Asynchronous Advantage Actor-Critic (A3C) Algorithm

- The **Asynchronous Advantage Actor Critic (A3C)** algorithm is one of the newest algorithms to be developed under the field of Deep Reinforcement Learning Algorithms.

- This algorithm was developed by **Google's DeepMind** which is the Artificial Intelligence division of Google.

- This algorithm was first mentioned in 2016 in a research paper appropriately named Asynchronous Methods for Deep Learning.

- Decoding the **different parts** of the algorithm's name:

- **Asynchronous:** Unlike other popular Deep Reinforcement Learning algorithms (like Deep Q-Learning which uses a **single agent and a single environment**), this algorithm uses **multiple agents with each agent having its own network parameters and a copy of the environment**. These agents interact with their respective environments **Asynchronously**, learning with each interaction.

- Each agent is controlled by a **global network**. As each agent gains more knowledge, it **contributes to the total knowledge of the global network**.

- The presence of a global network allows **each agent to have more diversified training data**.

- This setup mimics the **real-life environment** in which, humans live as **each human gains' knowledge from the experiences of some other human** thus allowing the whole "global network" to be better.

- **Actor-Critic:** Unlike some simpler techniques which are based on either Value-Iteration methods or Policy-Gradient methods, the A3C algorithm combines the best parts of both the methods i.e. the algorithm predicts both the value function **V(s)** as well as the optimal policy function **π(s)**.

- The learning agent uses the value of the **Value function (Critic)** to update the optimal **policy function (Actor)**. Here, the policy function means the **probabilistic distribution of the action space**.

- **Advantage:** Typically, in the implementation of **Policy Gradients**, the value of Discounted Policy Values Returns to tell the agent which of its actions were rewarding and which ones were penalized. But in A3C, by using **the value of Advantage** instead, the agent also learns how much better the rewards were than its expectation.

- The **advantage metric** is given by the following expression:

$$\boldsymbol{A = Q(s, a) - V(s)}$$

- This gives a new-found insight to the agent into the environment and thus the **learning process is better**.

-