Reinforcement Learning: Deep Q Networks

DTU Compute

Authors: Ștefan Bîrs(s183047), Tiberiu-Ioan Szatmari(s183050), Kamran Thomas Alimagham(s182856), Mihai Nipomici(s184432)

1. Introduction

What: We used reinforcement learning methods to create an AI agent that out-performs the hard-coded agent in the classic Atari game, Pong.

How: The proposed solution: sample the Pong environment, feed raw game frames to a deep Q network (DQN), then take actions based on its output.

2. Computation:

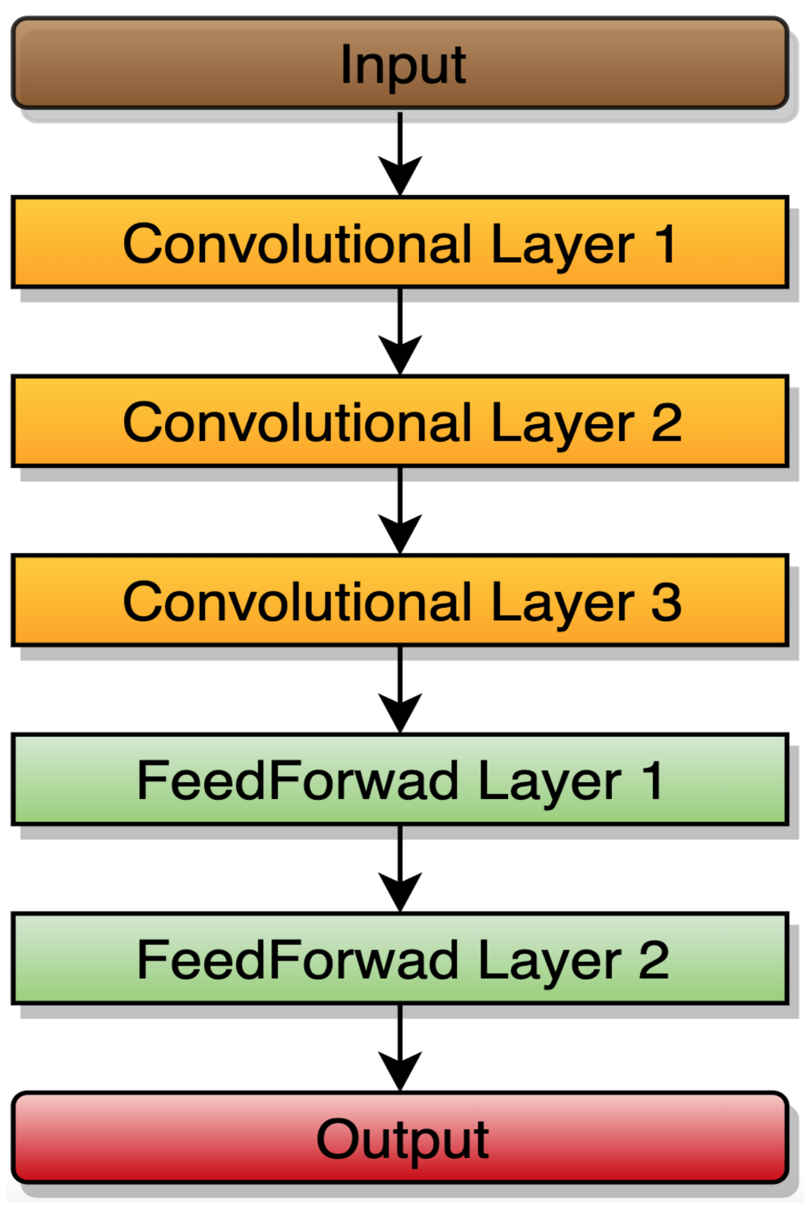


Figure 1: Complete architecture of the faction selection model created for Pong on Atari 2600.

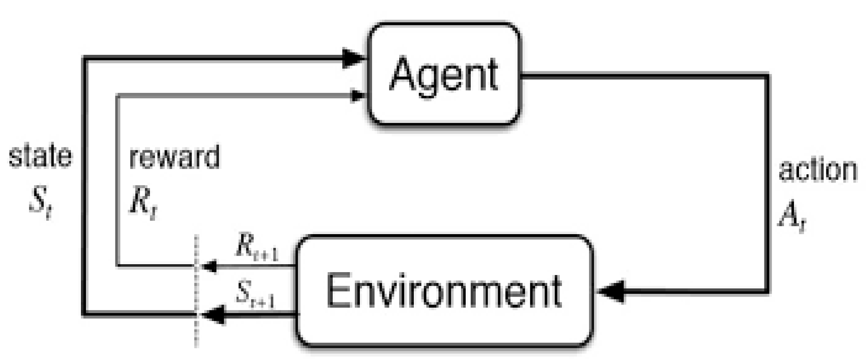


Figure 2: Agent vs. Environment control loop

a) **Exploration Vs. Exploitation:** Reinforcement learning is not restricted by the amount of training data -> take unlimited random actions. The key to improving is the **epsilon-greedy** algorithm that decreases the chance of taking random actions over the first n epochs.

