Deep Reinforcement Learning Final Assignment

Group 1

Alejandro Sánchez Roncero
a.sanchez.roncero@student.rug.nl
(s5279402)

Yorick Juffer
o.y.juffer@student.rug.nl
(s1993623)

13 – 06 – 2023

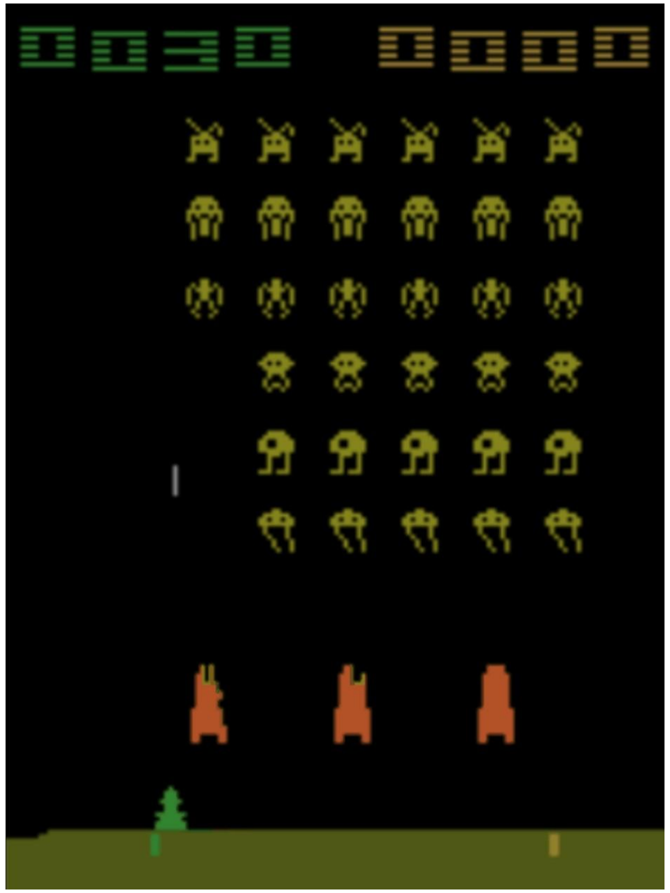
Environments

Catch
The Catch environment is a single-player game where the goal is to catch falling balls by moving a paddle left or right.

Action space includes left, right and keep position.

Space Invaders
A classic Atari game where players control a laser cannon at the bottom of the screen and are tasked to shoot down rows of descending alien invaders.

Action space includes left, right, fire, move left and fire, move right and fire and keep position.



Method - Deep Q-Network

For the **Catch** environment a Deep Q-Network (DQN) was used.

The DQN algorithm trains a convolutional neural network (CNN) as the Q-Network to output the expected cumulative rewards of any given action. Trained using the MSE loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{\langle s_t, a_t, r_t, s_{t+1}\rangle \sim U(D)} \left[\left(r_t + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a; \theta^-) - Q(s_t, a_t; \theta) \right)^2 \right]$$

Makes use of a target network, replay buffer and epsilon greedy strategy.

Method - Deep Q-Network

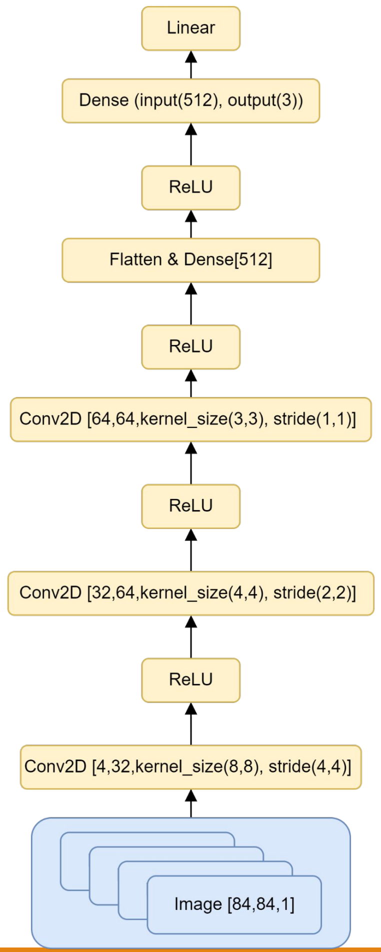
A grid search was conducted to find suitable hyperparameters.

