Deep Reinforcement Learning for Computer Games

Sean Yin, Chung-En Yu

Dec 13, 2022

UNIVERSITY OF MINNESOTA
Driven to Discover®

Content

- Project Motivation/Goal
- Approach/Solution
  - DQN
  - A2C
- Results/Demo
- Q&A

Motivation / Goal

Motivation

Neural Networks + Reinforcement learning → DQN, A2C

The first deep reinforcement learning is to play Atari game.

Playing Atari with Deep Reinforcement Learning, Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller

Goal

Implement the code from scratch : build and train DNN based on Atari game

Deep Reinforcement Learning

On-Policy	Off-Policy
Policy Gradient, Actor Critic (AC)	DQN, Double DQN Proximal Policy Gradient

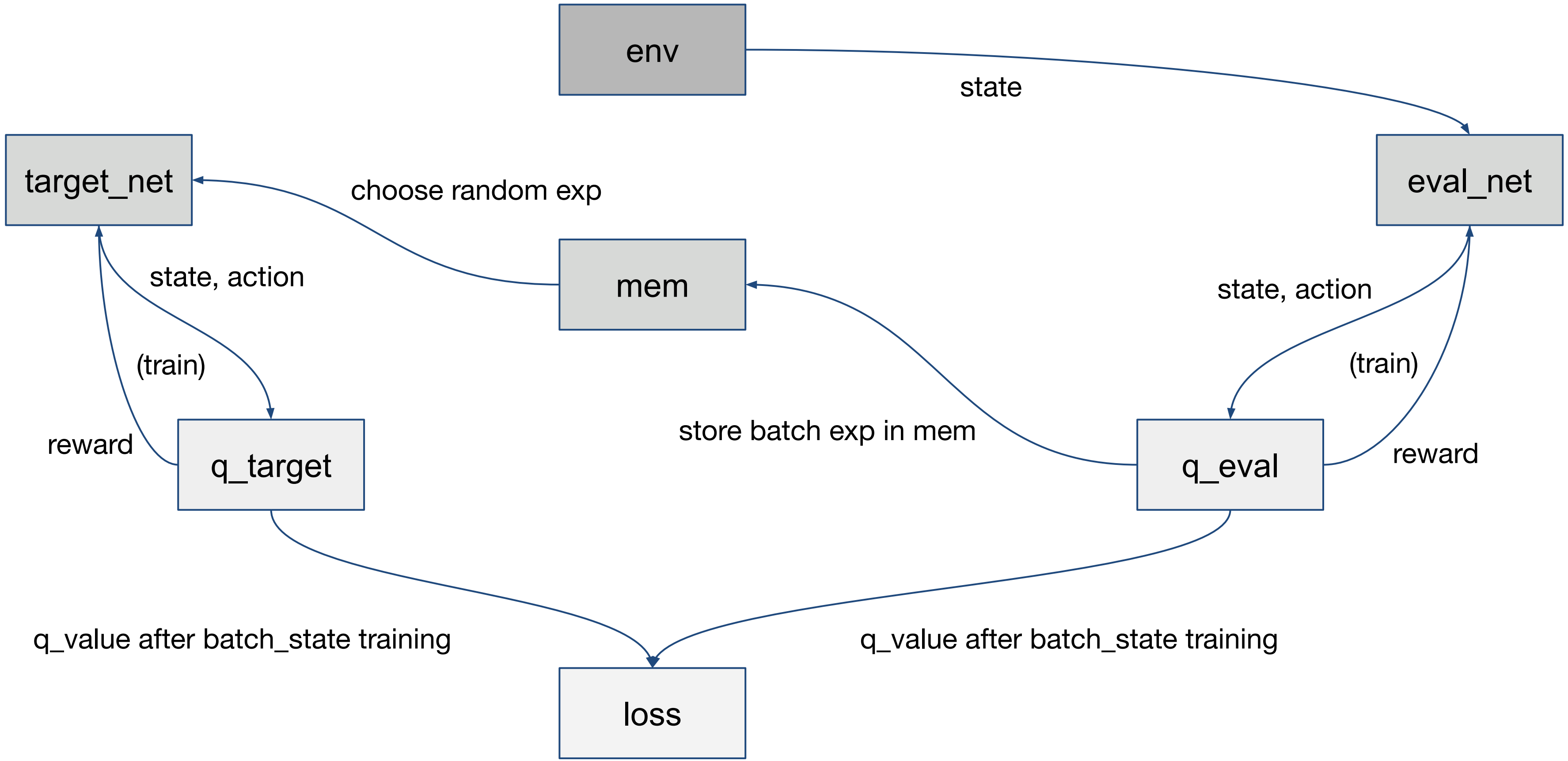
Off-policy: can train NN based on data collected from different policy.

