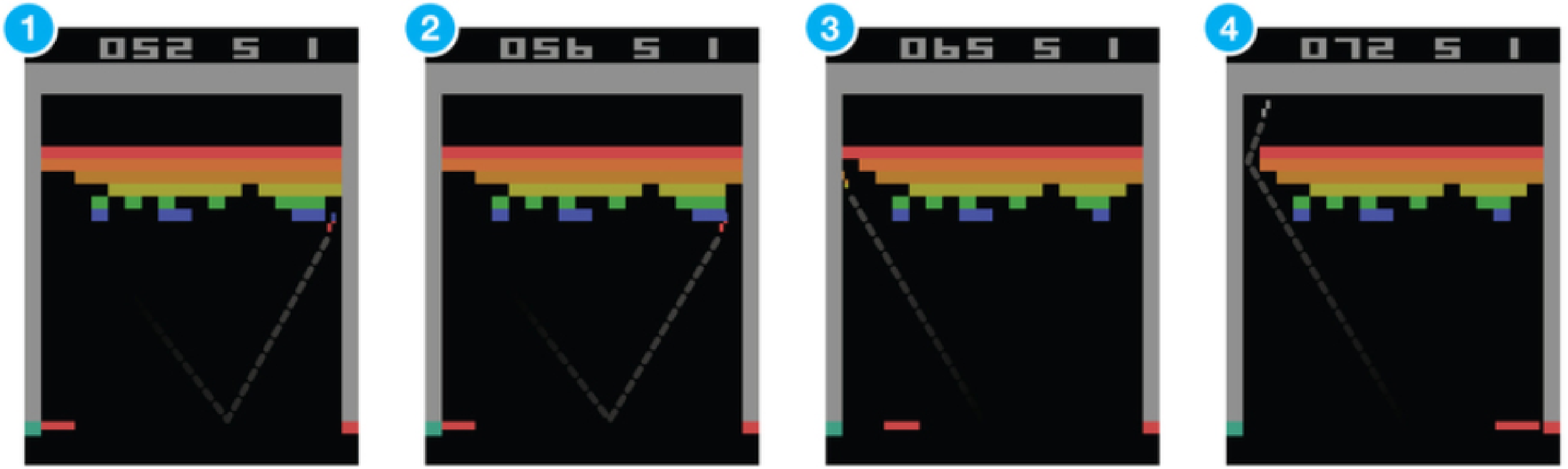
Demystifying Reinforcement Learning

By AKASH S VORA

WHAT IS REINFORCEMENT LEARNING?

LET'S LOOK AT AN EXAMPLE!

Consider the game Breakout.
In this game you control a paddle at the bottom of the screen and have to bounce the ball back to clear all the bricks in the upper half of the screen. Each time you hit a brick, it disappears and your score increases – you get a reward.