Batch Size	{32, 64}
Discount Factor γ	{0.95, 0.99}
Episodes	{2000, 3000, 4000}
Target Update	{50, 100, 500}
Learning Rate	{0.00025, 0.0005}

Epsilon was decayed by 0.001 each episode, starting at 1.0 and ending at 0.1.

The most suitable hyperparameters were found by looking at the average reward of the last 50 evaluations which were 32, 0.99, 4000, 50, 0.00025 respectively.

The final model was trained five times and the results were averaged for a better estimate of the performance.


Linear
Dense (input(512), output(3))
ReLU
Flatten & Dense[512]
ReLU
Conv2D [64,64,kernel_size(3,3), stride(1,1)]
ReLU
Conv2D [32,64,kernel_size(4,4), stride(2,2)]
ReLU
Conv2D [4,32,kernel_size(8,8), stride(4,4)]
Image [84,84,1]

Method - Proximal Policy Optimization

For the **Space Invaders** environment Proximal Policy Optimization (PPO) was used.

- OpenAI's Gym
- Stable Baselines3

NoFrameSkip variant enables the execution of every action for precisely one step.

The PPO algorithm optimizes a surrogate loss while limiting the extent of policy updates and optimizing the value function. It also includes an entropy bonus to ensure sufficient exploration.

$$\mathcal{L}_t(\theta) = [\mathcal{L}_t^{CLIP}(\theta) - c_1\mathcal{L}_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)]$$

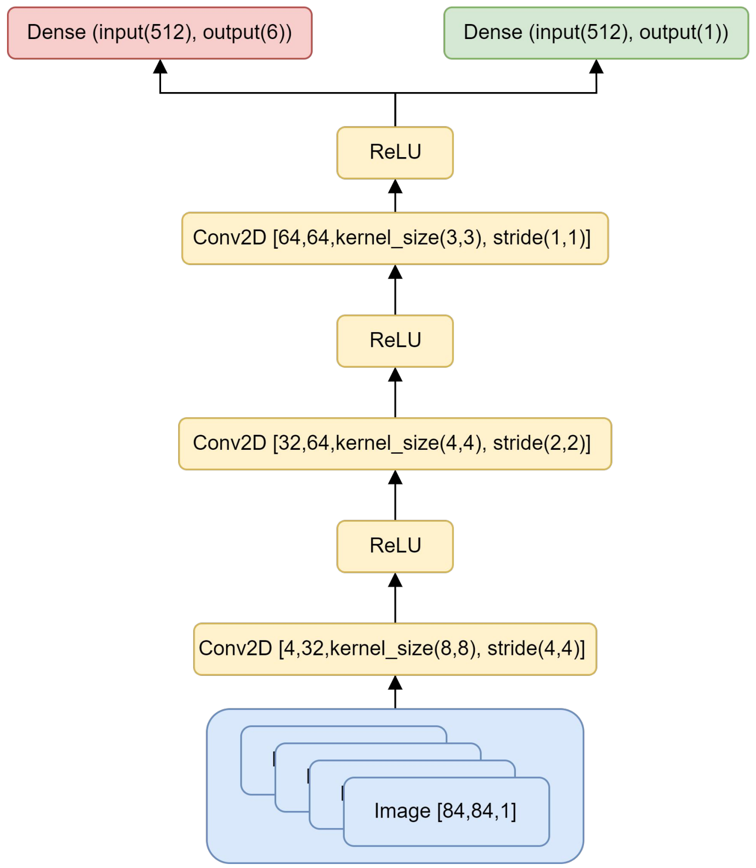
Method - Proximal Policy Optimization

A grid search was conducted to find suitable hyperparameters.

Steps	{2048, 4096}
Discount Factor γ	{0.99, 0.995}
Coefficient	{0.2, 0.5}
Frame Skips	{1, 4}

Learning rate (0.0003), batch size (64) and epoch (10) were held constant.

