Reinforcement Learning Project: Midway report

erazumov **ntadiche**

1 Introduction

In the project, the work is conducted with the navigation environment. As a baseline method, a simple DQN with replay memory was chosen. As a simple extension to that, we have decided to experiment with double DQN network.

2 Methods

The reinforcement algorithm with which experiments were conducted in this part is a simple DQN with experience replay. The experiments were conducted with different hyperparameters of the network and the environment.

The overall archtechture was as follows. The vision of two consecutive states was passed through the convolutional layers, then flattened and concatenated with the scent information. Then, this vector was passed through the feedforward layers. In all the layers, ReLU was used as the nonlinearity. Epsilon greedy policy was used for the choice of action. We have used the Adam optimizer.

The experiments included varying the number of convolutional layers from 1 to 3, the number of channels in the convolutional layers, size of the kernel, excluding the MaxPooling layers from the top of convolutional layer, varying the number of linear layers from 1 to 3, varying the gamma between 0.5 and 1.0. The learning rate was varied between 0.001 and 0.1. In addition, the double network with the best determined architecture was trained.

3 Preliminary results

The hyperparameters of the best performing model were as follows. The vision information was passed through 2 convolutional layers with 16 and 32 channels in them with with the kernel size of 3×3without the maxpooling layers. The scent was concatenated with the flattened output of convolutional layers. The result of concatenation was passed through 3 feedforward layers with 1000, 100 and 32 units in each. The best learning rate and gamma were 0.01 and 0.9, respectively.

The results of the best performing model are presented in the figures below (figs. 2, 3, 4, 5). When looking at the cumulative rewards (figs. 4, 5) we can see that the the cumulative reward keeps growing, thus, the agent keeps increasing the overall rewards. However, when we look at the reward for every next 100 steps, we can see that the rewards obtained are quite sparse. There is a break between 40000 steps and 70000 when the average reward for 100 steps is 0. This could be due to the plotting style as from the cumulative reward graph we can see that the cumulative reward keeps growing throughout this period.

Preprint. Work in progress.

4 Future work

We plan on extending the work further by exploring two ideas i.e., Imitation Learning and Distributed Learning. These ideas stem from the fact that the environment is filled with sparse rewards and the results so far have shown that it is going to be quite challenging for the network to learn from scratch.

The idea of imitation learning here is that agents could learn from the human experts and try to clone the behaviour to interact with an environment in the same way as an expert demonstrator. From a human's perspective, maximizing the reward in the given game world of 'Navigate, Fetch and Find' is a trivial task. The human demonstrator would rely on two modalities of sensor input, namely, vision and scent. The human way to move around in the environment is to first rely on the obtained vision features and since it is of short range and then get a global sense of direction using the scent values where each element corresponds to tongs, diamonds and jelly bean confidence scores respectively.

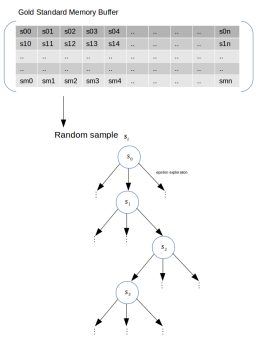
Since the above would possibly requires hundreds of human hours of gameplay, we would like to see how a mix of human behavior modeling and epsilon-exploration policy would work by randomly branching away from the human path. Implementation-wise, the human demonstrations (a fixed number of steps) can be stored in a memory buffer as gold standard paths and these could be used while training time by picking a path randomly and branching out at different stages of the path based on epsilon-exploration policy. The intuition is that the network would overtime visit every state in the environment and would not fail when it encounters a state which is not encountered by the human demonstrators path.

Coming to the other idea of distributed learning, we believe that this problem could be broken down into multiple learning problems. First, the agent should learn to maximize the number of tongs being carried so that it can gain high rewards from diamonds. Second, it should learn to maximize jelly bean rewards. Third, it should learn to maximize the diamond rewards. As a low level learning scheme, we can train different networks to learn these individual behaviours, and then, we can use ensemble methods to poll and update a common head network. It seems very likely that the learning should follow a sequential pattern for updates because of the inter-dependencies of rewards i.e., we need tongs first to hold the diamonds and the jelly beans exist mostly around diamonds.

