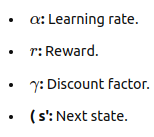
### **Deep Q-Network (DQN) in Reinforcement Learning**

### **1. Introduction to Reinforcement Learning**

* **Definition:** Reinforcement Learning (RL) is a type of machine learning where an **agent** learns to make decisions by interacting with an **environment**.
* **Goal:** Maximize cumulative reward by taking actions and receiving feedback in the form of rewards.
* **Key Concepts:**
  + **Agent:** The learner or decision-maker.
  + **Environment:** The world with which the agent interacts.
  + **State:** The current situation of the agent.
  + **Action:** What the agent can do.
  + **Reward:** The feedback from the environment.

### **2. The Q-Learning Algorithm**

* **Definition:** Q-learning is a value-based RL algorithm.
* **Q-Table:** Stores expected rewards (Q-values) for each state-action pair.
* **Update Rule:** 
* **Parameters:**



### **3. Limitations of Q-Learning**

* **Scalability Issue:** Q-table becomes impractical for large or high-dimensional state spaces.
* **Need for Function Approximation:** To handle complex environments, we need methods that approximate the Q-values efficiently.

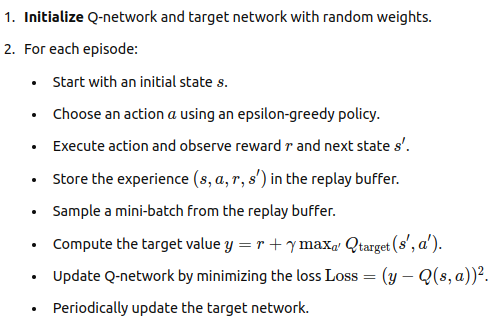
### **4. Deep Q-Network (DQN)**

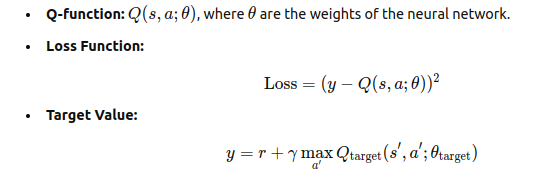
* **Definition:** DQN combines Q-learning with deep learning to approximate the Q-function using a neural network.
* **Key Advantage:** Can handle high-dimensional state spaces where Q-learning fails.

### **5. Key Components of DQN**

1. **Q-Function Approximation:** Uses a deep neural network to predict Q-values for state-action pairs.
2. **Experience Replay:** Stores experiences in a buffer and samples randomly during training to break the correlation between experiences.
3. **Target Network:** Uses two networks (online and target) to stabilize training by periodically updating the target network.

### **6. DQN Algorithm: Step-by-Step**



**7. Mathematical Formulation**

* **Optimization:** Gradient descent is used to minimize the loss and update the network weights.

### **8. Why DQN Works in Complex Environments**

* **Handles Large State Spaces:** Q-values are approximated using deep learning, making it possible to solve large-scale problems.
* **Experience Replay:** Ensures stability in learning by randomizing the training samples.
* **Target Network:** Provides more stable Q-value updates, preventing the instability common in Q-learning.

### **9. Conclusion**

* **Summary:** DQN is an effective approach to handle large, complex environments in reinforcement learning by combining deep learning with Q-learning.
* **Applications:** DQN is widely used in robotics, gaming, autonomous systems, and any environment requiring complex decision-making.