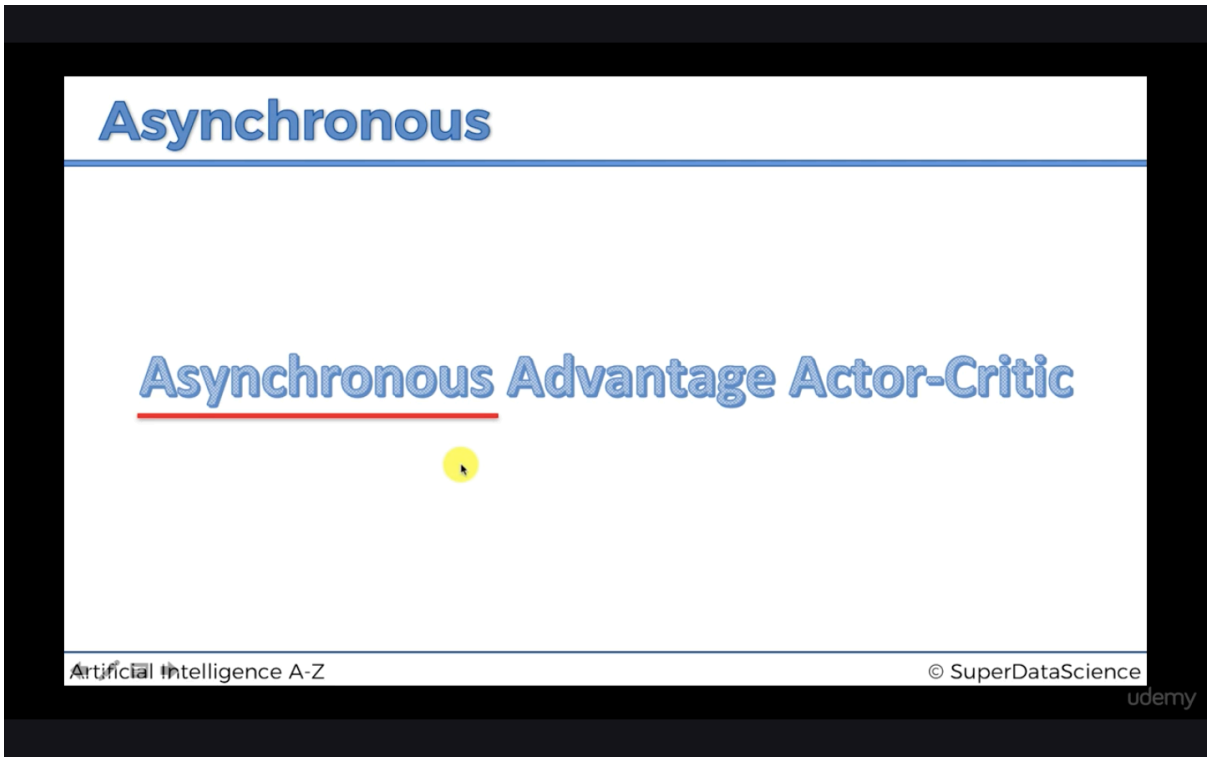
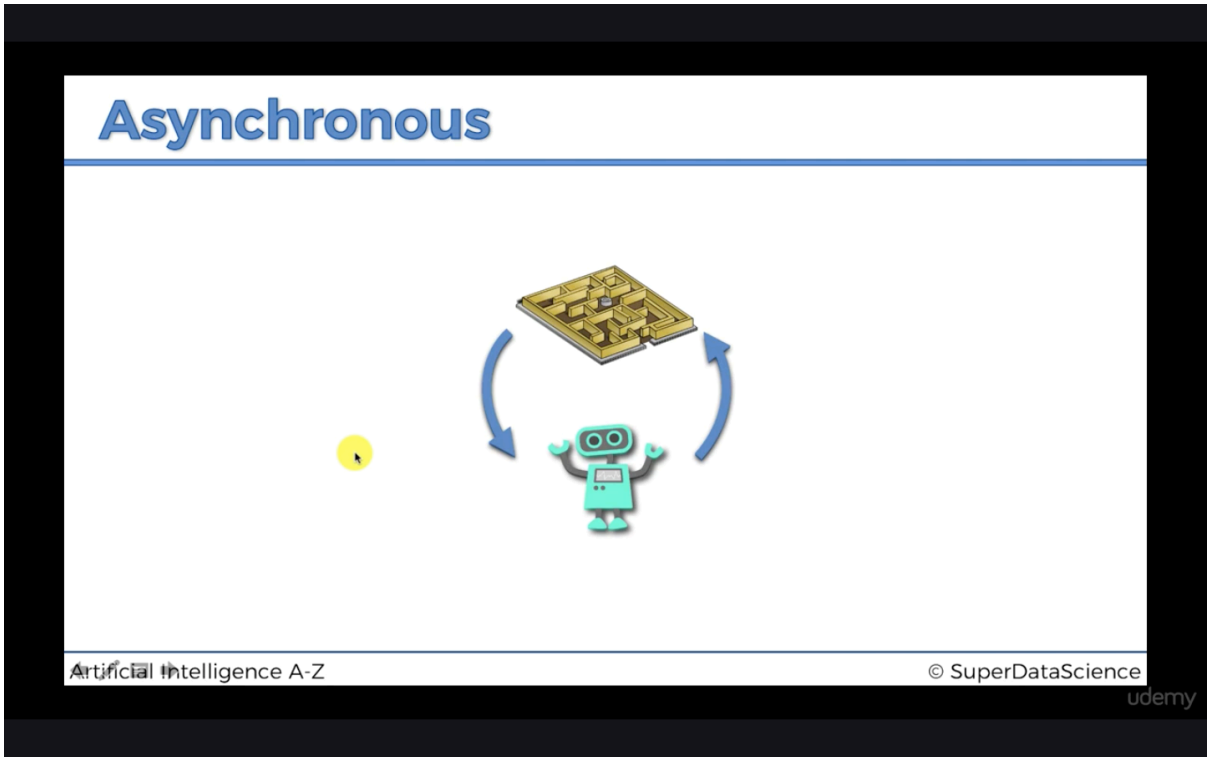