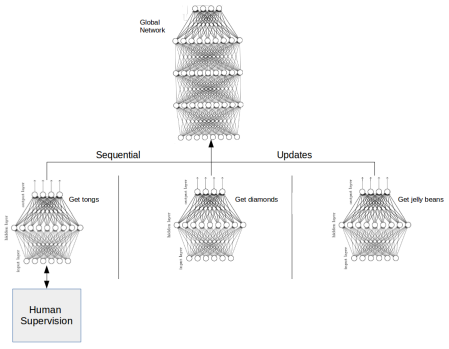
The above can also be trivially implemented by running local learning on three different threads and having a global update network which takes a weighted average of the update based on the total reward obtained.

Lastly, both the ideas mentioned above can be merged seamlessly. We think that this might also be necessary because obtaining tongs do not have rewards associated with it but it is a precursor to obtain the high diamond rewards. So we will use human demonstration to pick up tongs and then merge these updates with a fixed weight factor with the global network.

The timeline Firstly, we plan to manually produce the the gold standard paths and train the network based on those by 11/10. Next, we will train the low level networks specialised on picking up the three different objects by 11/14. Next, we will experiment with the ways to ensemble the three models and the imitation and distributed learning. We plan on finishing the main experiments by 19/11 and work on the write up after that. The work will be distributed equally between us.