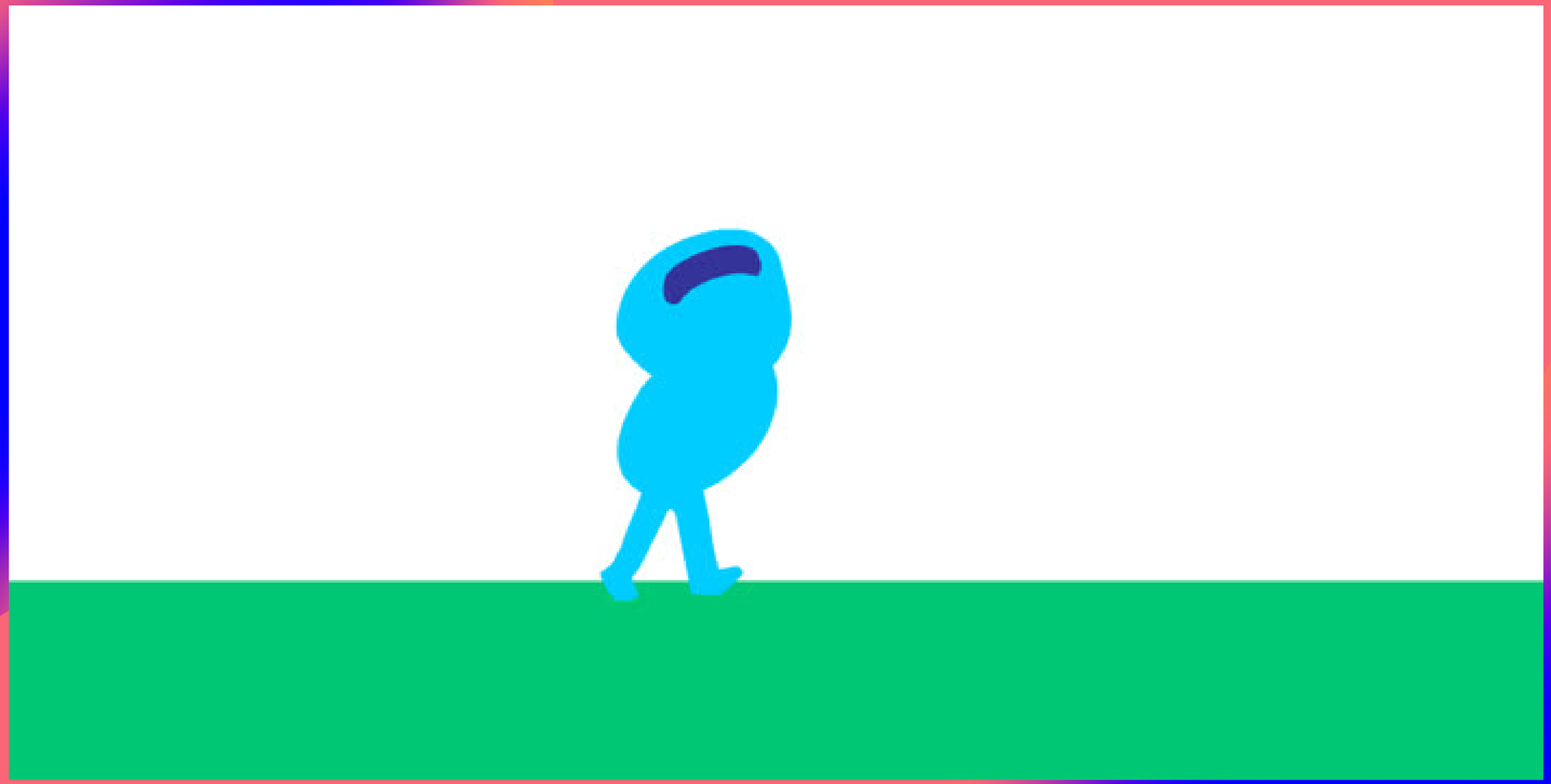


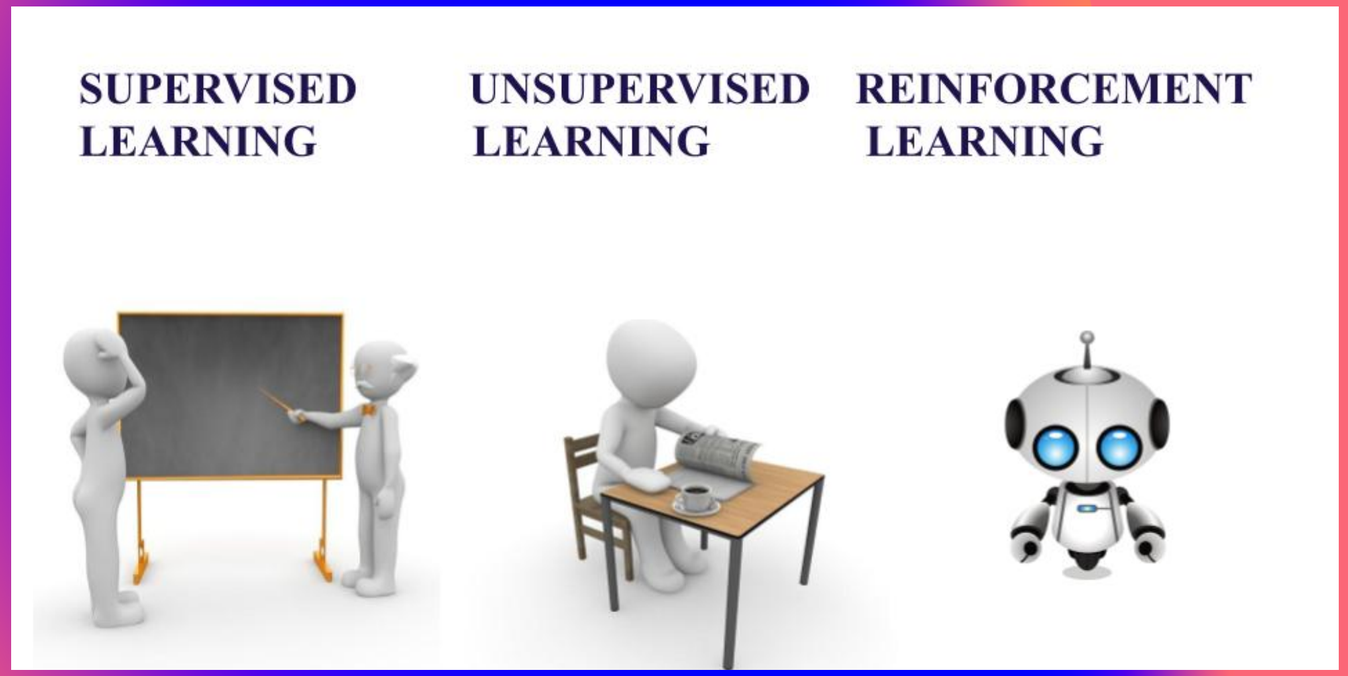
Suppose you want to teach a Neural Network to play this game.
Input to your network would be screen images, and output would be three actions : **left, right or fire** (to launch the ball). It would make sense to treat it as a classification problem – for each game screen you have to decide, whether you should move left, right or press fire.

