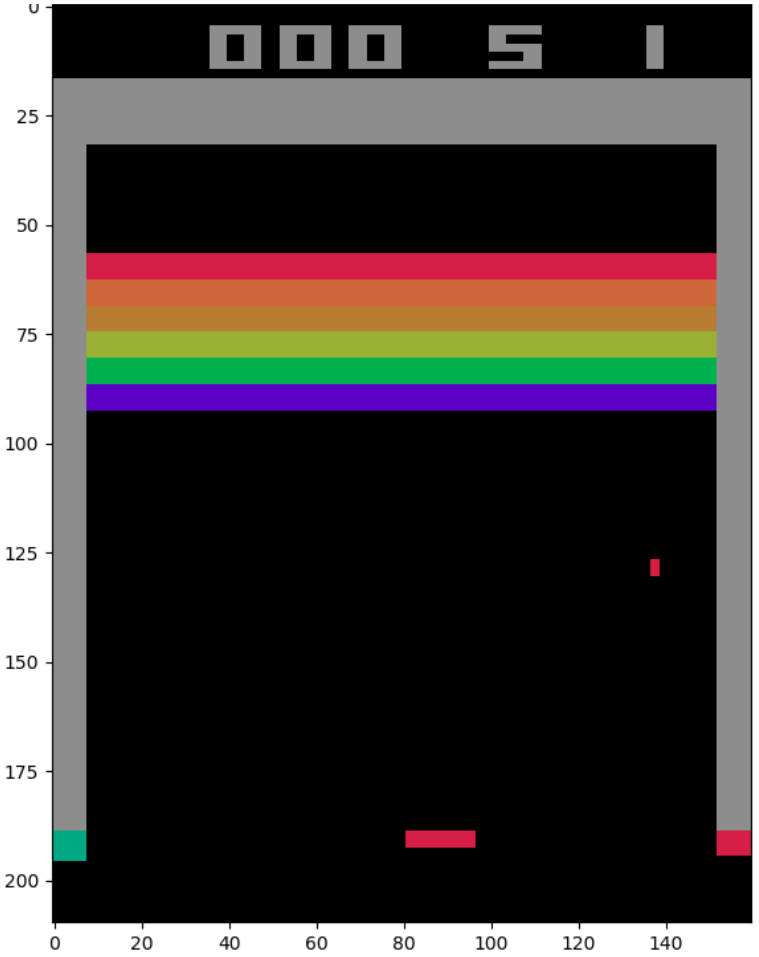
DQN homework.

