An Exploration on Q-Learning

Reinforcement Learning

- Type of machine learning where an agent learns to make decisions by interacting with an environment to achieve a specific goal

- Basic elements: Agent, Environment, Actions and Rewards

Q-Learning

- Q-learning is a widely used reinforcement learning algorithm.
  - It's an off-policy, model-free learning technique.

- Inputs
  - States
  - Actions
  - Rewards

- Output
  - Optimal Policy (Q-Table)

- Evaluation
  - Cumulative reward per episode
  - Cumulative steps per episode

How it works

- Initialization

- Taking Action and Updating Q-Values

- Repeat

Q-Value update equation

Q(s, a) = Q(s, a) + α * [R + γ * max(Q(s', a')) - Q(s, a)]

- **Q(s, a)** is the current Q-value for state s and action a.
- **α** is the learning rate, controlling the step size for updates.
- **R** is the immediate reward received for the action.
- **γ** is the discount factor, which accounts for the agent's preference for immediate rewards over delayed ones.
- **max(Q(s', a'))** represents the maximum Q-value for the next state s' and all possible actions a'.

A basic implementation of Q-Learning

Exploration Policies

- **Epsilon-Greedy (ε-Greedy)**:
  - The agent chooses the action with the highest Q-value with probability 1 - ε (exploitation).
  - It selects a random action with probability ε (exploration).
- Softmax Exploration:
  - The probability of selecting an action is determined by the Softmax function applied to the Q-values.
  - It allows for a more gradual exploration strategy compared to ε-Greedy.
- Upper Confidence Bound (UCB):
  - This policy selects actions that maximize an upper confidence bound on their estimated Q-values.
  - It balances exploration and exploitation by considering uncertainty in the Q-value estimates.
- Thompson Sampling:
  - It uses a Bayesian approach to maintain a probability distribution over Q-values for each action.
  - The agent samples from this distribution to select actions, favoring those with higher estimated rewards.

Challenges

- Difficult to measure success
  - Ended up using cumulative reward per episode and number of steps per episode.
- Volume of new information
  - Ended up finding a lot of resources
  - There is a free course about reinforcement learning on udemy which helped

Learnings

- Q-learning is a relatively intuitive introduction to Reinforcement Learning
- By itself it is a quite limited by needing to have a relatively small and discrete state-action space
- Identify potential success metrics before hand and document their estimated values

AusPost Scenario (if we have time)

Thank You