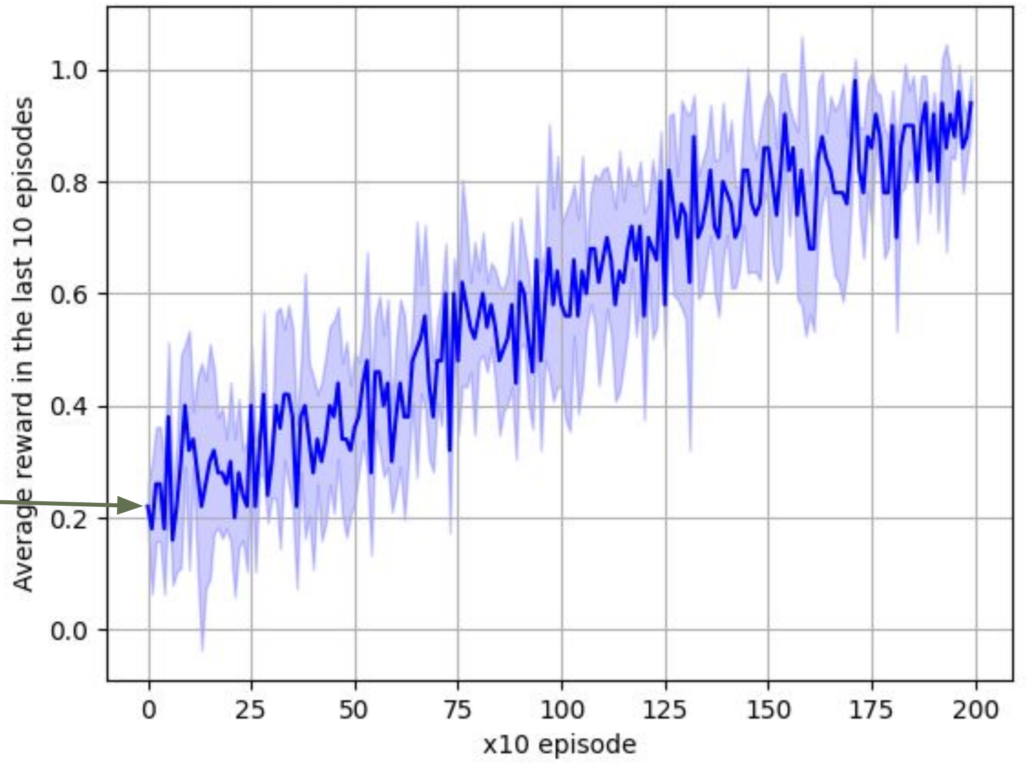
The most suitable hyperparameters were found by training for 250k and taking the two best configurations and training those for 1 million time steps. The best of the two was trained for 1 million steps for the final model.


Dense (input(512), output(6))
Dense (input(512), output(1))
ReLU
Conv2D [64,64,kernel_size(3,3), stride(1,1)]
ReLU
Conv2D [32,64,kernel_size(4,4), stride(2,2)]
ReLU
Conv2D [4,32,kernel_size(8,8), stride(4,4)]
Image [84,84,1]

Results - Catch

The average reward of the evaluation period across the five training runs of the model are considered below.

Paddle_size / width



Results - Space Invaders

The model was evaluated for 100 episodes and the mean/sd game length and total rewards were considered.