Sounds straightforward?

Yes! But then you need training examples, and a lots of them. Of course you could go and record game sessions using expert players, but that’s not really how we learn. We don’t need somebody to tell us a million times which move to choose at each screen. We just need occasional feedback that we did the right thing and can then figure out everything else ourselves.

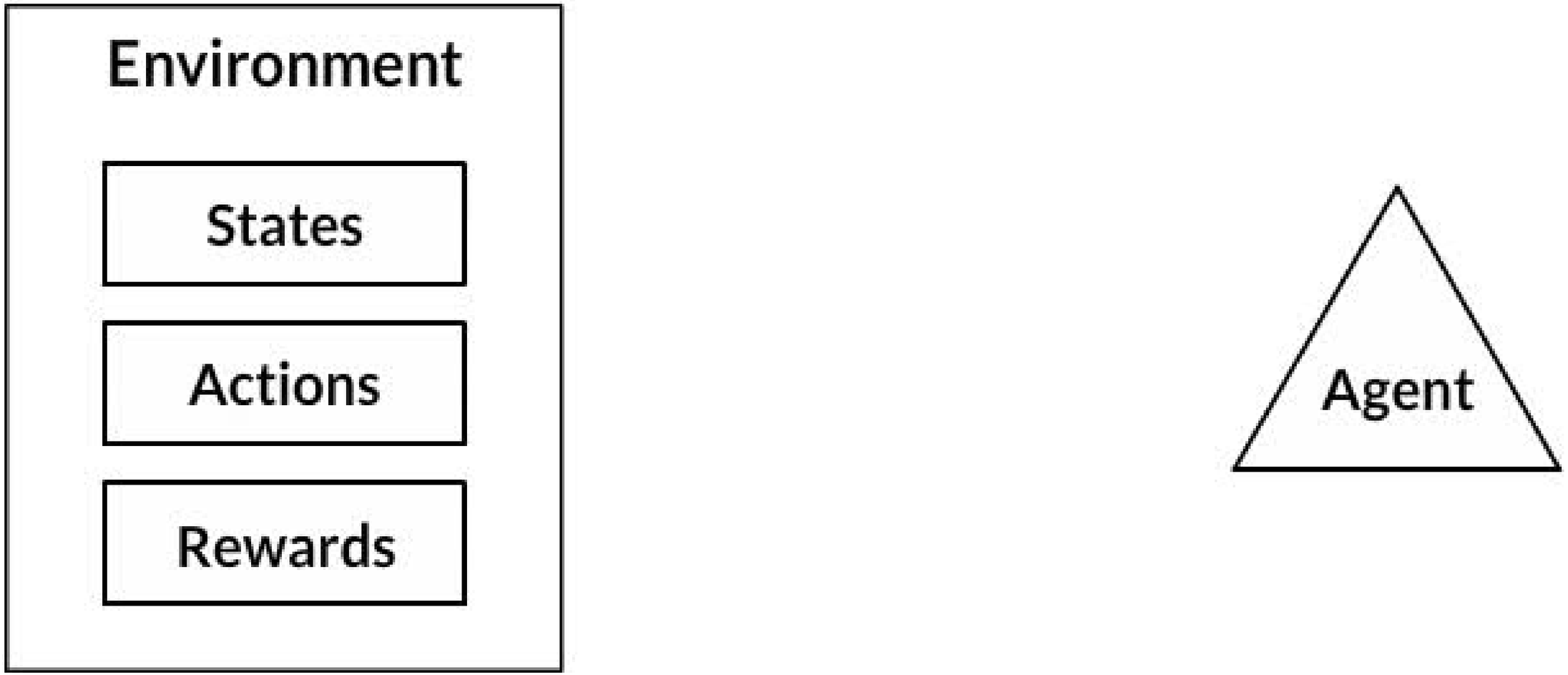
This is the task Reinforcement Learning tries to solve.





Reinforcement learning lies somewhere in between supervised and unsupervised learning. Whereas in supervised learning one has a target label for each training example and in unsupervised learning one has no labels at all, in reinforcement learning one has sparse and time-delayed labels – the rewards. Based only on those rewards the agent has to learn to behave in the environment.

