Artificial Intelligence
(Machine Learning: Reinforcement learning)

Prof K R Chowdhary

CSE Dept., MBM University

February 10, 2025
Lecture #5

Reinforcement Learning (RL)

- Human learning is through feedback of their actions in the real-world. We always learn by interactions with the teacher (environment), which is in the form of *cause* and *effect* relations.
- The environment or the world around us is like a teacher, but its lessons are often difficult to detect or grasp or analyze.
- The best example is learning by a dog where good actions are rewarded and bad actions are discouraged. RL has four components:
  - policy,
  - a reward function,
  - a value mapping, and
  - a model of environment.
- The "reward function" is a relationship between *state* and *goal*, it maps each state into a reward measure, and indicate the need of that action to achieve the goal.
- A RL system may not have a teacher to respond each action, the learner creates a policy to interpret feedback.

Reinforcement Learning...

- With the objective to maximize the expected reward, RL algorithms attempt to learn policies.
- *State-space* of real-world problems contains infinitely large number of possible states features. So, the designer of any such task must pickup only the most relevant features. Say, for task: "Travel to Mumbai for work, and the weather report for New Delhi is not likely to be relevant."
- From these we construct a feature set, and break it down into a number of subsets, so that each subset can learn specific concept of the domain. Some concepts/ feature-set subsets may be more important than others.

Example

Formulate a task of navigation to be carried out by an agent, with a goal to investigate best plan to go from point A to point B, and may choose a path and transport method of walking, driving, taxi,..

Reinforcement learning...

The feature-set of the agent's state-space may include,

- Positions of A and B,
- Raining (yes/no),
- Type of shoes of agent,
- Agent is with umbrella (Yes/no),
- Current time, and
- Day of week.

• Using these features, the agent can learn the concept of position and basic path planing.

• The features: raining, shoes, ..., are useful in learning as how the weather governs the policy. The features: time and day in a week may be useful in learning to handle traffic.

• A conventional approach to solve this problem through RL is to learn in a six dimensional space (positions, raining, shoes, umbrella, time, weekday) when all the features are taken into account.

• Think of six dimensions, vs. two/three dimensions!

Some functions in Reinforcement Learning

• In RL system, an agent recognizes itself in some state $p \in S$, then takes some action $a \in A$, and then recognizes itself in a new state q. The q is decided by the agent's *transition function* T, e.g., $T(p, a) \rightarrow q$, in general:

$$T(S \times A) \rightarrow S. \qquad (1)$$

• Also, the agent receives a reward r for arriving to q based on the *reward function* R:

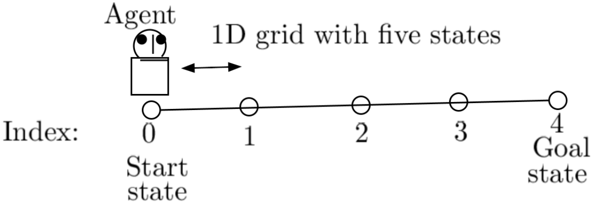
$$R(S \times A) \rightarrow \mathbb{R}. \qquad (2)$$

• A *value function* $V^{\pi}(p)$ is based on average sum-of-rewards received when an agent starts in p, and enter into q, following a policy π. Relation between *value function* and *optimal policy* $V^{*}(p)$ is:

$$\forall \pi, p : V^{*}(p) \geq V^{\pi}(p). \qquad (3)$$

• The RL used in many real-world domains, where discounted total reward is optimized.

Q-Learning Example: 1D Grid World



RL Example: The agent is placed in a 1-dimensional grid with a starting point at index 0, and the goal is at the last index.

- The agent can move either left or right at each step, and it tries to maximize the cumulative reward. Q-value (quality of actions): $Q(s, a)$, is the expected cumulative reward of taking action a in state s.

- Grid: A 1D grid with 5 states (0 to 4). Agent starts at state 0 and aims to reach state 4 (goal).

- Actions: The agent can take two actions: 0: Move left (decrease the state index), 1: Move right (increase the state index).

Q-Learning Example: 1D Grid World ..

- Q-learning: The agent uses the Q-learning algorithm to update its knowledge about the environment.

- Rule: $Q(s,a) = Q(s,a) + \alpha[r(s') + \gamma.\ max_{a'}\ Q(s',a') - Q(s,a)]$, where $Q(s,a)$ is current estimate of Q-value, α is learning rate, r is immediate reward, γ is discount factor.

command: $ python3 reinf.py

- Rewards (r): The agent receives: +10 for reaching the goal, -1 for each step (penalty to encourage the agent to reach the goal quickly).

Hyperparameters:

- learning_rate (α): Determines how much new information overrides old Q-values.

- discount_factor (γ)): How much the agent values future rewards.

- Exploration rate(epsilon): Probability of taking a random action (exploration vs. exploitation).

- Episodes: Number of training cycles or iterations (1000).

- Max. steps(actions) agent will take per episode.

Q-Learning Example: 1D Grid World ...

Output:

- Training Progress: The program prints the total reward every 100 episodes during training.
Testing: After training, the agent tests its learned behavior to see if it successfully reaches the goal.

Key Points:

- This example demonstrates a very simple Q-learning setup.

- The environment is 1D, and the agent's task is to learn to move towards the goal state.

- The reward structure encourages the agent to minimize steps to reach the goal.

- This example has demonstrated fundamentals of reinforcement learning with minimal complexity!

References

[1] Chowdhary, K.R. (2020). Statistical Learning Theory. In: Fundamentals of Artificial Intelligence. Springer, New Delhi. https://doi.org/10.1007/978-81-322-3972-7_14