Atari Breakout

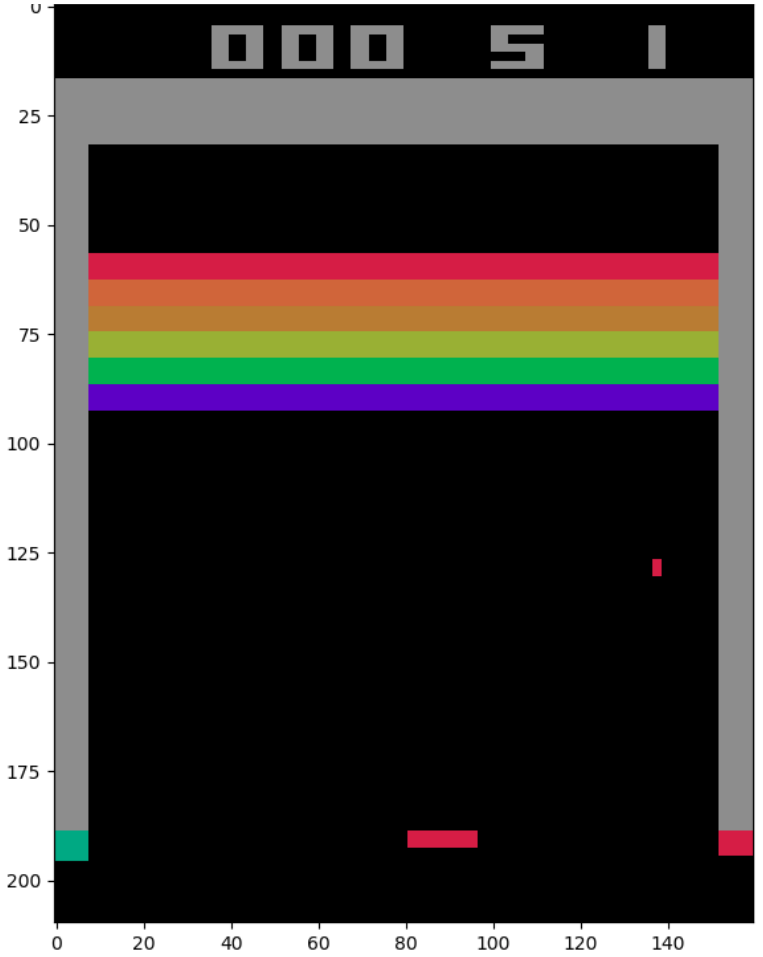


- Observation: rgb image, 210x160x3
- 4 Actions:

  Fire, Left, Right, Do Nothing
- Reward: Break the wall

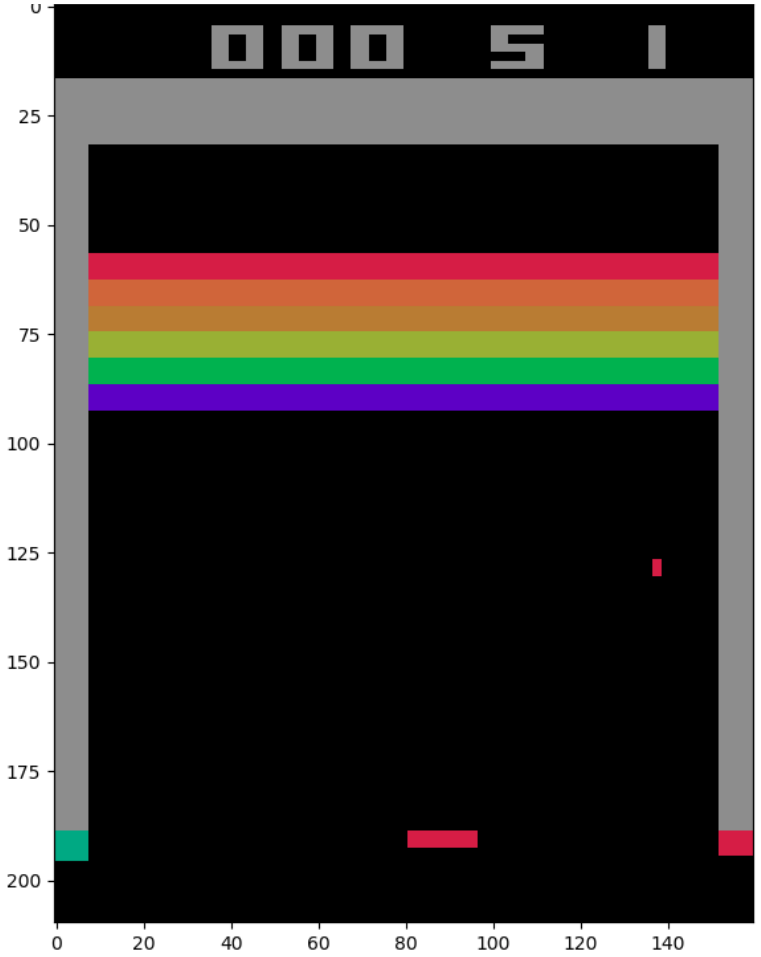
Let’s play!