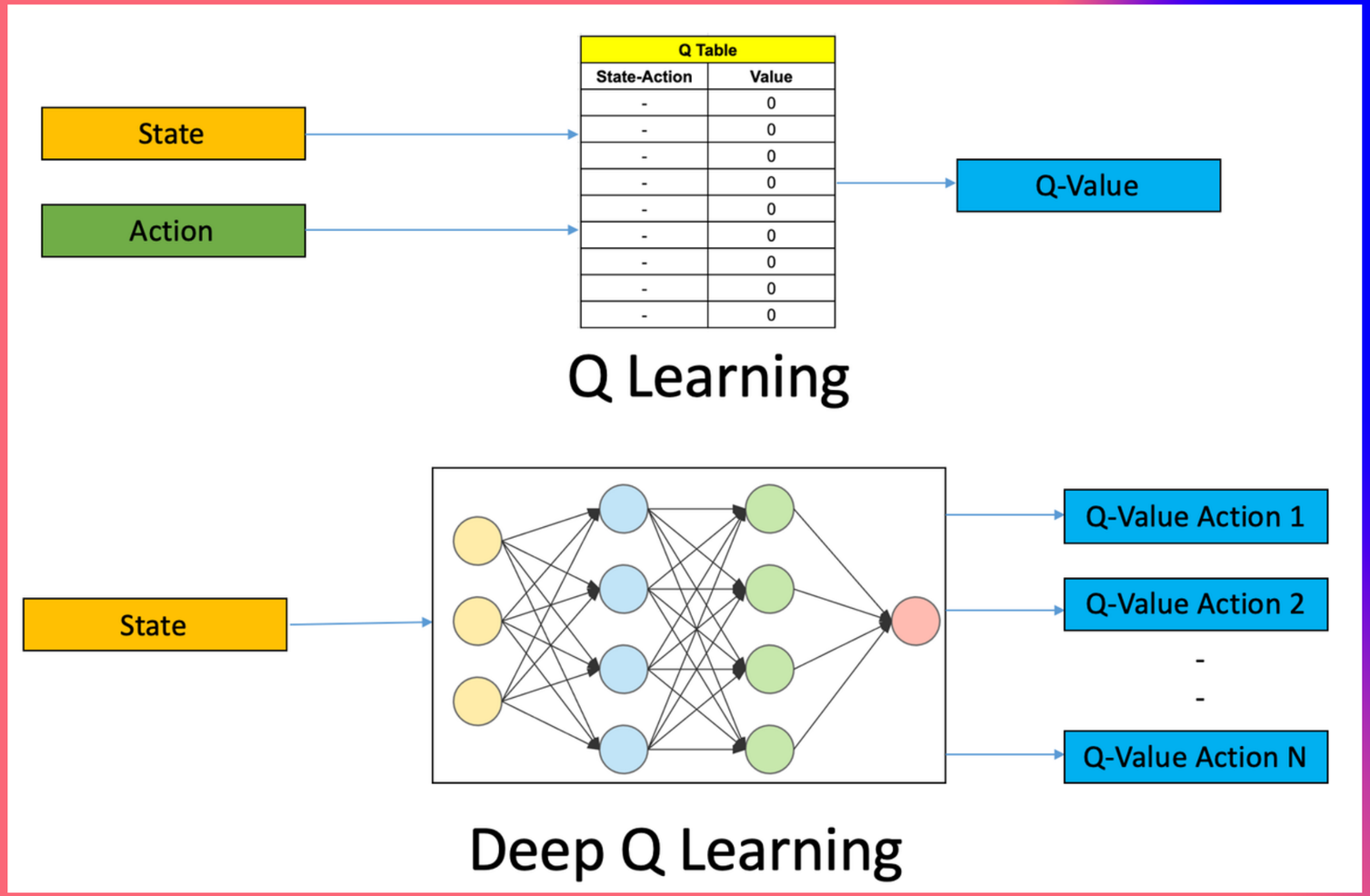
Reinforcement learning is an important model of how we (and all animals in general) learn. Praise from our parents, grades in school, salary at work – these are all examples of rewards. Credit assignment problems and exploration-exploitation dilemmas come up every day both in business and in relationships. That's why it is important to study this problem.

