Imperial College
London

COURSEWORK 2

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Reinforcement Learning - AI MSc

Author:
Konstantinos Mitsides (CID: 01857560)

Date: November 22, 2023

Question 1

1.

In the hyperparameter tuning process, each adjustment focuses on a single hyperparameter while keeping all others fixed, with the exception of memory and mini-batch size, which are varied concurrently, as detailed in the table below. Initial fixed values for hyperparameters were set based on research and the model described in the paper (adjusted accordingly based on differences in the complexity of the problems and number of episodes used during training): $\epsilon = 0.35$, memory size = 20000, mini-batch size = 1000, learning rate = 0.01, and update network frequency = 1. The architecture, being the first hyperparameter tuned, has not any initial fixed value. The hyperparameters were tuned in the sequence presented in the table, with each tuning iteration determining the optimal value, which remained constant for subsequent experiments. The specified threshold for performance requires achieving a mean average return per episode over 10 runs of over 100 in at least 50 consecutive episodes. To align with this criterion, the emphasis was placed on mean average return over the last 150 episodes, prioritising it over the stability of the learning curve. Additionally, there was a transition from stochastic gradient descent to the Adam optimiser, known for its robustness, frequent utilisation in DQN models and faster convergence rate.

Table 1: Hyperparameter table

Hyperparameter	Value	Justification
Architecture	[4, 4, 16, 2]	Through experiments with 2 and 3 hidden layers, testing neuron counts from 4 to 64, the architecture [4, 4, 16, 2] emerged as optimal, being consistent with the problem's simplicity. This architecture strikes a balance between model complexity and task simplicity, mitigating the risk of overfitting that a more complex architecture would might have led to.
Memory size	10000	Memory and mini-batch size were jointly tested due to their interdependence. Memory sizes from 1000 to 30000 were paired with batch sizes of 1%, 3%, and 5%. Optimal performance was consistently observed with a memory size of 10000 and a 1% batch size (100), displaying the highest average return post the 150th episode. Compared to other high performing configurations, it exhibited a relatively more stable learning curve as well.
Minibatch Size	100	

Table continued from previous page

Hyperparameter	Value	Justification
Exploration rate	ϵ decreases 10% of its previous value in each episode	Many experiments were conducted by employing a variety of exploration-exploitation strategies, including low constant epsilon, linearly decreasing epsilon, and exponentially decreasing epsilon. Among these trials, the approach of setting epsilon to be 10% less than its value in the previous episode consistently outperformed others in achieving a high mean return over the last episodes and relatively stabilizing the learning curve.
Learning rate	0.01	Values ranging from 0.001 to 0.05 (inclusive) were tested. Among these, 0.01 produced the most favourable results in terms of average return performance over the last episodes and the stabilization of the learning curve. Given our training duration of only 300 episodes, the faster convergence of 0.01 compared to lower learning rates may explain these observations.
Update network frequency	1	Experiments with values ranging from 1 to 10 were conducted. Probably due to the relatively low training duration, only one value, apart from 1, met the threshold performance provided, and that was 2. However, despite the beneficial impact of 2 on stabilizing the learning curve, the value 1 was preferred since it led to a significantly higher average return over the last 150 episodes.
Optimiser	Adam	Adam converges faster than SGD, increasing the likelihood of achieving high results within the limited training duration. Additionally, it is a common choice for DQN models.

Strategy behind the range chosen to vary hyperparameters during testing

The general approach involved initially exploring a wide range of values surrounding the fixed initial values mentioned in the introduction to test each hyperparameter. After each tuning iteration, if the optimal value was at the low or high bound, I would repeat the tuning iteration by adjusting the range of values accordingly. However, if the optimal value fell somewhere in the middle, I would proceed to the next iteration. Once the iteration cycle was complete, if the final learning outcome did not significantly outperform the threshold, I would revisit the cycle using a range of values with a much lower spread, centered around the previously identified optimal hyperparameters. Luckily, the results of the first tuning cycle were satisfactory. Note that, although the network architecture [4,4,16,2] was a lower bound, I refrained from moving the range of values a spread lower, as intuitively, this could have resulted in an overly simplistic network model prone to higher bias.

Final comments

During the experiments, I observed a tradeoff between the stability of the learning curve (indicated by the standard deviation of returns) and the mean average return of the last 150 episodes of the training period. This tradeoff could be minimized by significantly increasing the training duration and adjusting the values of some hyperparameters to suit this extended duration. However, the training duration remained at 300 episodes throughout the experiments due to limited available time.

2.

