Reinforcement Learning (DQN)

Pierre Squarra

Cognitive Algorithms Seminar

Technische Universität Berlin

Outline

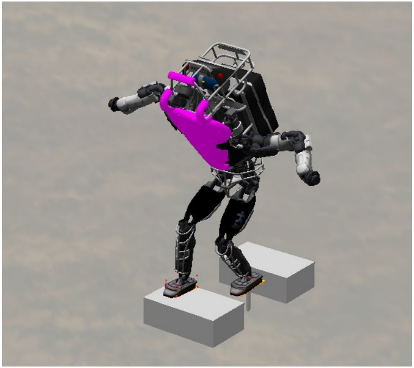


Figure 1: Walking Robot [1]

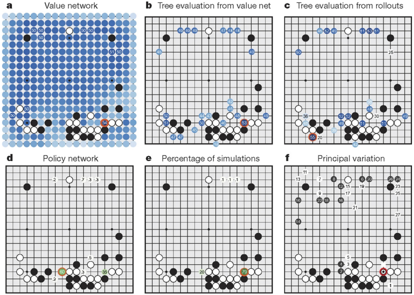


Figure 2: AlphaGo [2]

Outline

Agent-Environment Interface

Agent: Decision-maker taking actions

Environment: World which the agent interacts with

Figure 3: The agent-environment interface [3]

Agent-Environment Interface

Action: Move the agent can make in the environment

Observation: Agent's perception of the environment

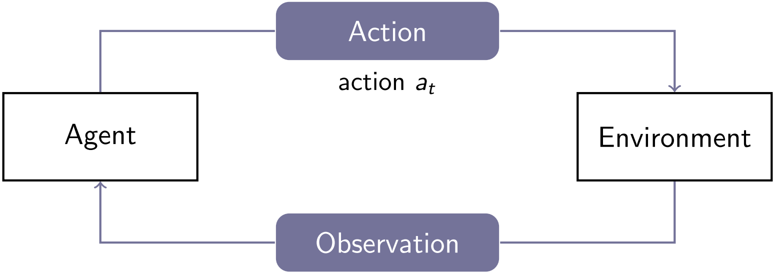


Figure 3: The agent-environment interface [3]

Agent-Environment Interface

State: Current situation or configuration of the environment

Reward: Scalar value given to an agent as feedback for its actions

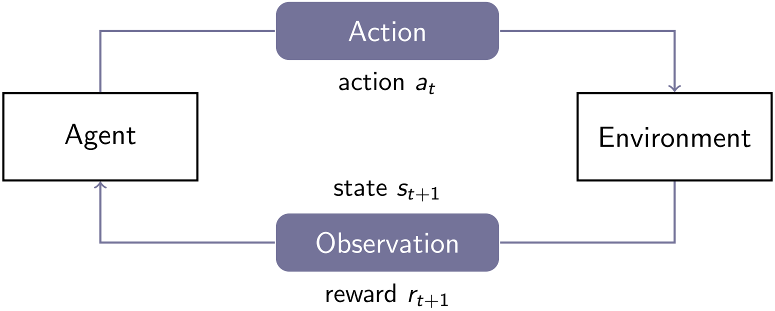


Figure 3: The agent-environment interface [3]

Returns and Value Functions

- **Goal:** maximize cumulative reward, called the return [4]

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

Returns and Value Functions

- **Goal:** maximize cumulative reward, called the discounted return [4]

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- γ: discount factor; $0 \leq \gamma \leq 1$

Returns and Value Functions

- **Goal:** maximize cumulative reward, called the discounted return [4]

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- γ: discount factor; $0 \leq \gamma \leq 1$

State-Value function [4]

$$v(s) = \mathbb{E}[G_t \mid S_t = s]$$

How good it is for the agent to be in a given state

Action-Value function [4]

$$q(s,a) = \mathbb{E}[G_t \mid S_t = s, A_t = a]$$

How good it is to perform a certain action in a given state

- A policy is a strategy that the agent follows to decide actions based on the current state $\pi(s) \rightarrow a$