If you were an agent



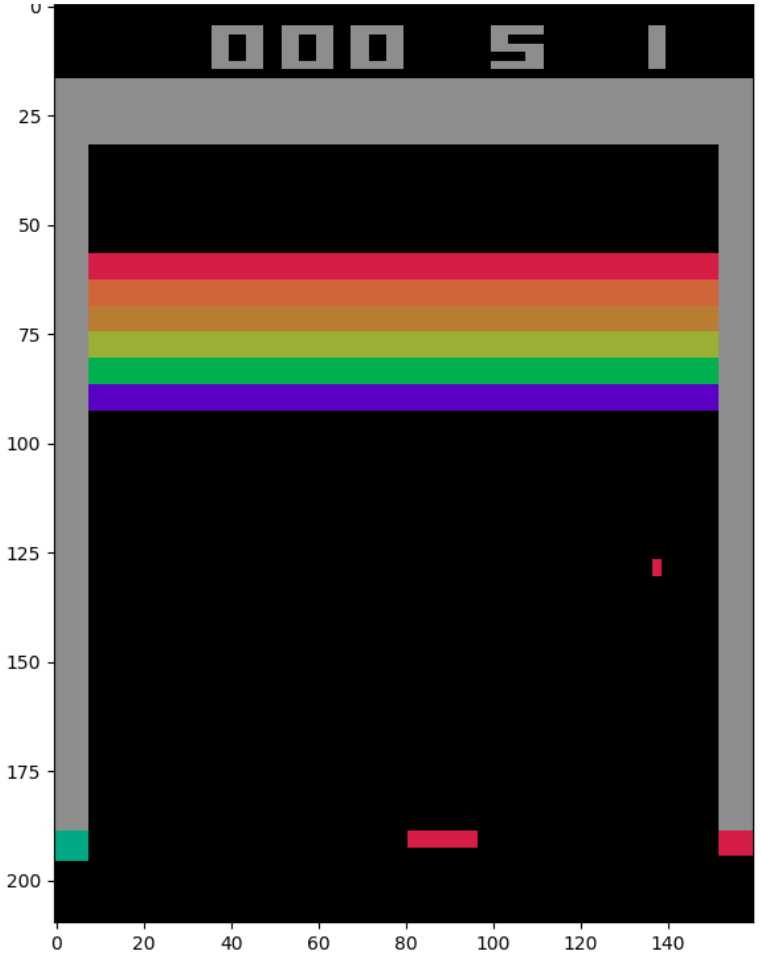
- Is one observation enough?

