Lab 5

Value-Based Reinforcement Learning

Alison Wen, Wei Hung

Contents

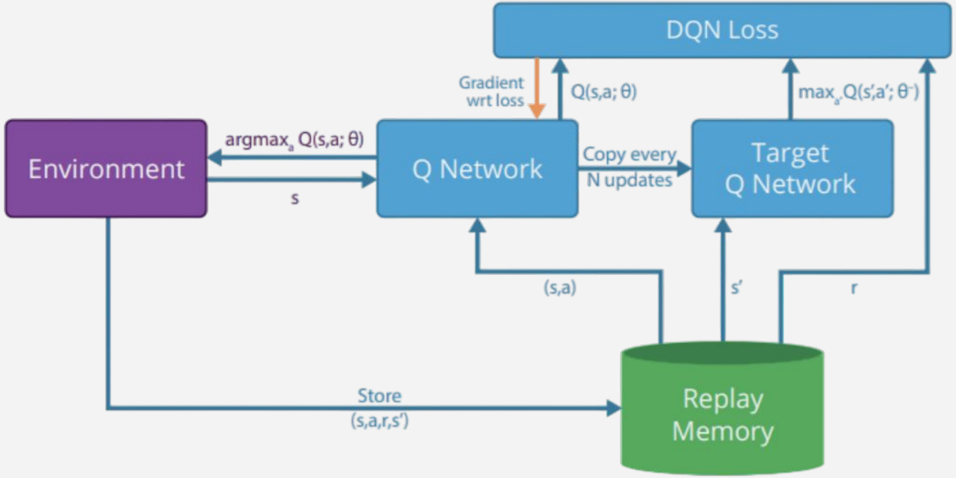
Background	Vanilla DQN, Double DQN, Prioritized experience Replay, Multi-Step Return
Lab Description	Task 1: Vanilla DQN Task 2: Vanilla DQN on Atari Task 3: Enhanced DQN
Model & Packages	Classes and required packages
Grading Policy	Report + Code + Video
Submission Policy	There will be penalty using the wrong file names!!

Background

Value-Based RL

Vanilla DQN

$$L_{\mathrm{DQN}}(\theta) := \frac{1}{2} \sum_{(s,a,r,s') \in D} \left(r + \gamma \max_{a' \in \mathcal{A}} Q(s', a'; \bar{\theta}) - Q(s, a; \theta) \right)^2$$



- **Double DQN (DDQN):**

$$L_{\text{DDQN}}(\theta) := \frac{1}{2} \sum_{(s,a,r,s')\sim D} \left(r + \gamma Q\left(s', \arg\max_{a' \in A} Q(s,a;\theta); \bar{\theta}\right) - Q(s,a;\theta) \right)^2$$

- **Prioritized experience Replay**
  - Priority: $p_i = |\delta_i| + \epsilon$

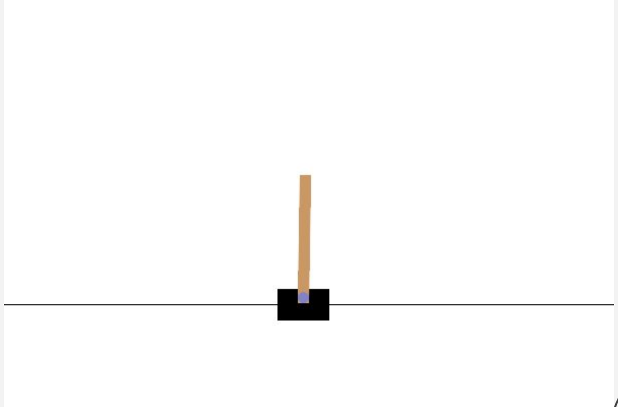
    where $\delta_i = r_i + \gamma \max_{a'} Q(s_i', a') - Q(s_i, a_i)$
  - Sampling Transition Probability: $P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}$
  - Importance Sampling Weight: $w_i = \left(\frac{1}{N \cdot P(i)} \right)^\beta$
- **Multi-Step Return**

$$R_t^{(n)} = \sum_{k=0}^{n-1} \gamma^k r_{t+k} + \gamma^n \max_{a'} Q(s_{t+n}, a')$$

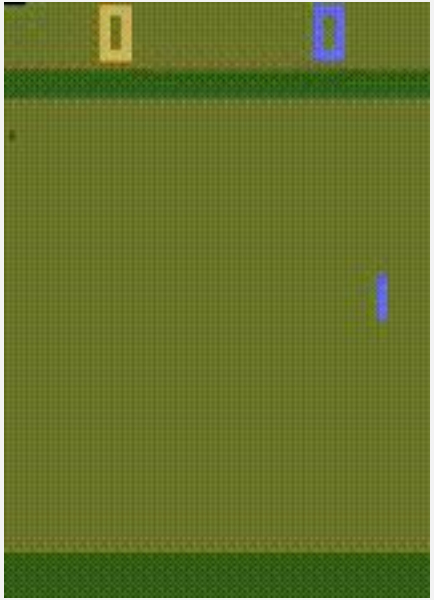
Lab Description

Task 1: DQN on CartPole

- Goal: The pole on the cart stays still
- State: v_{cart}, a_{cart}, $\theta_{pole \rightarrow cart}$, a_{pole}
- Action: Push left or right
- Reward:
  - Die: 0
  - Alive: 1
- Q-function approximate
- Experience Replay: Uniform sampling & target tentwork
- Logging and evaluation



Task 2: Vanilla DQN with Visual Observations on Atari



Goal:

Defeat the opponent by bouncing the ball past them.