(a) Behavior Cloning



(b) Distributed Learning

Figure 1: Ideas to explore

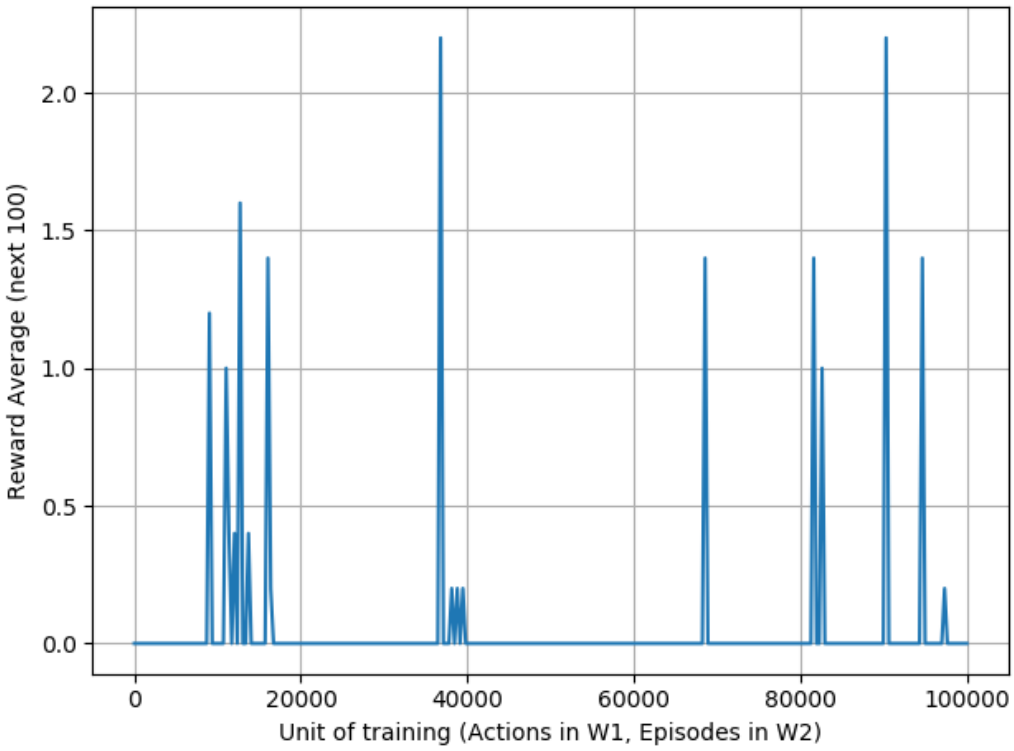


Figure 2: Average rewards

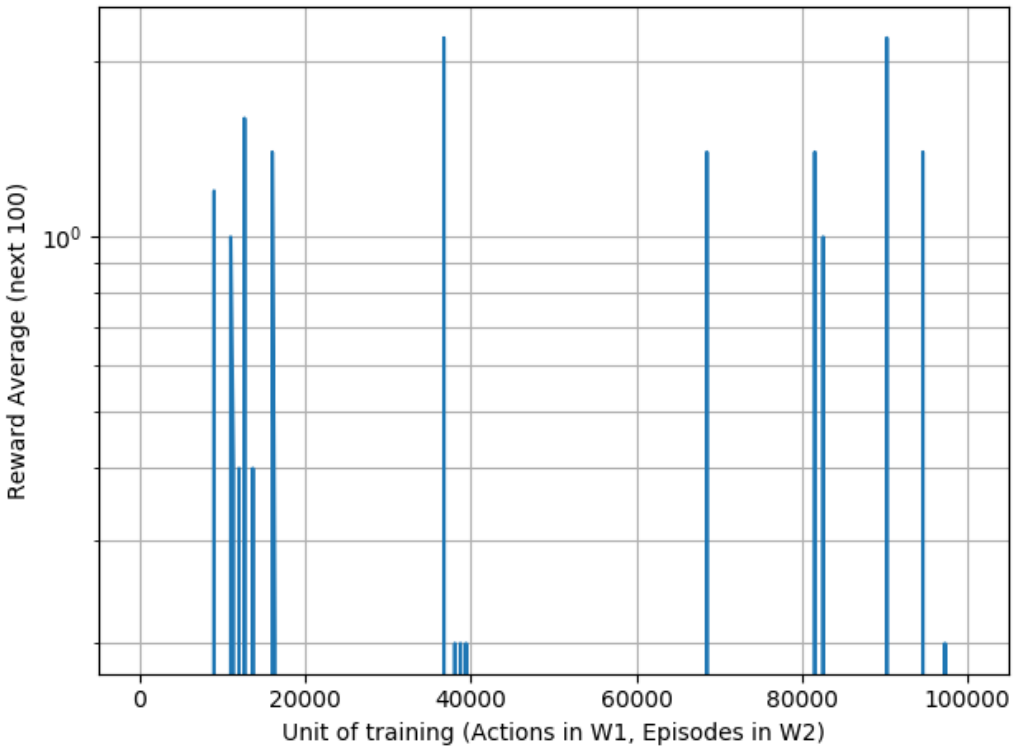


