BiPedalWalker-V2

Saurabh Kumar *2015088* Nishant Sinha *2015066*

***Abstract*—This project demonstrates the approaches to solve the BipedalWalker-v2 problem on OpenAI gym from scratch. The goal is to make the agent walk toward right on the rough terrain without falling. We have simulated the environment using OpenAI Gym and used reinforcement learning and genetic learning for model free control. Hence no data-set required.**

I. Problem statement

Bidedal Walker is an agent which learns to walk on rough terrain. Environment is provided by openAI gym. Action is an array of size 4 where each element defines the velocity of the 4 joints of Bipedal walker. As reward, Bipedal walker receives positive reward for moving forward summing upto 300+ for reaching far end, -100 for falling, and some negative rewards for applying motor torques. state consists of hull angle speed, angular velocity, horizontal speed, vertical speed, position of joints and joints angular speed, legs contact with ground, and 10 LIDAR rangefinder measurements.

II. Literature review

A. Human level control through deep reinforcement learning

This revolutionary paper for the first time stabilized deep networks with q learning. The major noteworthy ideas in this paper are that of experience replay and that of a target network. Using the experience replay helps to maintain the i.i.d. assumption which is required for most stochastic based algorithms to converge. The target network helps keep the target steady while the behavorial network tries to come closer to it.

B. Deep Reinforcement Learning with Double Q Learning

This paper first investigates if the performance of the earlier DQN algorithm was affected by maximization bias. This kind of bias forces the agent to choose the wrong path for a long time if on the first occurrence of that state, it recieved a high reward from the noisy probability distribution. In order to tackle this problem, the authors introduce a second network and decouple the selection of the next maximizing action with the actual value selection of that state action value.

C. Prioritized Experience Replay

This paper claims that some experiences are more valuable than others and tries to prioritize replaying more valuable experiences over the rest. This problem is specifically valuable to our current problem of interest as most of the initial experiences of the biped involve redundantly falling over. Hence rewards are very rare and so are the experience which will actually help us learn to walk. The paper uses TD - error as the metric to prioritize experiences and notice almost exponential speedups.

D. Deep Recurrent Q-Learning for Partially Observable MDPs

This paper tries to solve the problem that most of everyday environments around us are POMDP and not an MDP. Even the atari environment is a POMDP if we don't stack the frames together. This paper introduces using an LSTM in order to store more temporal information about the observations and hence successfully solves POMDPs like the blinking pong.

III. Methodology followed

We implemented all the different q networks and trained extensively on them. However we soon realized that we were short on training resources as making a biped walk is a fairly hard reinforcement learning problem. We then tried considering the environment as a POMDP instead of an MDP by first stacking multiple frames on top of each other and then later trying a RNN based Q learning. However meanwhile we were able to get amazing results using genetic algorithms.

A. Deep Q-Learfning

We followed the same algorithm as mentioned in the Nature paper by Deep Mind using Experience Replay and a Target network to stabilize results.

B. Double Deep Q-Learning

Another DeepMind Paper which builds on the DQN algorithm by using two separate networks for target selection and prediction. This helps reduce maximization bias.

C. Prioritized Double Deep Q-Learning

Some experiences are more useful and enable more learning than the rest. This is especially evident in the case of a bipedal walker in which most experiences are utter failures with not much to learn. We prioritize the experiences based on their TD-error.

D. Stacking of frames and Deep Recurrent Q-network

We consider that the bipedal walker is not an MDP but a POMDP where one observation and the corresponding action does not completely define the future. Hence we first try stacking 4 frames together before sending it to our model. Also, we tried to use an LSTM as the second last layer in order to remember some context.

E. Genetic Learning

We also tried to solve the problem using genetic algorithm. we considered gene to be the layers of weights and biases of a neural network and chromosome as neural network with layers [24, 100, 80, 25]. we defined fitness function as the total reward of the complete episode. We choose from parents to be a part of the new generation based on the probability. P = (number of parents - index in sorted list based on fitness function)/ number of parents. And rest of the chromosome comes from these above selected parents only. Better the parent's fitness score better their probability to get selected for random child generation/crossover for next generation of population.