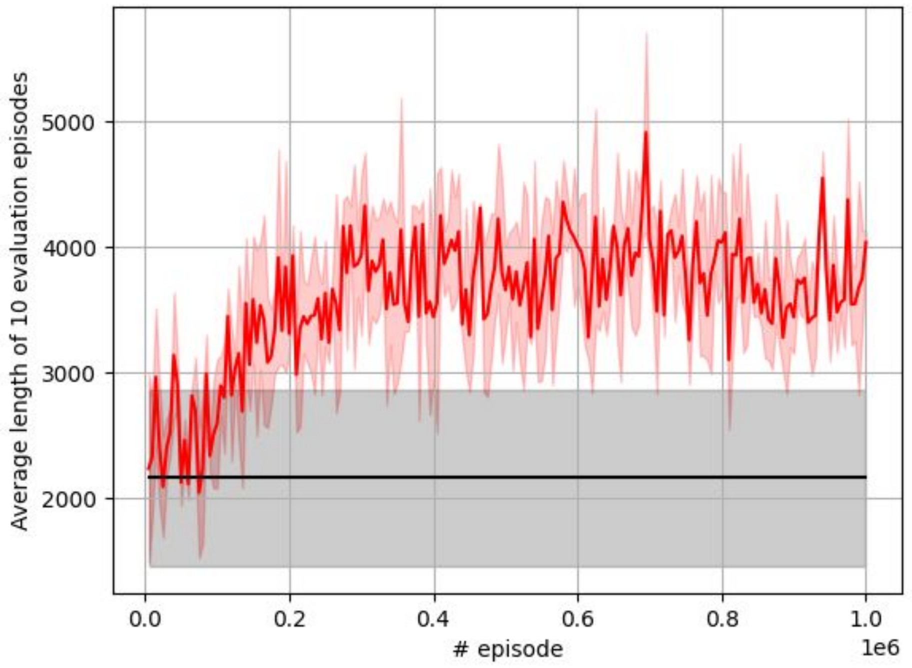
Reward	**485.50 ± 142.98**

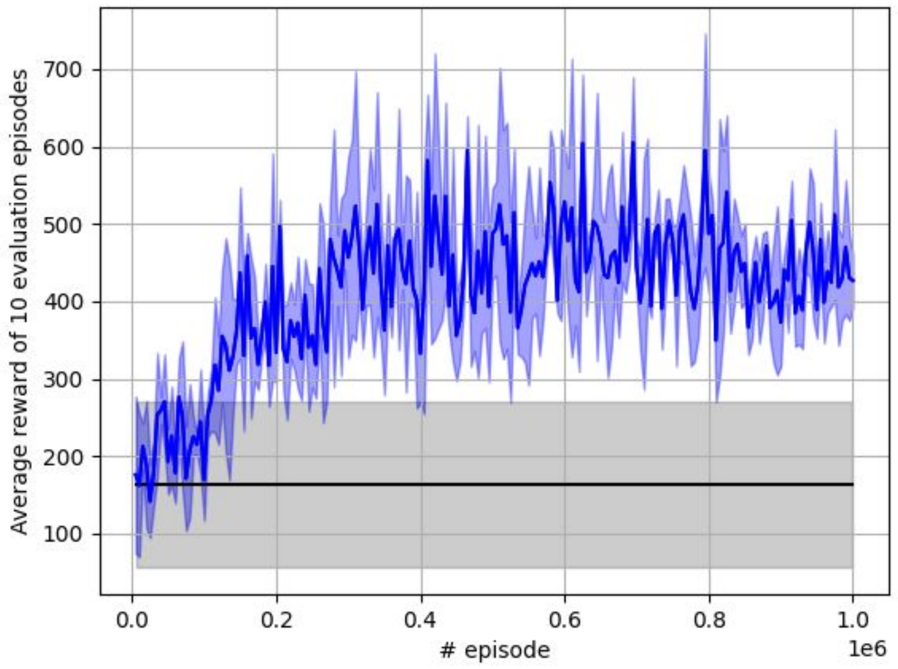
For comparison, these are the results of a model with random actions.

Reward	**149.69 ± 84.22**

Results - Space Invaders

Both the trained and random model are displayed below. On the left is the game length, and on the right the total score.





Discussion

Catch
The DQN may have seen better results if it had been trained a bit longer. It was still improving toward the end.

Space Invaders
The agent would stay close to spawn and take cover behind that barrier, even when it was destroyed. The agent would quickly die when the aliens transition to moving down the screen more quickly.

It may be the case that the agent got stuck in a local minimum and wasn't able to learn a better policy.

High variance!

Demonstration

https://1drv.ms/v/s!Aj4Mx3T9op0ijeUWkmDB0almLv3d8A?e=Bnn9MS

Thank you for your attention!