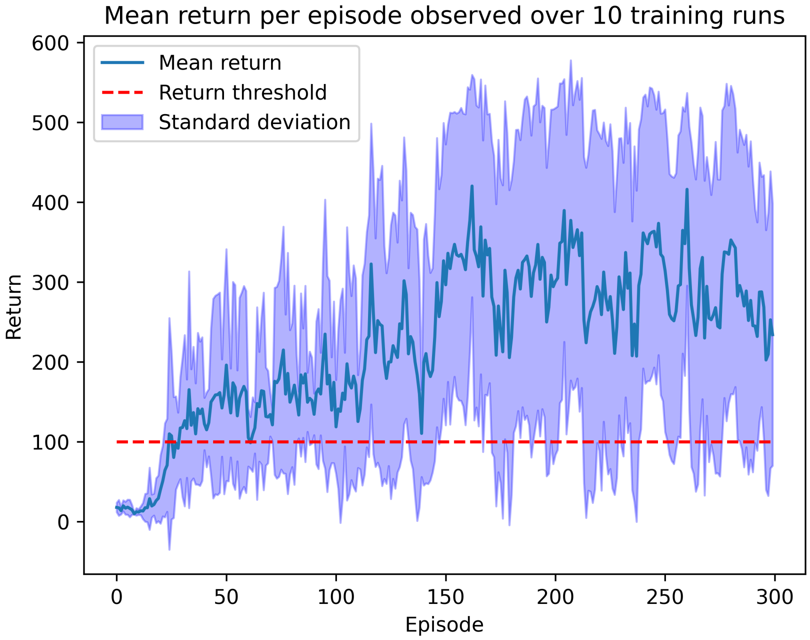

Figure 1: Learning curve of DQN agent plotted as mean return per episode with the associated standard deviation observed over 10 training runs

The learning curve exhibits a smooth increase over the initial 25 episodes. Subsequently, it shows a slightly more unstable ascent, peaking around the 150th episode before rapidly dropping to nearly a return of 100 at approximately the 150th episode. After the 150th episode, it fluctuates between returns of 225 and 425. In particular, the mean average return over the last 150 episode is 299 (3sf), whereas the mean average standard deviation of the return over the last 150 episodes is 189 (3sf). It is concluded that although the agent achieves some high returns, it lacks consistency, as evidenced by the substantial standard deviation. A longer training duration, along with a target network update frequency greater than 1, and a higher memory size could potentially stabilize the learning curve.

Question 2

1.