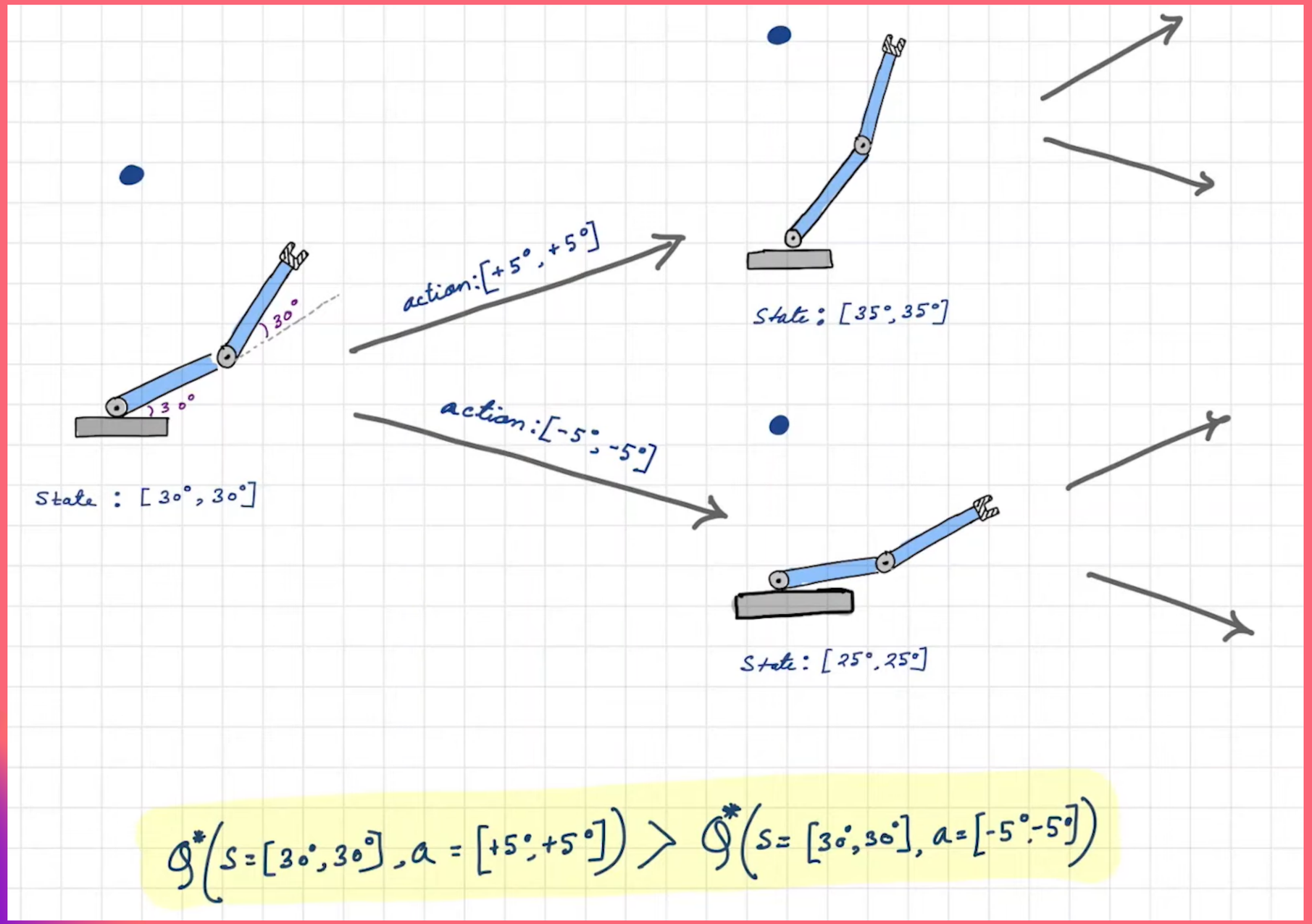

Environment
States
Actions
Rewards
Agent

Q Learning

Q-learning is a model-free reinforcement learning algorithm.

Q-learning is a values-based learning algorithm. Value based algorithms updates the value function based on an equation

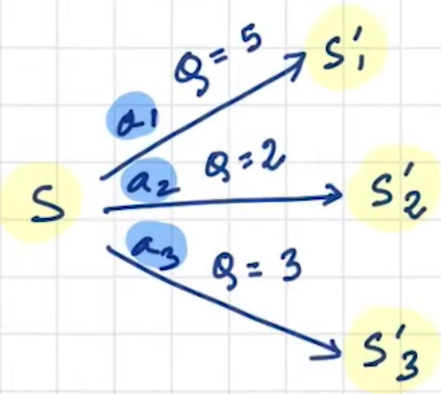




Let's take an Example!

Policy

↳ What action at state S?



$$action = \underset{a}{\text{argmax}}\ Q(s,a) \quad \dots\dots\rightarrow \text{Greedy}$$

$$action = \begin{cases} \underset{a}{\text{argmax}}\ Q(s,a), & \text{with } P = 1-\epsilon \\ \text{random action}, & \text{with } P = \epsilon \end{cases} \quad \dots\rightarrow \epsilon\text{-Greedy}$$

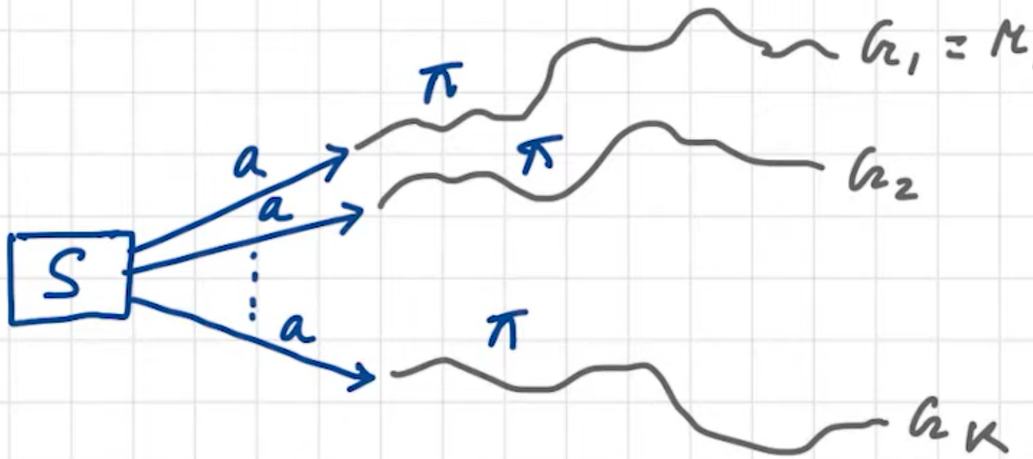
$$action = \text{random action} \quad \dots\dots\rightarrow \text{Random}$$

Q-VALUE

$Q_\pi(s, a) = ?$

π is policy.