If you were an agent



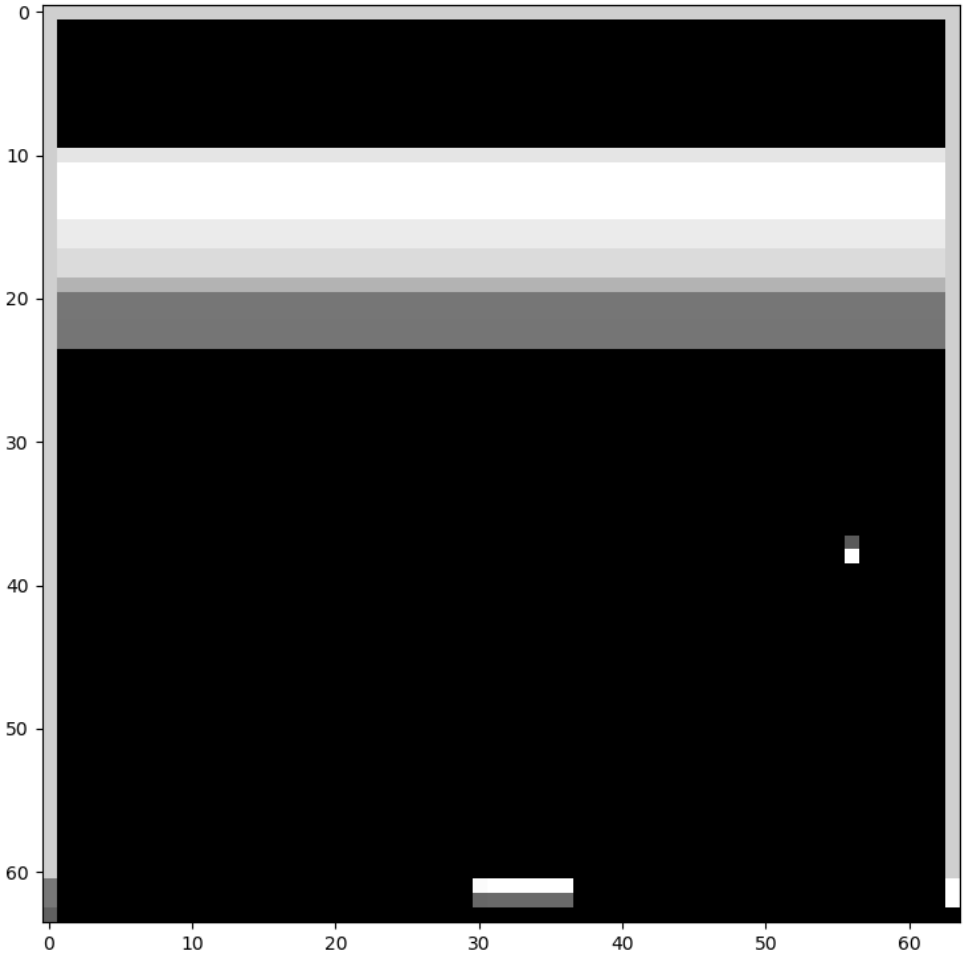
- Is one observation enough?
- How should Q-leaning agent (deep or not) understand that losing the ball is bad if there is no negative reward for that?

If you were an agent



- Is one observation enough?
- How should Q-leaning agent (deep or not) understand that losing the ball is bad if there is no negative reward for that?
- Does time step matter?