Observation Space:

210 × 160 RGB image

Action Space:

0: NOOP 1: FIRE 2: RIGHT

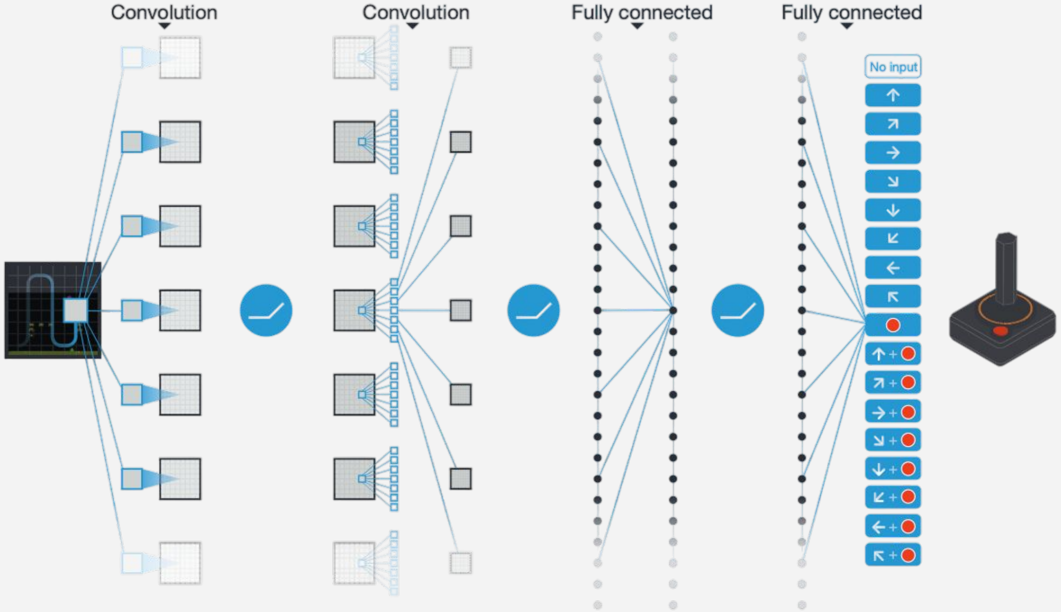
3: LEFT 4: RIGHTFIRE 5: LEFTFIRE

Reward:

+1: When the agent scores

-1: When the opponent scores

Task 2: Vanilla DQN with Visual Observations on Atari



Source: Human-level control through deep reinforcement learning, Minh et. al., Nature Vol. 518, Pg. 530

Task 2: Vanilla DQN with Visual Observations on Atari

- Preprocess the input frames (grayscale, resize, and stack frames)
- Use a convolutional neural network (CNN) as the Q-function approximator
- Evaluate and plot the total episodic rewards versus environment steps

Task 3: Enhanced DQN

Goal: Improve the learning efficiency of your DQN agent by incorporating the following enhancements:

- Double DQN
- Prioritized experience Replay (PER)
- Multi-Step Return

Task 3: Enhanced DQN

Requirements:

- Integrate the enhancements into your DQN code
- Justify the integration choices.
- Compare training performance against vanilla DQN using the Pong-v5 environment

Grading Policy

Report

- Introduction (5%): Please provide a high-level introduction to your report. You can mention the most important findings and the overall organization of this report.
- Your implementation (20%): Please briefly explain your implementation for Tasks 1-3. Specifically, please describe:
  - How do you obtain the Bellman error for DQN?
  - How do you modify DQN to Double DQN?
  - How do you implement the memory buffer for PER?
  - How do you modify the 1-step return to multi-step return?
  - explain how you use Weight & Bias to track the model performance

Report

- Analysis and discussions (25%)
  - Plot the training curves.
  - Analyze the sample efficiency with and without the DQN enhancements. If possible, perform an ablation study on each technique separately (15%).
  - Additional analysis on other training strategies (Bonus up to 10%).