$0 \leq \gamma \leq 1$ discount factor.



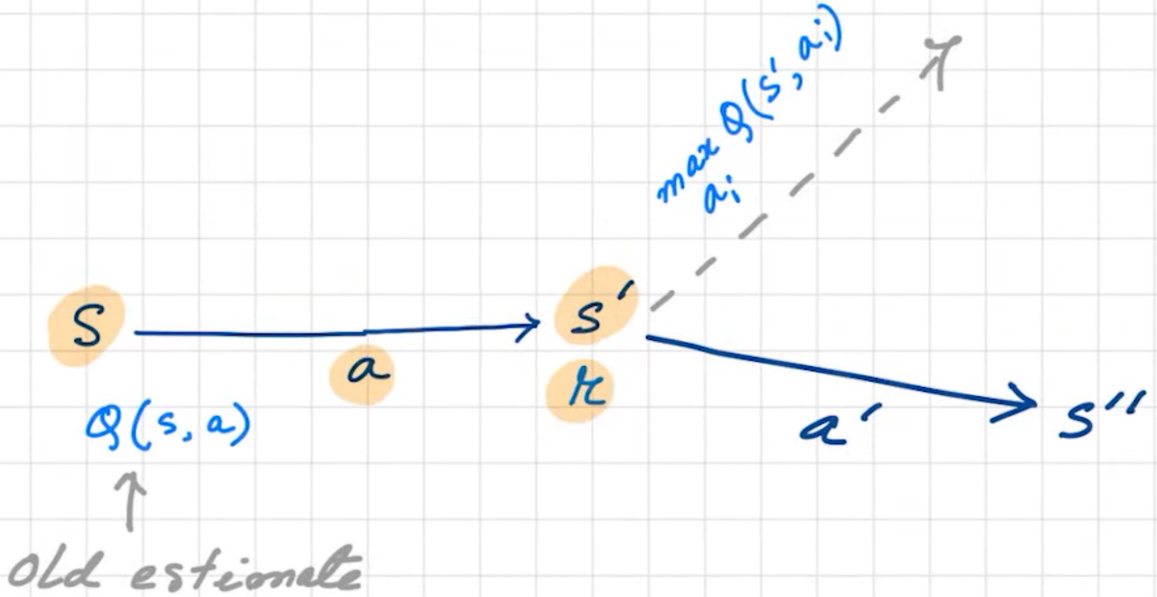
$G_1 = r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n r_n$

G_2

G_K

$$Q_\pi(s, a) = \mathbb{E}[G_t] \approx \frac{1}{K} \sum_i G_i$$

Q - Learning update.