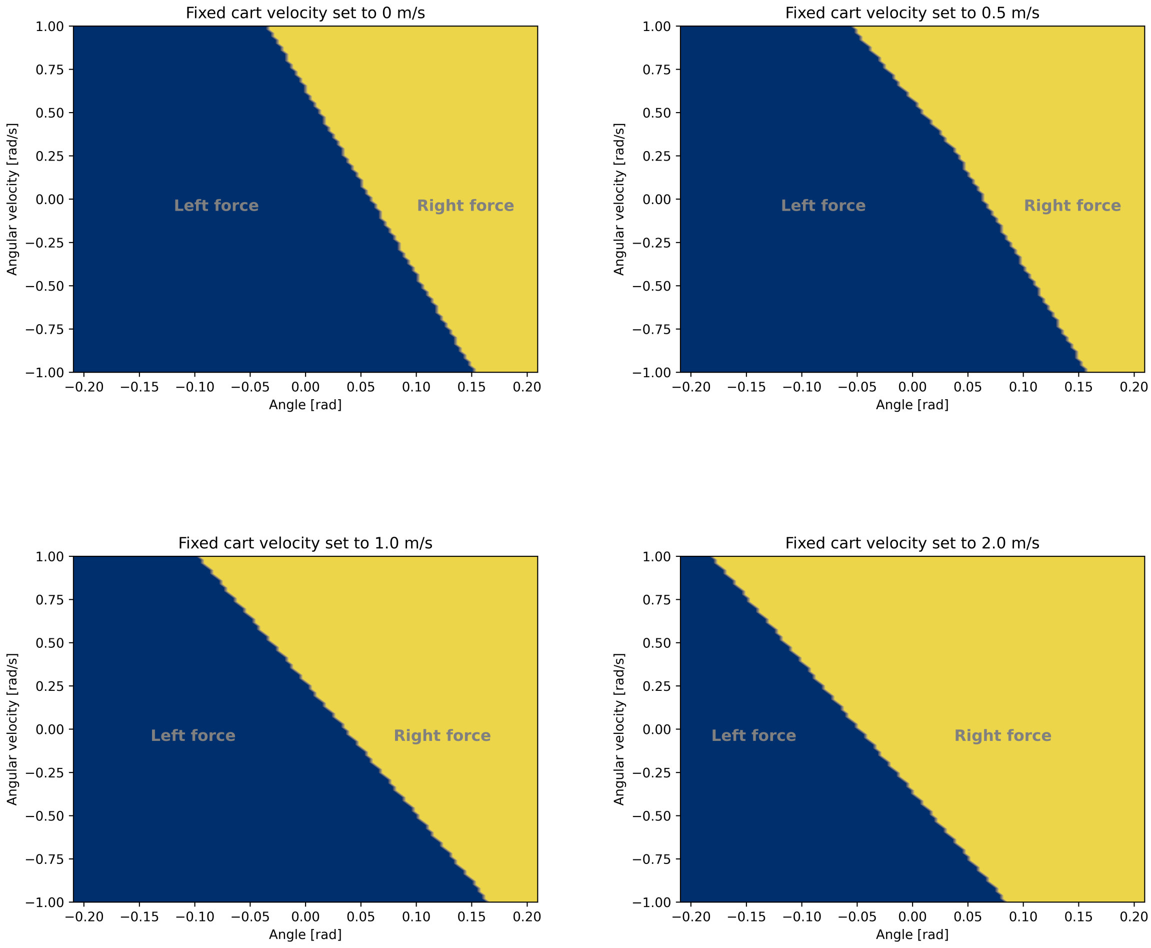

Figure 2: Greedy policy of DQN agent with varying angular velocity and pole angle, and fixed cart position and velocity

General effect on policy without considering velocity

Assuming the angular velocity is zero, when the pole angle is positive (leaning towards the right), the desired action is to exert a force on the cart to the right for balance, and vice versa. The graphs indicate that the agent's policy aligns with this principle, with the right side of the graphs generally coloured yellow (indicating a rightward force) and the left side coloured blue (indicating a leftward force). However, the policy observed in the graphs is not entirely optimal, as it occasionally exhibits suboptimal behaviour. For instance, when the velocity is set to 0 and the angular velocity is also 0, an ideal agent would apply a leftward force if the pole angle is negative, or a rightward force if the pole angle is positive. This implies that the boundary line should have intersected the coordinates (0,0). Evident in the top left graph, our agent's behaviour acts differently in this case, generally showing a slight preference for exerting a leftward force over a rightward one, and thus causing the boundary line to shift slightly to the right.

When the angular velocity is not zero, the situation becomes more complex. For example, a combination of high angular velocity and a small angle of the pole in the opposite direction might suggest exerting a force in the opposite direction to the pole's lean, as the angular velocity may have a more significant effect. This complexity is reflected in the graphs, where a diagonal boundary line emerges, illustrating a tendency for a rightward force with high positive angular velocity and low negative pole angle, and vice versa.

Velocity effect on policy

(*) Let f be the frequency at which a force is exerted on the cart, and let x be a negative pole angle. Then, there is a range of x values, where if f is adjusted accordingly, the cart will consistently move to the left with the pole remaining almost stable over time (the pole keeps swinging having maximum angle x).

As the cart velocity increases, the boundary line in the graphs shifts to the left. In particular, there is an increase in the number of states where the agent pushes the cart to the right instead to the left. This can be explained by (*). Since the cart moves to the right (has positive velocity), and considering that the episode terminates if the cart distance from the center exceeds 2.4m, the agent aims to maintain a stable pole while moving the cart to the left as quickly as possible. To do that, the agent aims to achieve a particular pole angle 'x', within the range defined by (*). Achieving that angle, while adjusting f accordingly, the agent might prevent the episode from terminating. Hence, in states where the pole angle deviates from the value x, either by being a smaller negative angle or a positive angle, the agent pushes the cart to the right to achieve the desired pole angle. Consequently, there is an increase in the number of states, in particular those concentrated in the middle of the graphs, with a corresponding optimal action of a rightward force.

2.