IV. Results

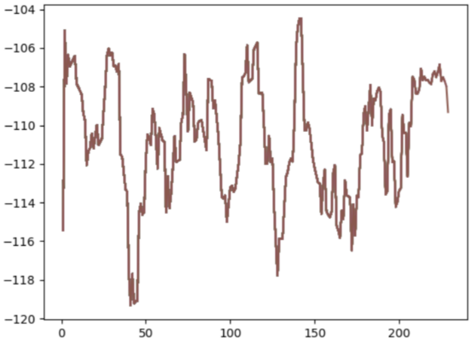

Fig. 1. DQN

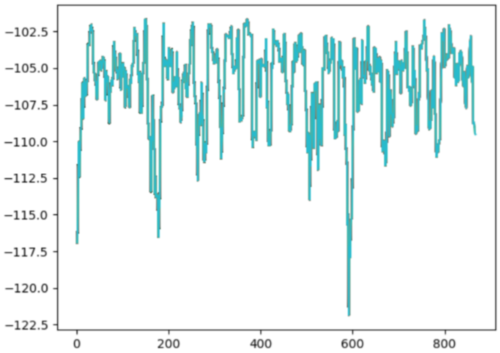

Fig. 2. Double DQN

V. Analysis

A. Reinforcement Learning based approaches

1) What worked nad what didn't: We are sure that the reinforcement learning based approaches that we tried would all work given enough training. However the major reason why this is such a hard reinforcement learning problem is that the reward space is really sparse. We tried the Prioritized experience replay in order to track this problem but the training grew slower and slower as the size of our replay memory increased. This was due to the fact that we had to sample from the memory according to a certain probability distribution.

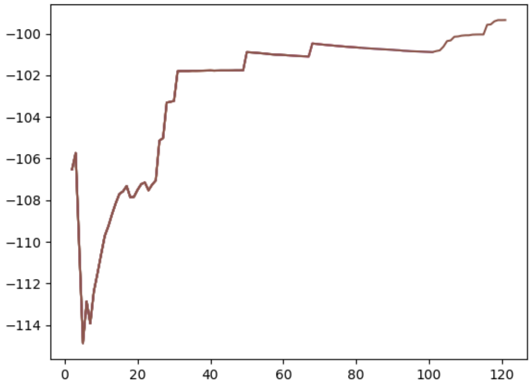

Fig. 3. Prioritized Double Deep Q-Learning

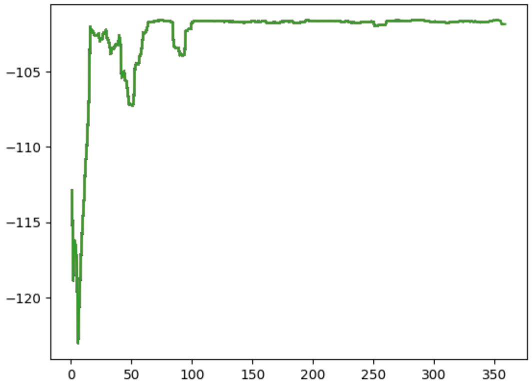

Fig. 4. Stacking of frames

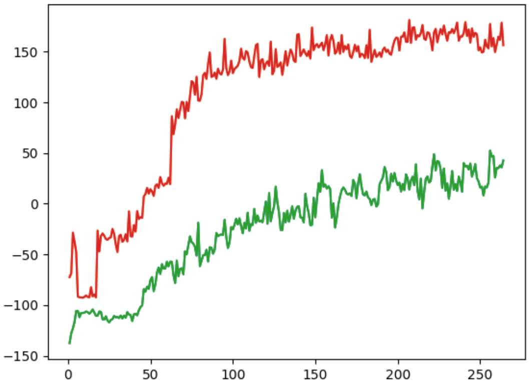

Fig. 5. [GA]Max score(red) and average score(green) over generations

B. Genetic Learning

We can use genetic algorithm instead of gradient descend to update/generate Neural network's weights and biases. This neural network can be used to learn the problem where size of observation space is very large or infinite. Genetic learning performed very well on our problem statement. Fitness score of best and average chromosome is shown in fig1. The process of convergence was computationally very expensive as we are evaluating entire population every generation. This made the program very slow.

VI. Summary

Bipedal Walker is one of the most difficult problem of OpenAI gym which so far has been solved by only 2 people. We tried DQN, Double DQN, Prioritized Double DQN, Stacking Frame but our model tend to either saturated or became unstable. We will need to train the model for way more time stamp to come to any conclusion. On the other hand, genetic algorithm, although being too slow, slowly converges to get more reward and reduce motor torque based cost. This shows how powerful heurestic based approaches are on tough problems. This is almost unbelievably amazing!

References

[1] Hausknecht, Matthew and Stone, Peter *Deep recurrent q-learning for partially observable mdps*.

[2] *Van Hasselt, Hado and Guez, Arthur and Silver, David*. [*Deep Reinforcement Learning with Double Q-Learning.*].

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller. *Playing Atari with Deep Reinforcement Learning*.

[4] *Tom Schaul, JohnQuan, Ioannis Antonoglouand ,DavidSilver*. [*PRIORITIZED EXPERIENCE REPLAY*].