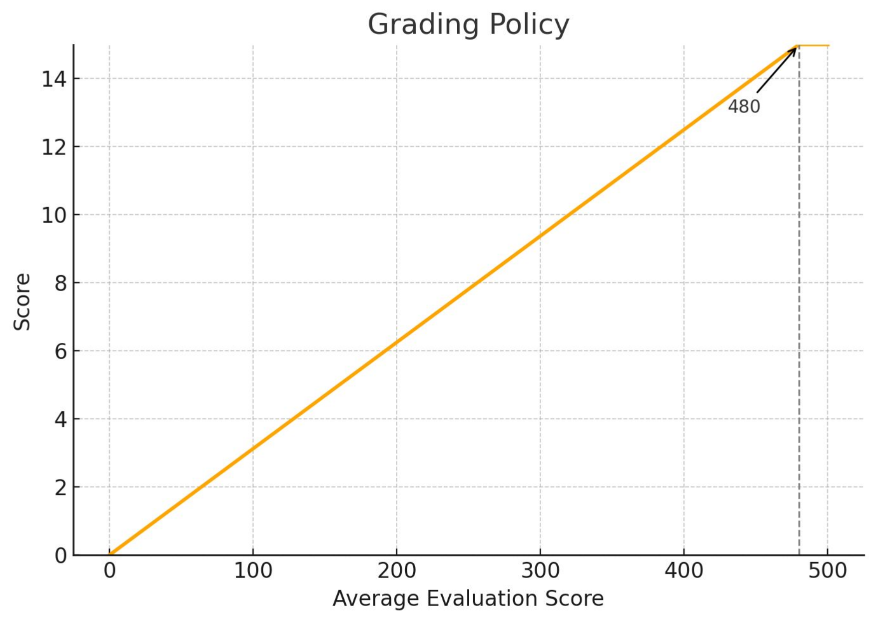
Demo Video

- Total Duration: 5–6 minutes
- Language: English (unless pre-approved by TAs)
  - 🔹 Source Code (~2 minutes): Describe your implementation
  - 🔹 Model Performance (~3 minutes): Demonstrate your obtained models

⚠️ Model snapshots will NOT be graded if no valid demo video is provided.

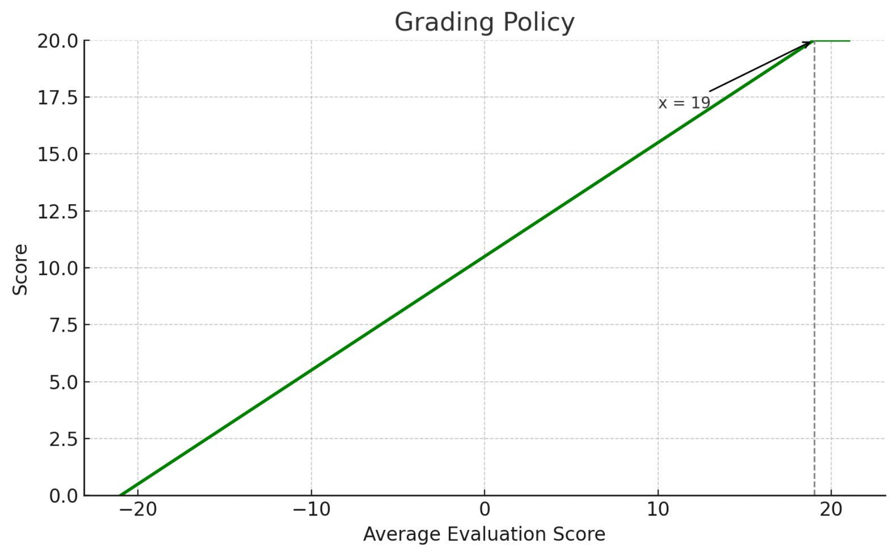
Model Snapshots – Task 1 (15%)

- **The grading of Task 1 would depend on the evaluation score of your submitted snapshot.**
- **Please use the best snapshot that you have obtained during the training process.**



Model Snapshots - Task 2 (20%)

- **The grading of Task 2 would depend on the evaluation score of your submitted snapshot.**
- **Please use the best snapshot that you have obtained during the training process.**



Model Snapshots - Task 3 (15%)

- **The grading of Task 3 would depend on the sample efficiency of your enhanced DQN.**
- **Please submit 5 model snapshots that are trained for 400k, 800k, 1.2M, 1.6M, and 2M environment steps.**

Submission Policy

- **Please strictly follow the naming policy and zip all your deliverables into a folder !!!**

Directory Structure

```
LAB5_StudentID.zip
|-- LAB5_StudentID_Code/          <- Source code folder
|    |-- (any other .py files)    <- Your code files
|    |-- requirements.txt
|    |-- (any other .sh files)    <- Optional
|-- LAB5_StudentID.pdf            <- Technical report (single PDF)
|-- LAB5_StudentID.mp4            <- Demo video (5 - 6 minutes)
|-- LAB5_StudentID_task1.pt       <- Task 1 model snapshot
|-- LAB5_StudentID_task2.pt           <- Task 2 model snapshot
|-- LAB5_StudentID_task3_400000.pt    <- Task 3 snapshot (step = 400k)
|-- LAB5_StudentID_task3_800000.pt    <- Task 3 snapshot (step = 800k)
|-- ...
|-- LAB5_StudentID_task3_2000000.pt   <- Task 3 snapshot (step = 2M)
|-- LAB5_StudentID_task3_best.pt   <- Task 3 snapshot (any step reach score 19)
```

Submission Policy

- **You Must**
  - Include screenshots of your “evaluation results” in your report
  - Include commands to reproduce your results in your report
  - Ensure the results in your report are reproducible!!

Thanks for Your Attention