Figure 3: Average log rewards

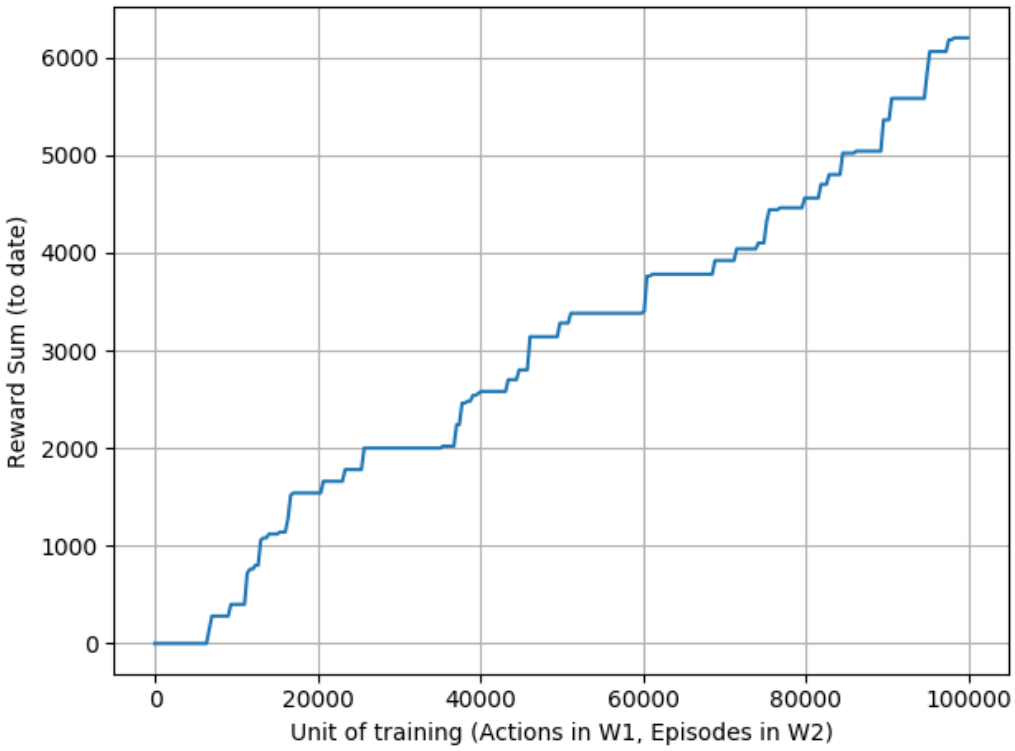


Figure 4: Cumulative rewards

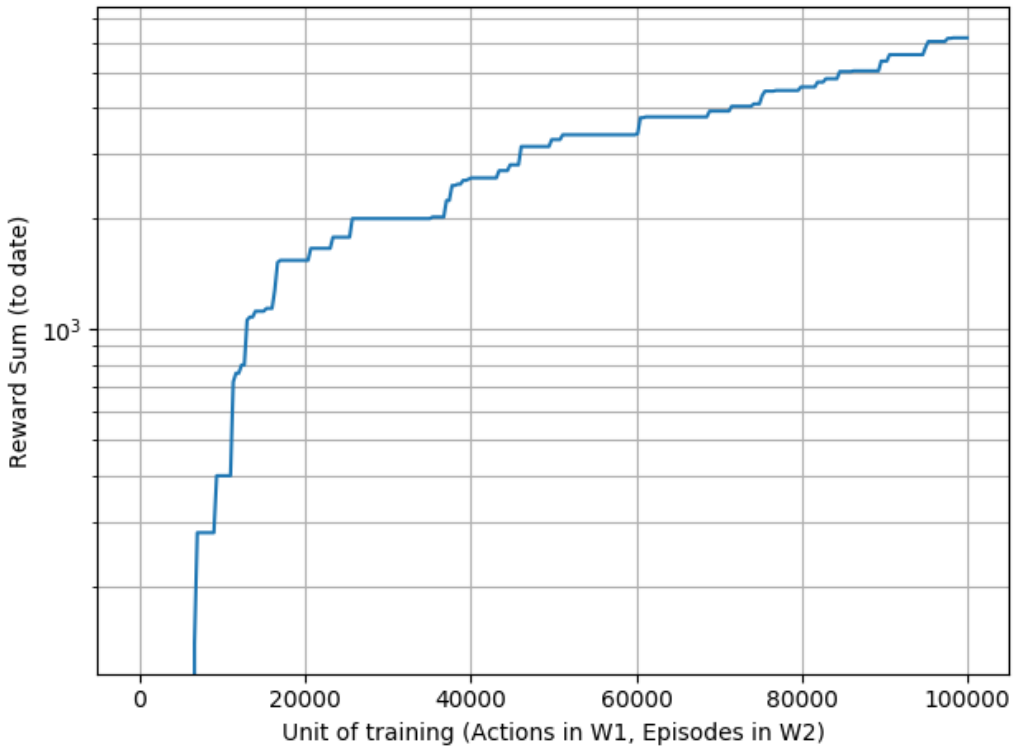


Figure 5: Cumulative log rewards