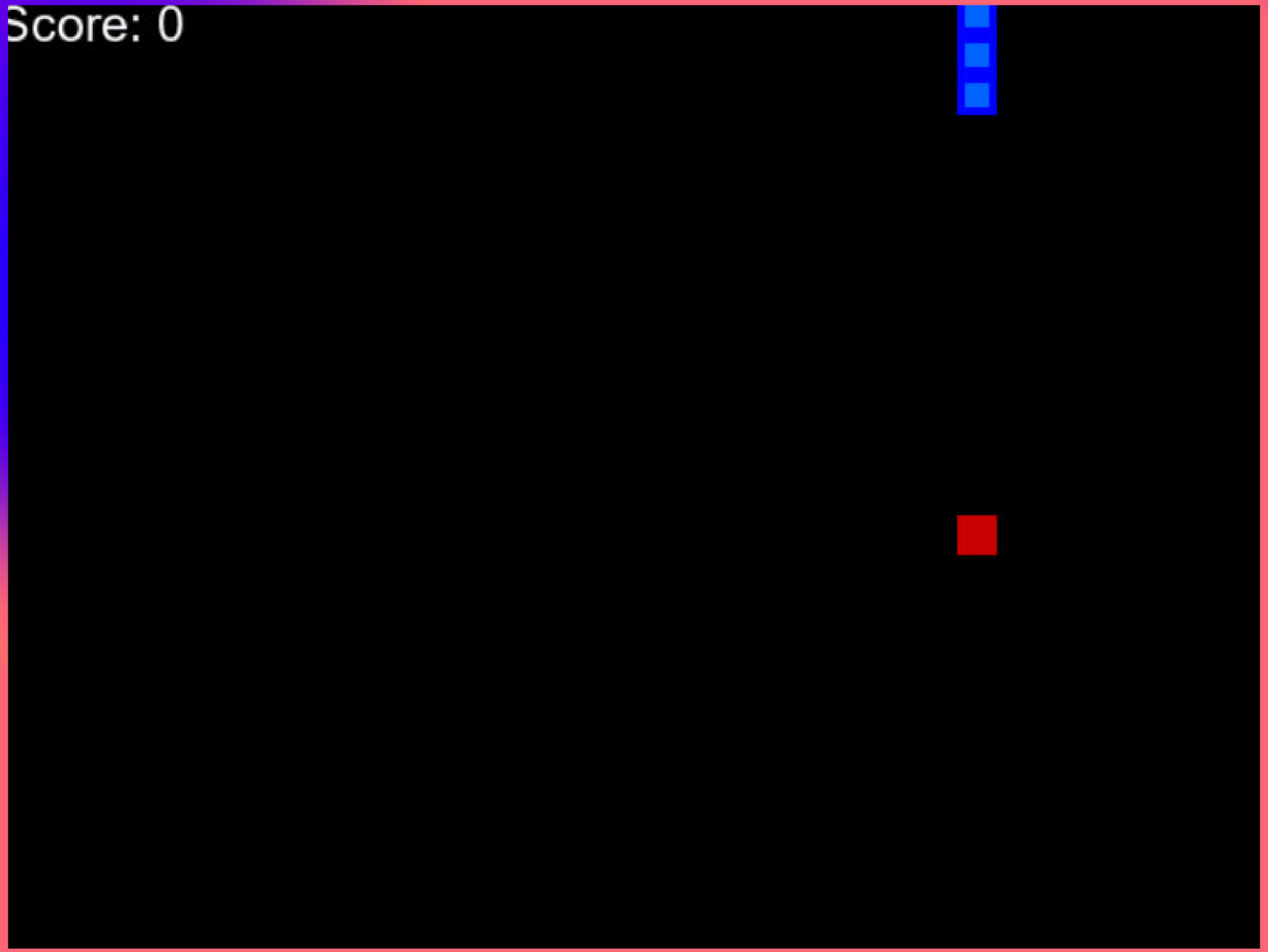


New estimate

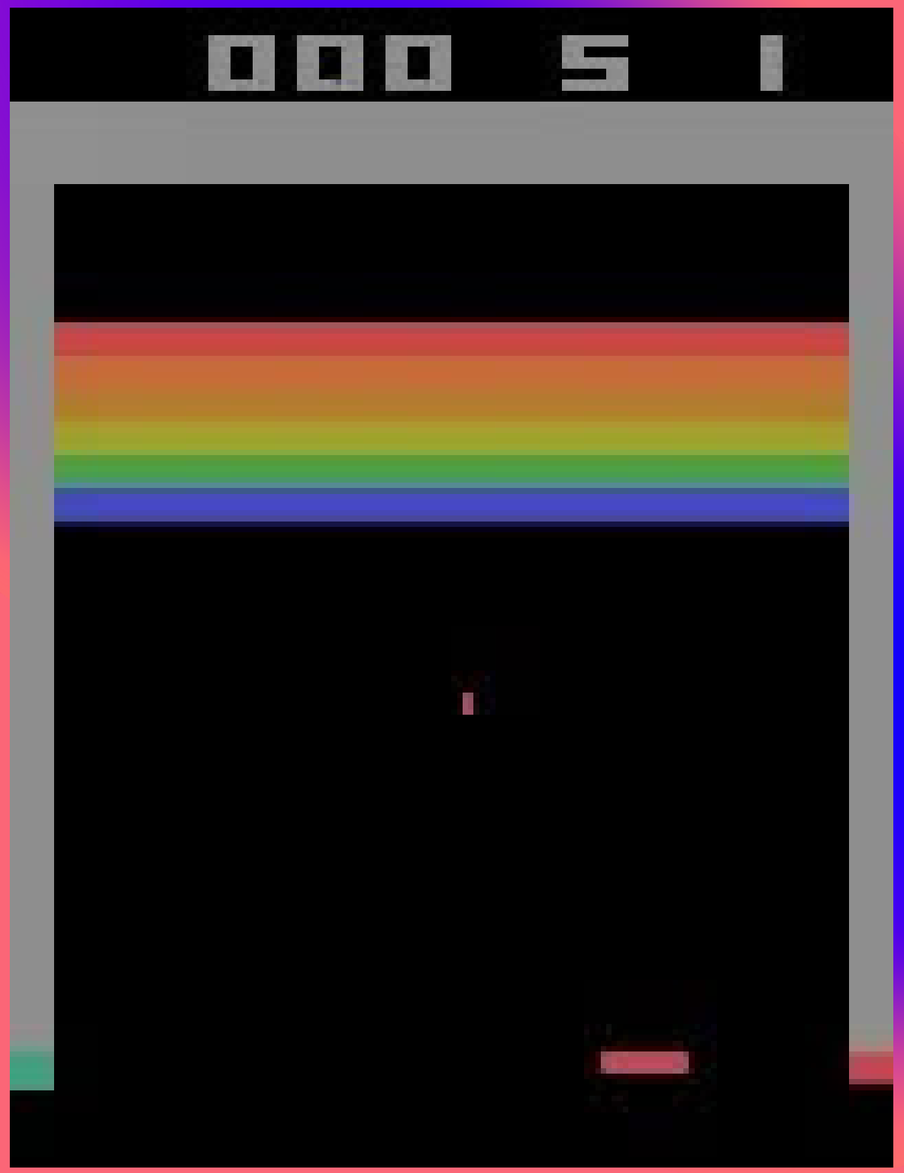
discount factor.