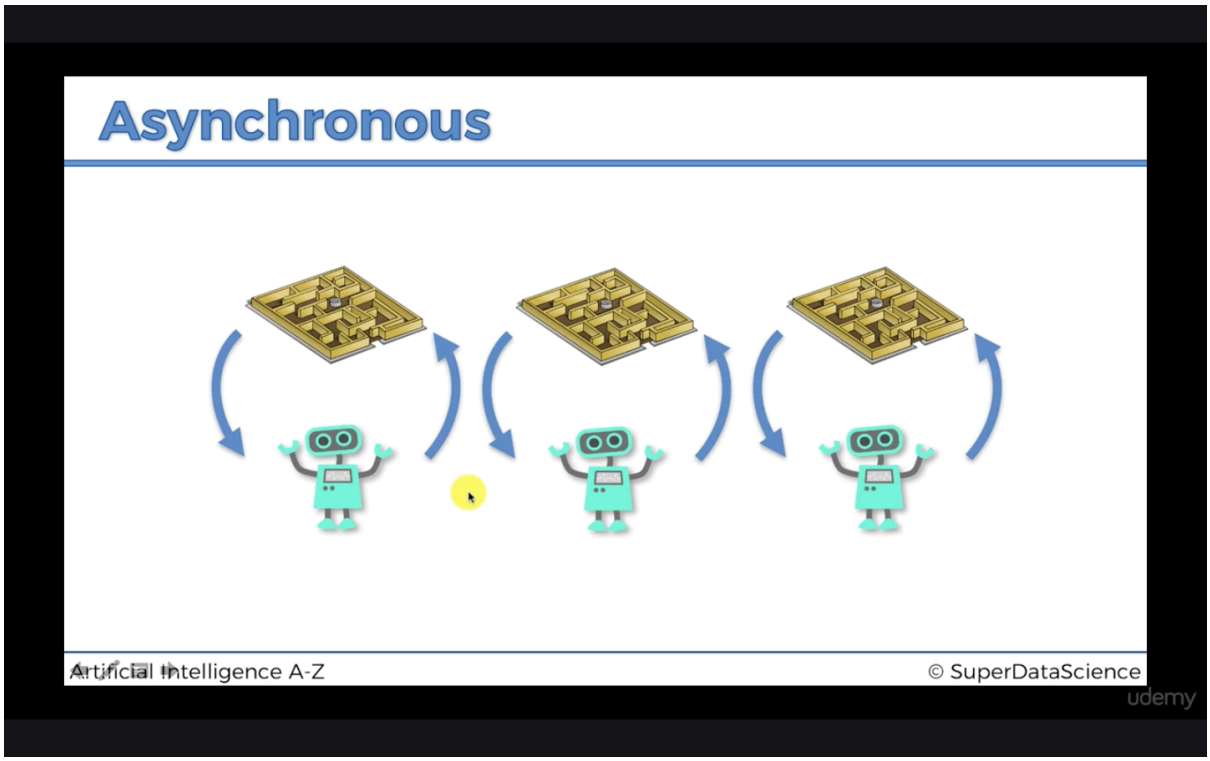
Asynchronous