Double DQN

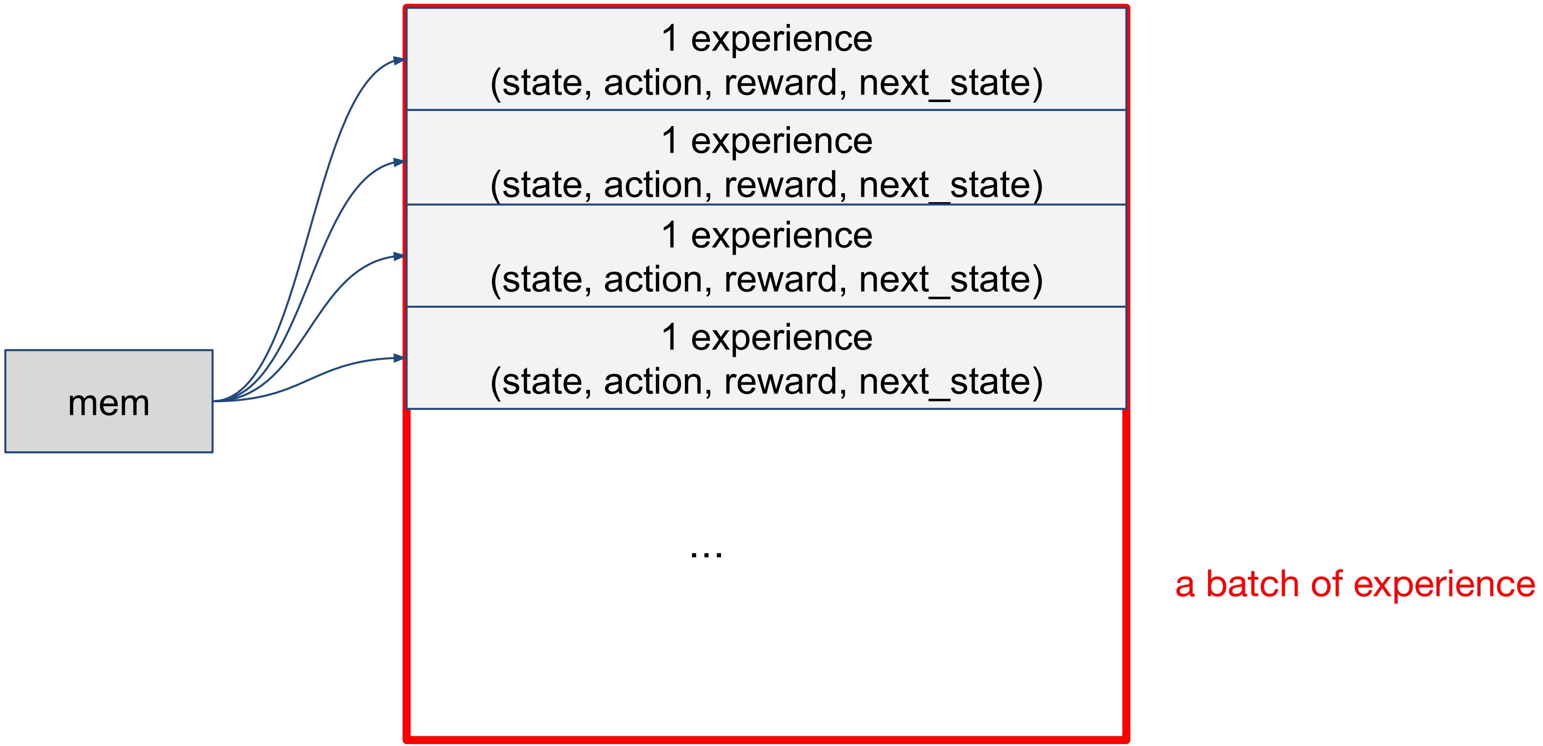
Approach/Solution

Pseudo Code : DQN Algorithm

```
Create 2 DQN networks : [1] target_net [2] eval_net
Episode loop:
    Loop:
        - Extract image features ( obs_space, act_space )
        - Choose an action from the current state
        - Do the action, and observe the next state and reward
        - Store the experience(observation) to the memory bank
        - Accumulate the reward
        - If the memory bank is full :
            = Target_net learns(trains) from the memory
            = End loop, start a new episode
```



* **exp** includes: state, action, reward



* **experience** = ‘transition’ variable in the code
1 experience is a 1D array

AC

Actor Critic

- Critic:

  Value Function served as a baseline to reduce the variance.
- Shared Common features map between actor and critic to reduce the bias of the critic.

```
PolicyNetwork(
  (features): Sequential(
    (0): Conv2d(3, 32, kernel_size=(5, 5), stride=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=3, stride=3, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(32, 16, kernel_size=(3, 3), stride=(1, 1))
    (4): ReLU()
    (5): MaxPool2d(kernel_size=3, stride=3, padding=0, dilation=1, ceil_mode=False)
    (6): ReLU()
    (7): Flatten(start_dim=1, end_dim=-1)
    (8): Linear(in_features=11664, out_features=128, bias=True)
    (9): ReLU()
    (10): Linear(in_features=128, out_features=64, bias=True)
    (11): ReLU()
  )
  (actor): Linear(in_features=64, out_features=2, bias=True)
  (critic): Linear(in_features=64, out_features=1, bias=True)
)
```

Simulation Setups

Atari - Assault v4

0: NOOP
1: FIRE,
2: UP
3: RIGHT
4: LEFT
5: RIGHTFIRE
6: LEFTFIRE

State	Image (210 x 160) x RGB (3)
Action	Discrete (7)
Reward	Score

Results

Results - Double DQN vs. A2C

Double-DQN

AC

Resources:
RTX 3080 10 G

Conclusion

Conclusion

- **Project novelty and difficulty:**
  We built and trained two of the most popular deep reinforcement learning algorithm from scratch.

- **Demo:**
  Two type of agents are trained to learn the optimal policy.

Q&A

UNIVERSITY OF MINNESOTA

Driven to Discover®

Crookston Duluth Morris Rochester Twin Cities