$$Q_{new}(s,a) = Q(s,a) + \alpha \left[r + \gamma \max_{a_i} Q(s', a_i) - Q(s,a) \right]$$

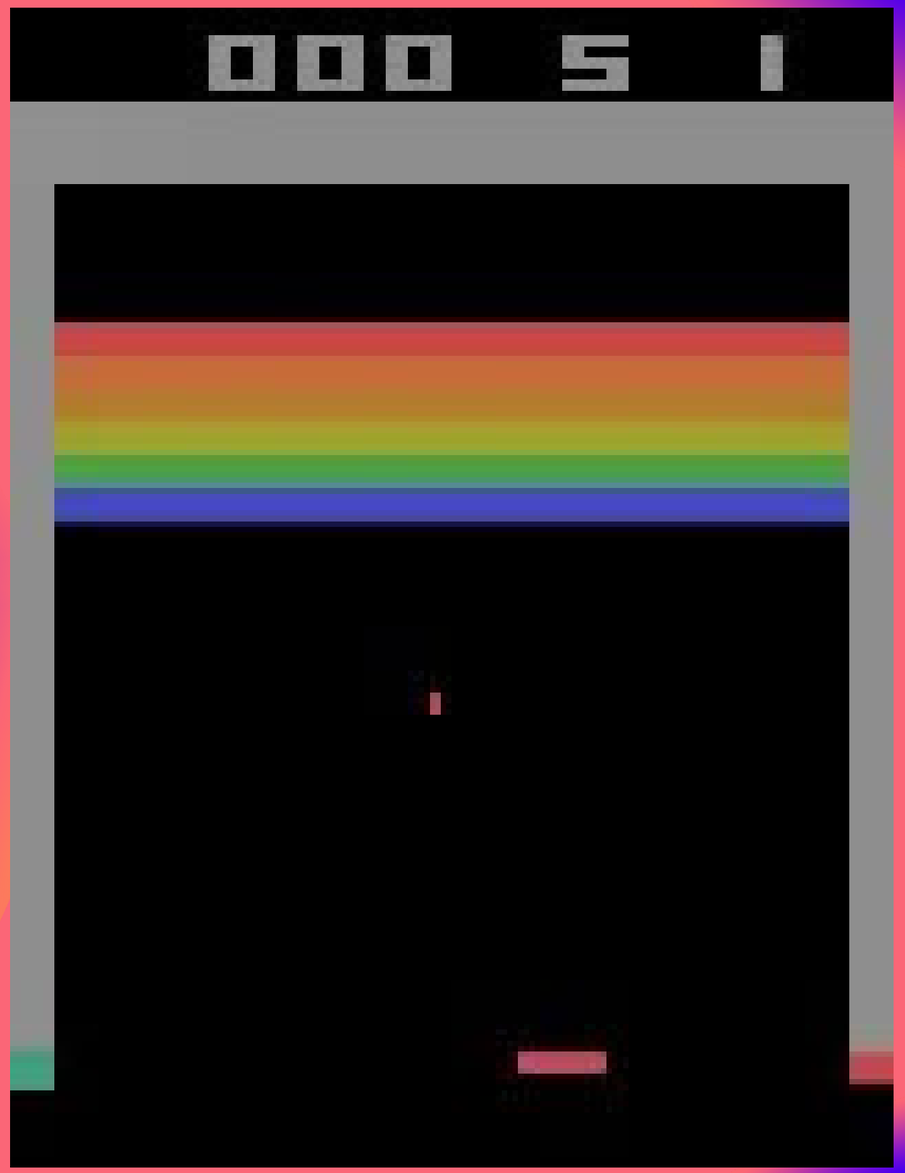
Learning rate



An AI driven Snake Game using Deep Q Learning



Initial Performance



After 30 minutes of training

The key distinguishing factor of reinforcement learning is how the agent is trained. Instead of inspecting the data provided, the model interacts with the environment, seeking ways to maximize the reward. In the case of deep reinforcement learning, a neural network is in charge of storing the experiences and thus improves the way the task is performed.

SUMMARY

Reinforcement learning is no doubt a cutting-edge technology that has the potential to transform our world. However, it need not be used in every case. Nevertheless, reinforcement learning seems to be the most likely way to make a machine creative - as seeking new, innovative ways to perform its tasks is in fact creativity.

Thank you!