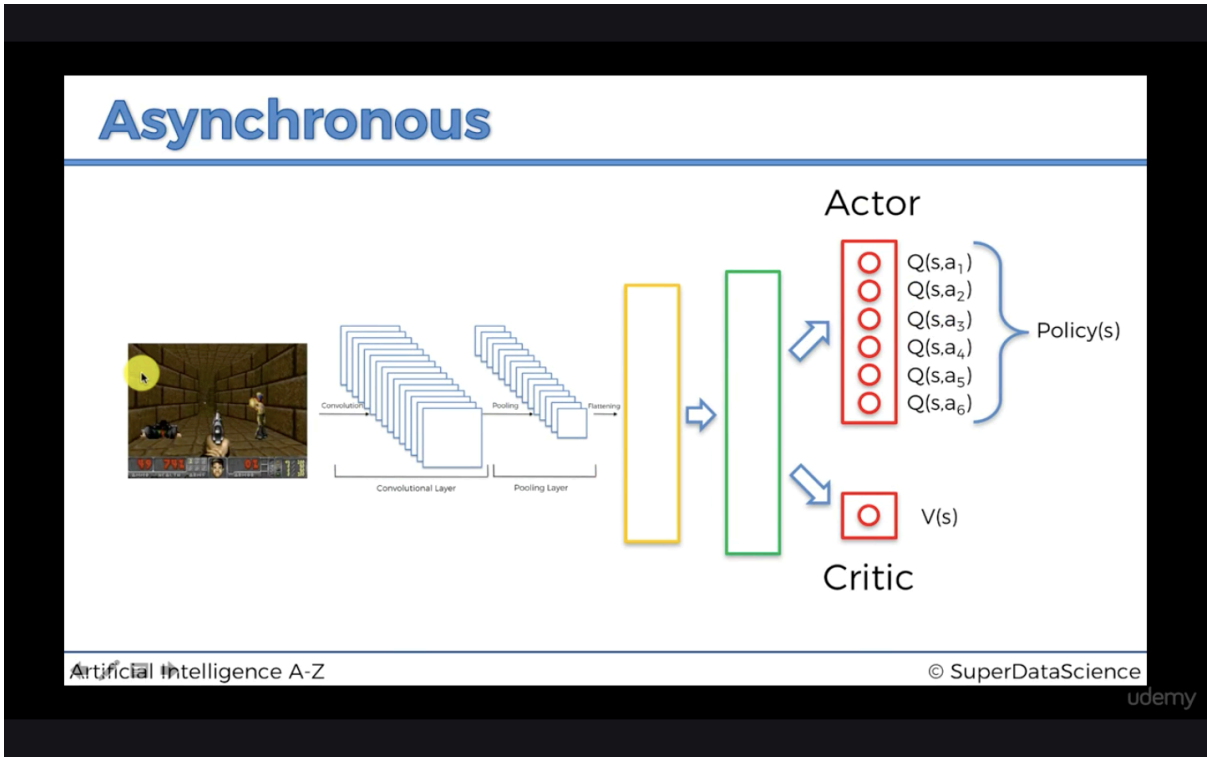


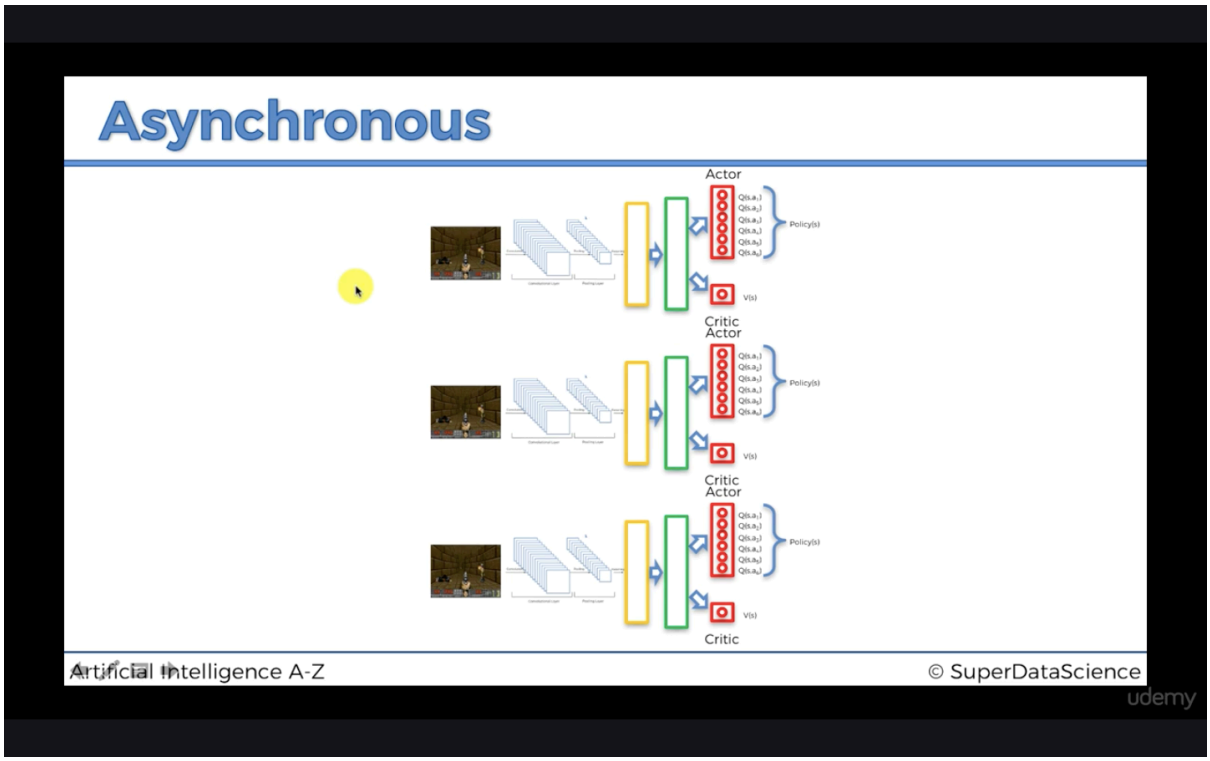




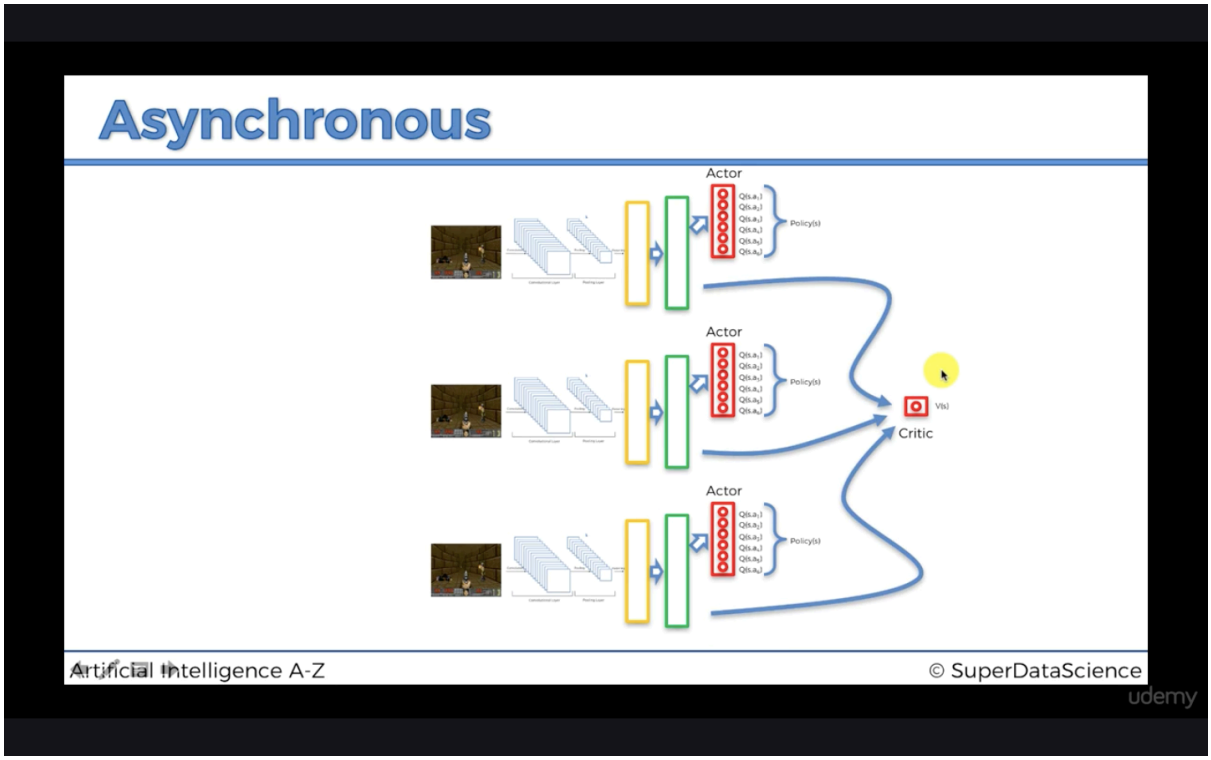


In asynchronous instead of having 1 agent attacking we have several agents to attack the same environment.
Starting point for each is different and explore in the different ways and triple the amount of experience.
It also reduces the chances that the agent stuck to the local maximum. The likelihood of getting stuck with several agents reduces.

