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 terrain without falling. We will simulate the environment using OpenAI Gym and use reinforcement learning for model free control. Hence no data-set required.**

I. Task Completed

A. Approach-1

We implemented a deep-q-learning model based on DeepMind's paper which uses DQN with experience replay. Since training it on the hard bipedal walker problem will definitely take days before any meaningful results, we run our model on the simple cart pole balancing problem discussed in Sutton's. There were two hidden layer with 24 nodes each and SGD was used to update the weights. Epsilon was initially 1 and had a decay rate of 0.995. The results we got were really meaningful as after training on 100 episodes our cart was able to balance the pole perfectly by changing its direction.

B. Approach-2

We also attacked the problem statement with traditional Q-learning approach. In each game, we fed the network with observations and extracted the action based on the best probability. Then we used Q-learning to reflect the effect of final reward backward to the previous steps. After every 10 games, we updated the network gradient based on the reward we received and repeated the above mentioned steps to a particular number of iterations. Following were the observations:

1) Adding dropout factor was not helping with over-fitting, rather it was resetting the graph to some extent.
2) We made the agent take random action over small probability of games to make sure it doesn't get stuck in local maxima but it didn't prove to be helpful as it was also resetting the graph.

II. Results

A. Approach-1

1) The episode terminates successfully if the cart stays upright for 500 time-steps.
2) Our model is successful and has an average score of 500 over 100 games, given that we always exploit(No exploring)
3) Since q-learning is a form of td-control, and is model free, we can safely assume that this model will successfully train our bipedal walker, given enough time.

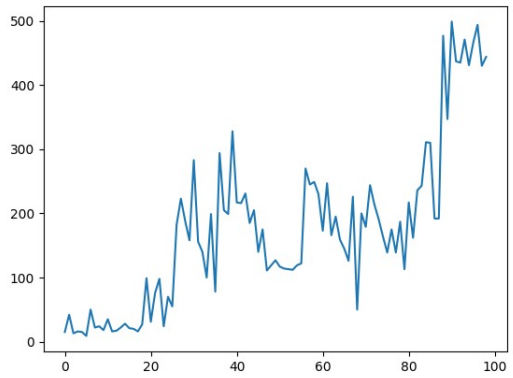

Fig. 1. Score VS Iteration. The given graph represents training and here does not reach maximal score because we continue exploring with a minimum epsilon of 0.01.

B. Approach-2

Currently, agent is making crippled attempt to move forward without falling. The possible causes for this can be insufficient number of iterations over which the model was trained, Number of layers, over-fitting, too much discretization of rewards.

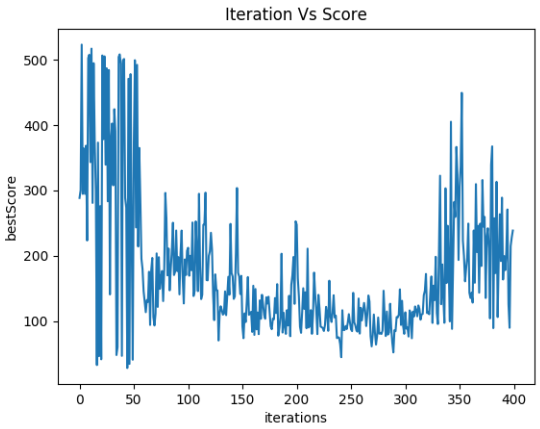

Fig. 2. Best Score VS Iteration

III. Future work

1) We'll first try to tune the parameters such as discount rate, number of hidden layers, dropout factors etc. and train it over much more iterations.
2) Try using double learning to reduce maximization bias.
3) We'll have to find out some ways to avoid too much discretization of rewards.