b) **SGD optimization:** Approximate the non-linear function Q(s,a) with NN -> Bellman equation. However, observations are not **independent and identically distributed (i.i.d)**. Solution: **replay buffer** of experience and sample training data from it.

c) **Step correlations:** The Bellman equation returns Q(s,a) via Q(s',a'). However, s and s' are too similar. Solution: **target network**. The new network is used for obtaining Q(s',a') in the Bellman eq. and it is updated every k iterations.

d) **Bellman equation:** Means of choosing actions while considering the immediate reward and the long-term state value.

$$Q(s,a) = r + \gamma max_{a' \in A} Q_{s',a'}$$

Figure 3: Bellman equation

4. Results

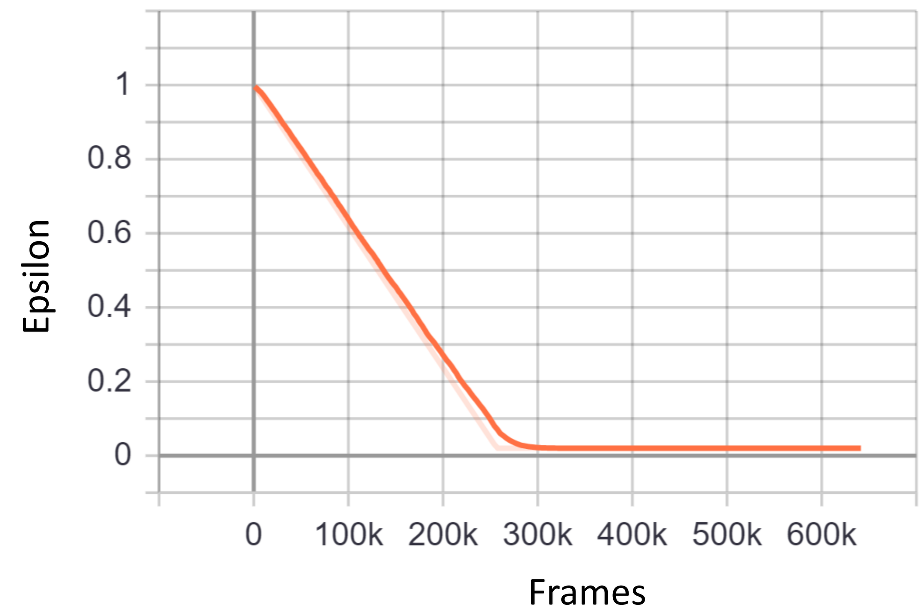
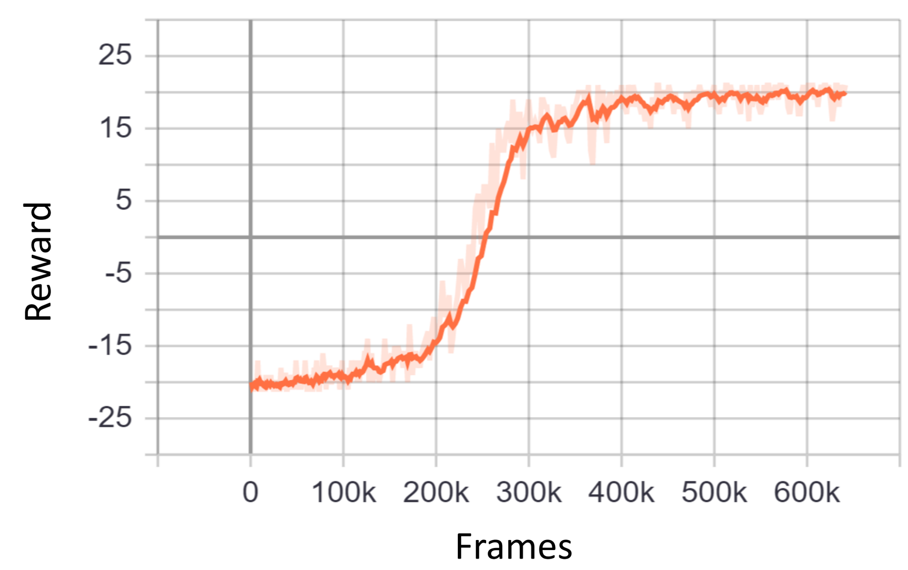


Figure 5: The decrease in the epsilon greedy value during the first 262k frames of the training process.

Figure 6: On the right, the graph shows the reward achieved by the model during the training loop which is updated every game.