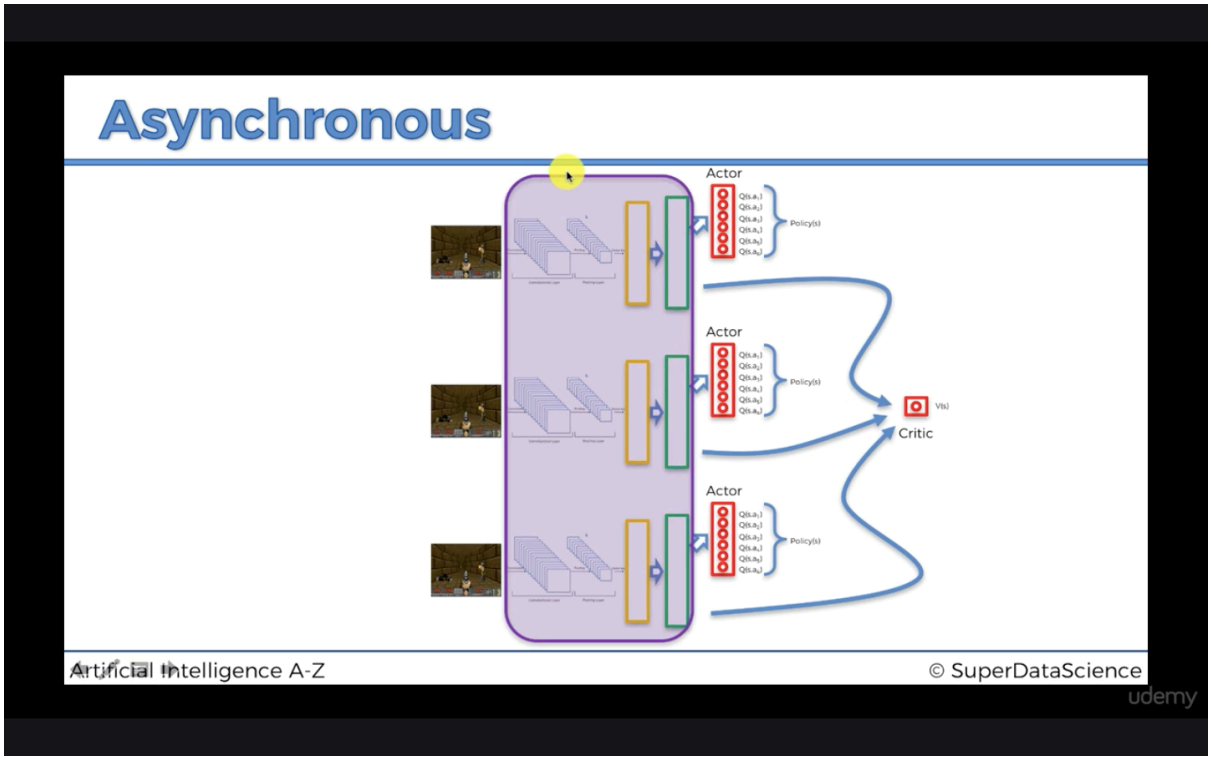


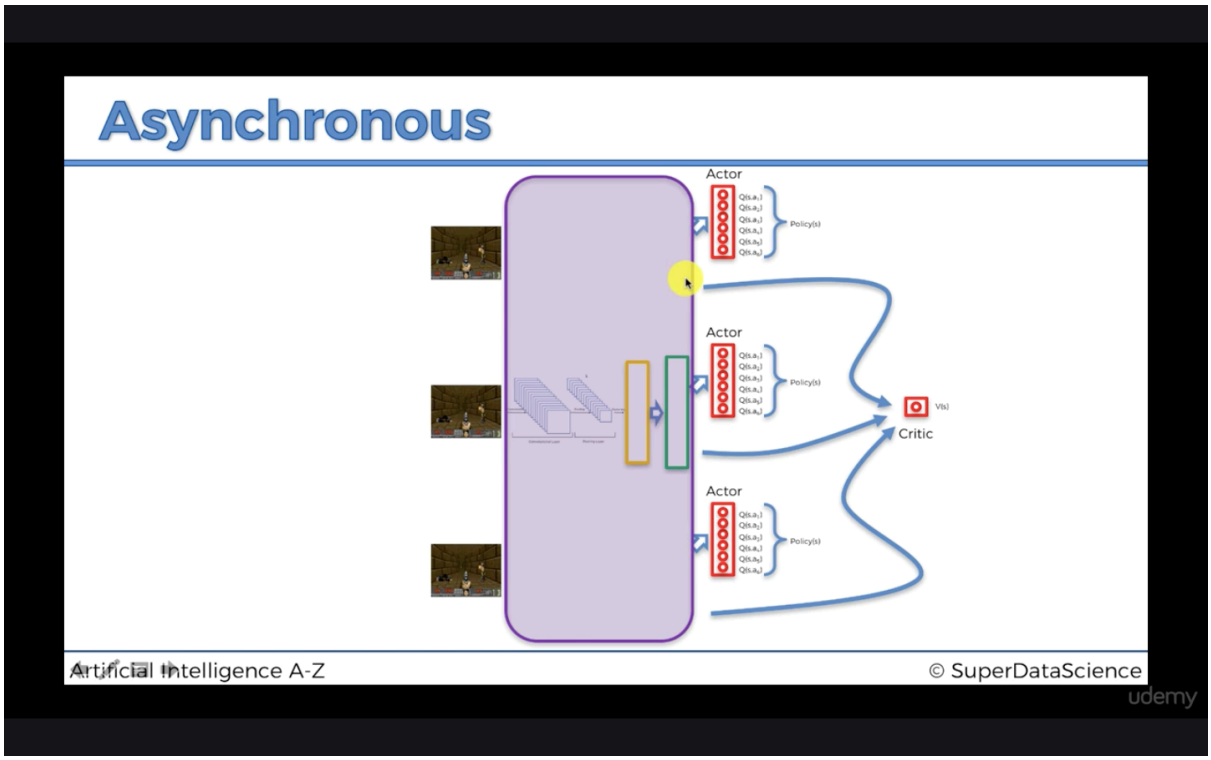


In here we are going to replicate this in three parts which each of these three is going to initialize differently and also, they are going to share experience through the V(s) we have calculated.



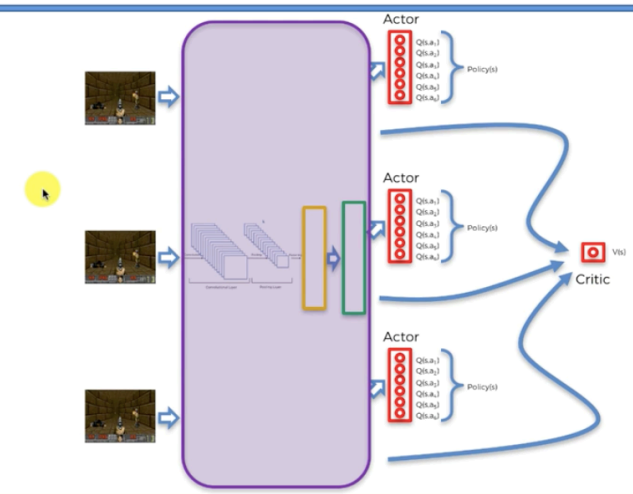
All agents contributing to the same critic and they don't have different critics.
The V is going to be calculated through the values or reward that we get through the environment. So as the agent explores the environment, they are predicting the V through the rewards that they know exist in this maze. Of course, as we explore that value can be change. As the agents go through this they will adjusting their neural network in order to better match that expected V. so critic part is shared between the agents and that's how they share information between each other. In order to use that to optimize how they're behaving in the environment.





After some modification that the creator of PyTorch did. Now we only use one neural network shared among the agents. Before it was one network for each actor and critic. Now there is one neural network for all actor and critic.

Asynchronous



Additional Reading

Additional Reading:

Let's Make An A3c: Implementation

Jaromír Janisch (2017)

Link:

https://jaromiru.com/2017/03/26/lets-make-an-a3c-implementation/