Wrapping

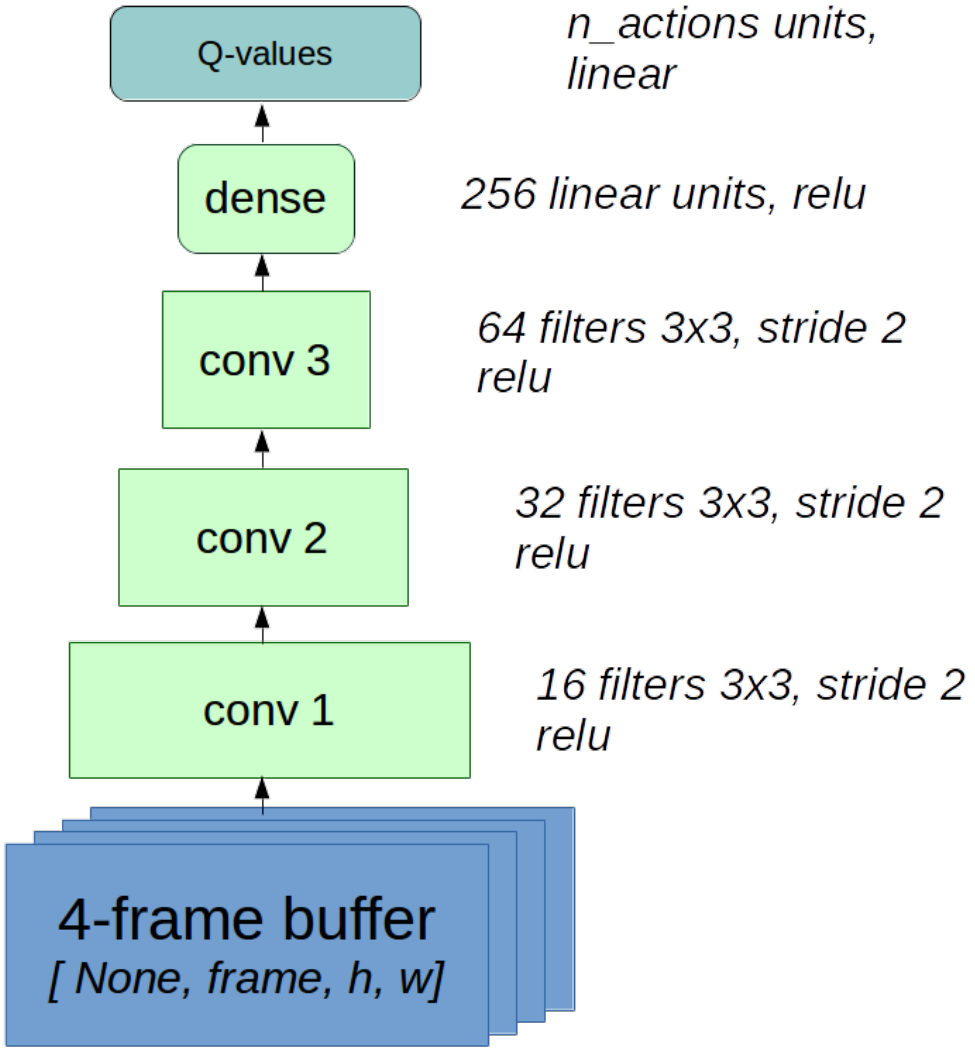


- 4-Frame Buffer
- Observation wrapper (No Python cycles!)
- MaxAndSkipEnv
- EpisodicLifeEnv
- FireResetEnv
- ClipRewardEnv

  (Note! The order of wrappers is not optimal here)

Let’s play again!

Network Architecture

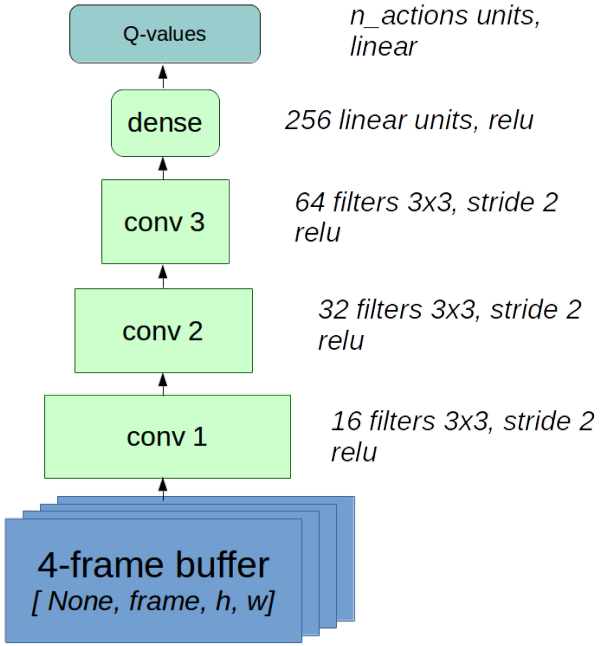


- Stride instead of Max Pool
- No BatchNorm
- No Dropout

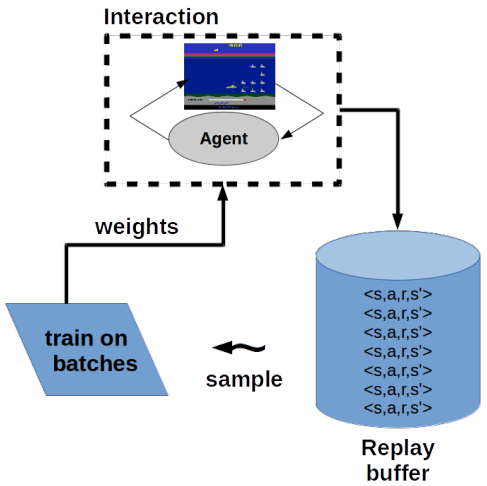
You may try the architecture from the original paper:

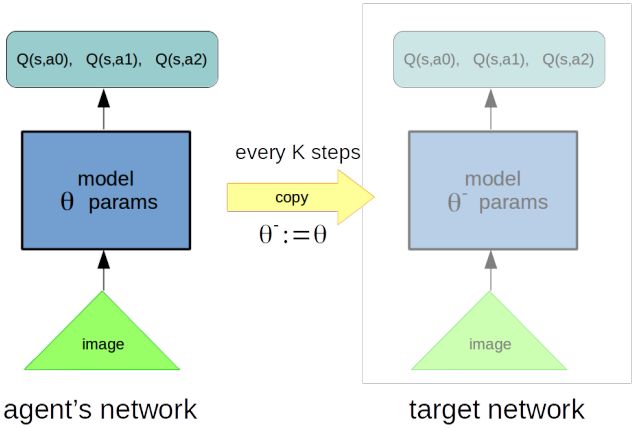
https://arxiv.org/pdf/1312.5602.pdf

Keep in mind

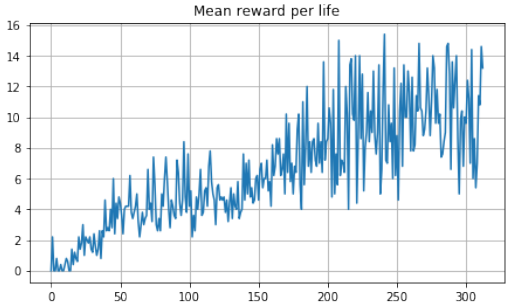


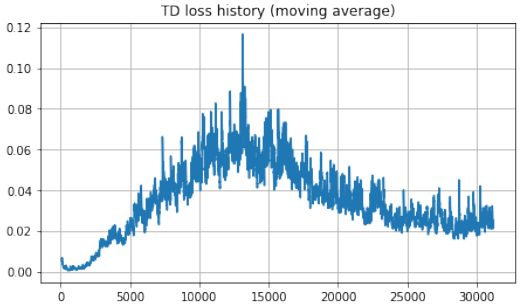
- Network
- Replay Buffer
- Target Network

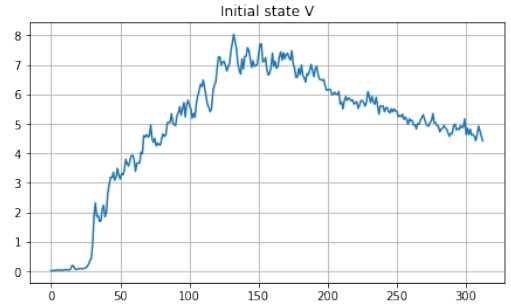


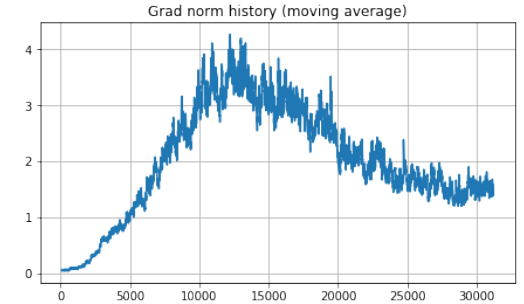


Main Loop









Hyperparameters:

- Linear schedule
- Small batch
- Slow refresh of target network (why?)
- Some Patience

  (about 8 hours of GPU-patience at the plots)

Policy must not be trivial!

Visualization

This sample is really good. Don’t panic if your plot looks worse. Mine did :)

Bonus Tasks

These are simple

- Some more interpretation
- Double DQN
- Dueling DQN

These are not

- Optimized and Prioritized Buffer
- Distributed RL

Pipeline, Scoring and Deadline

2 weeks:

- Implement Seminar notebook (5 pts)
- Implement Debug notebook

  (probably you will want to send it)
- Implement Main Notebook (10.5 pts)

+1 week

- Main loop (4.5 pts)
- Bonus tasks

Scoring and deadline are approximate. Deadline in Anytask has priority.