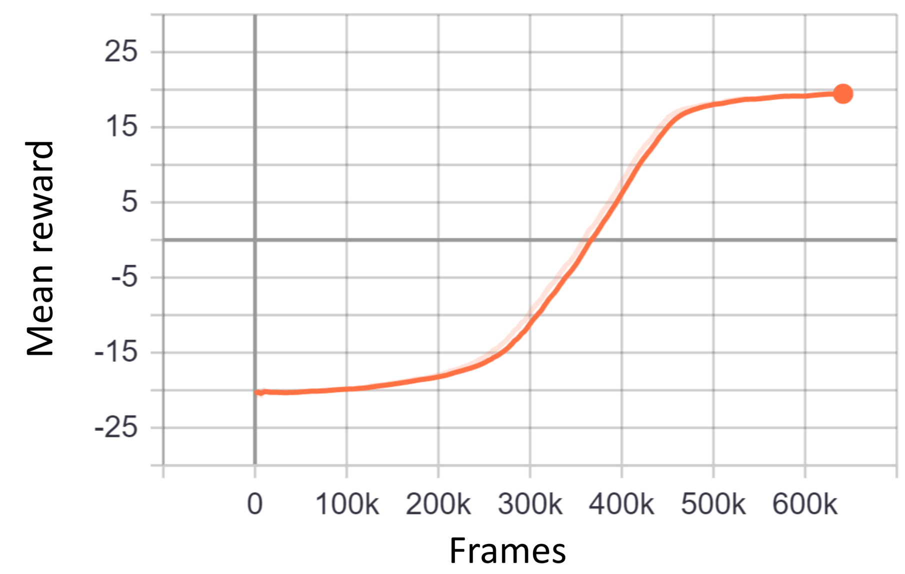


Figure 7: The mean reward over the last 100 episodes achieved by the model during training, which describes a smoothened variant of the local reward shown in the graph above.

Table 1: Model variations

Nr.	Training Games	Epsilon Decay	Batch Size	Buffer Size	Scheduler	Mean reward
1.	347	200k	32	10000	Yes	14.56
2.	470	200k	64	7000	No	16
3.	342	100k	32	10000	No	16.69
4.	382	262k	32	15000	Yes	19.52

The best model achieves a mean reward of 19.52 by beating the hard-coded agent used in Pong. Figure 5, 6 and 7 are displaying the epsilon decay and the reward on the last model in Table 1.
The second best model achieves a mean reward of 16.69, but the AI agent learns a smaller range of moves to use against the hard-coded agent.

3. Model

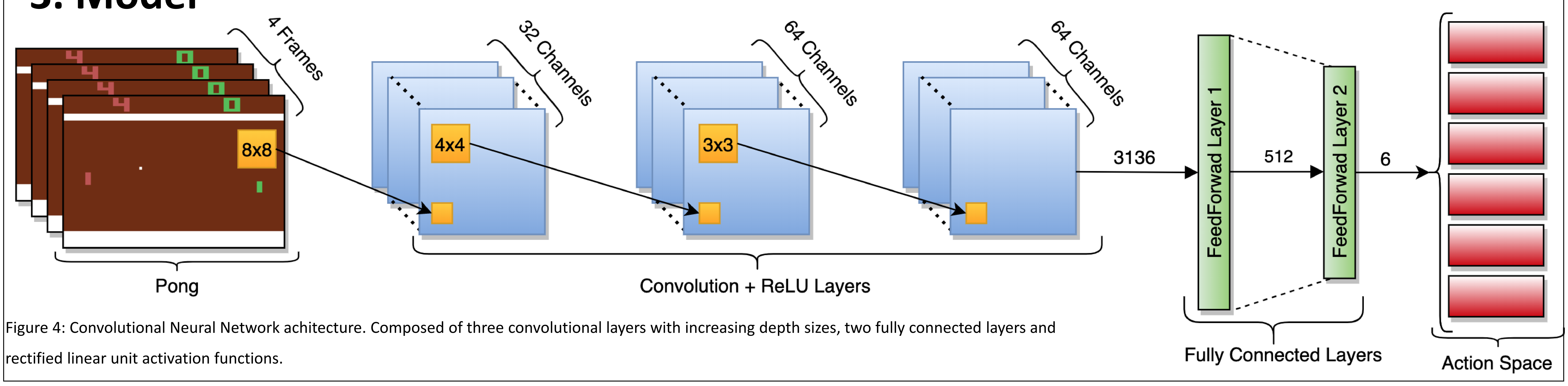


Figure 4: Convolutional Neural Network achitecture. Composed of three convolutional layers with increasing depth sizes, two fully connected layers and rectified linear unit activation functions.

5. References

Lapan, M., 2018. *Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more*. Packt Publishing Ltd.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M., 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D. *et al.* Human-level control through deep reinforcement learning. *Nature* **518,** 529–533 (2015) doi:10.1038/nature14236

Géron, A., 2017. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.".