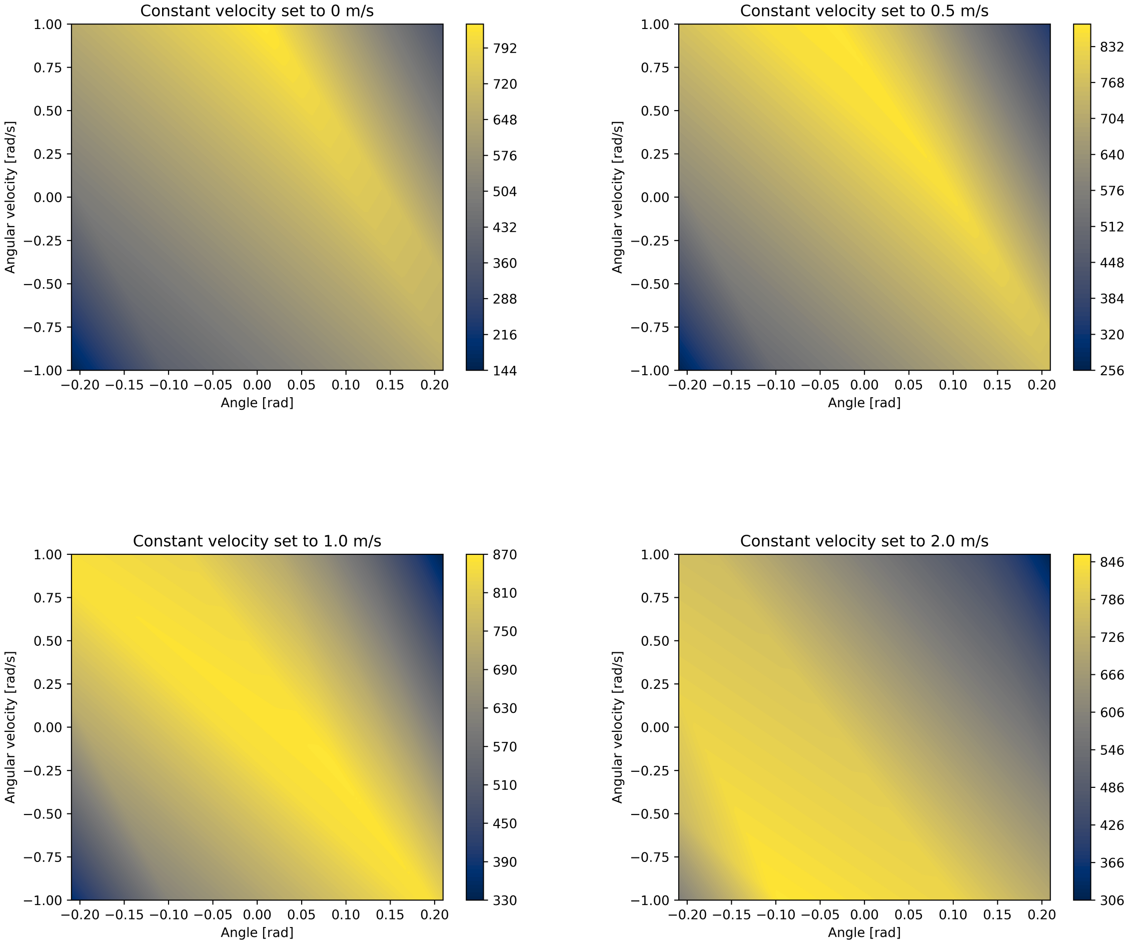

Figure 3: Greedy Q values of DQN agent with varying angular velocity and pole angle, and fixed cart position and velocity

Effect on Q values without considering velocity

In the above graphs, the Q values range from low values, indicated by dark blue colouring, to high values, indicated by dark yellow colouring. According to the graphs, states of high positive angular velocities paired with high positive pole angles, and high negative angular velocities paired with high negative pole angles, tend to have lower Q values. This observation aligns with expectations, as these states indicate that the episode faces the risk of termination, implying a diminished margin for suboptimal actions by the agent. Consequently, learning to act optimally in such cases becomes more challenging, leading to lower Q values. Conversely, the central region of the plots consistently displays yellow colouring, indicative of higher Q values. This corresponds to a ”safe region”, where the pole is close to a balanced position. In this case, the agent has a wider margin for suboptimal actions, facilitating a more straightforward learning process to act optimally and resulting in higher Q values. Intriguingly, there is an observation of Q value overestimation, as many Q values are larger than 500. In particular the Q values from all the cases are ranging from 144 to 870. Nevertheless, the learning curve demonstrates satisfactory performance given the low training duration, and the agent's policy seems to be generally on point. This suggests that, for a robust policy, the agent's ability to estimate the relative quality of returns from different states is more crucial than precise estimation of the return value for a particular state.

Velocity effect on Q values

If the cart is moving to the right, the episode faces the risk of termination because:

(a) pole angle may get greater than 0.2094 rad (1)
(b) cart distance from the center may get greater than 2.4m (2)

When the pole leans to the right (has a positive angle), there is no singular action capable of simultaneously addressing both issues (a) and (b). A leftward push increases the risk of (a), while a rightward push increases the risk of (b). Consequently, a strategic combination of optimal actions needs to take place to prevent the episode termination. This significantly challenges the agent's learning process to act optimally, resulting in lower Q values. Conversely, when the pole leans to the left (has a negative angle), one leftward push may suffice to prevent episode termination. In this case, it is easier for the agent to learn how to act optimally, resulting in higher Q values. These dynamics are visually evident in the graphs, where an increase in cart velocity shifts the yellow-colored region slightly to the left, and automatically causes a reduction in the lower left blue region and an expansion of the upper right blue region.