Outline

Q-Learning

- Q-Learning uses a Q-table with $S \times A$ entries to store the expected rewards for state-action pairs

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \underbrace{\left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]}_{\text{Bellman error}} [5]$$

- The Bellman error is the difference between the current estimate of the Q-value for a state-action pair and the "true" Q-value
- The learning rate α determines how quickly the agent learns from its experiences

Limitations of standard RL

- Tabular methods are impractical

  Problem: Curse of dimensionality
- Generalization across states

  Problem: Fail to generalize across similar states
- Continuous state and action spaces

  Problem: Q-Leaning designed for continuous domains

Limitations of standard RL

- Tabular methods are impractical

  Problem: Curse of dimensionality
- Generalization across states

  Problem: Fail to generalize across similar states
- Continuous state and action spaces

  Problem: Q-Leaning designed for continuous domains

Solution

Use a function approximator to estimate $Q(S, A)$

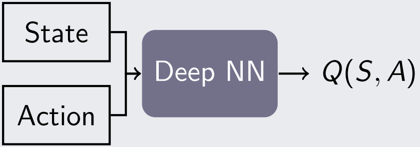
$\rightarrow$ Deep Neural Network

Outline

Deep Q Networks

- Extension of Q-Learning → Estimate the optimal Q-function

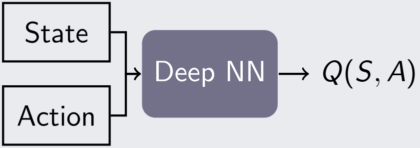
Action + State → Expected Return [3]



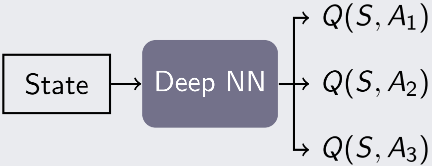
Deep Q Networks

- Extension of Q-Learning $\rightarrow$ Estimate the optimal Q-function

Action + State $\rightarrow$ Expected Return [3]



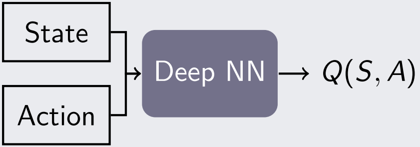
State $\rightarrow$ Expected Return for each Action [3]



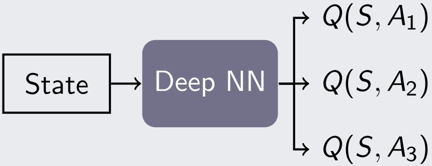
Deep Q Networks

- Extension of Q-Learning → Estimate the optimal Q-function

Action + State → Expected Return [3]



State → Expected Return for each Action [3]



$$\mathcal{L}(\theta) = \left|\left| \underbrace{\left(R_{t+1} + \gamma \max_a Q(S_{t+1}, a)\right)}_{\text{target}} - \underbrace{Q(S_t, A_t)}_{\text{predicted}} \right|\right|^2 \quad [3]$$

Challenges in Deep Q Networks

- **Target Network**

  Problem: Using the same network for selecting and evaluating actions makes it hard for the model to stabilize and converge

  Solution: Use a target network to evaluate actions

- **Experience Replay**

  Problem: Correlation of states lead to inefficiencies and instability

  Solution: Randomly sample from a memory buffer to break correlation

References I

[1] G. Wiedebach, S. Bertrand, T. Wu, L. Fiorio, S. McCrory, R. Griffin, F. Nori, and J. Pratt, "Walking on Partial Footholds Including Line Contacts with the Humanoid Robot Atlas," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 1312–1319, Nov. 2016.
arXiv:1607.08089 [cs].

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, Jan. 2016.
Number: 7587 Publisher: Nature Publishing Group.

[3] A. Amini, "Deep Reinforcement Learning," Jan. 2023.

References II

[4] R. S. Sutton and A. Barto, *Reinforcement learning: an introduction*. Adaptive computation and machine learning, Cambridge, Massachusetts London, England: The MIT Press, second edition ed., 2020.

[5] R. Grosse and J. Ba, "Q-Learning," Apr. 2019.