Project - 4
Reinforcement Learning
(Bonus Task Report)

Instructor: Sargur Srihari

Name: Varun Nagaraj
UB Person number: 50290761

Introduction

In this bonus task we implement Deep-Q-Network programs using 2 environments. One being the OpenAI's Gym library and the second being Atari implementation for games created by Atari Inc. like breakout, pacman etc.

So what is DQN?
DQN is a reinforcement learning algorithm that combines Q-Learning with deep neural networks to let RL work for complex, high-dimensional environments, like video games, or robotics. This technique is able to combine reinforcement learning with a class of artificial neural network known as deep neural networks. Notably, recent advances in deep neural networks in which several layers of nodes are used to build up progressively more abstract representations of the data, have made it possible for artificial neural networks to learn concepts such as object categories directly from raw sensory data.

We implement DQN in the below two tasks using the stable baselines fork which is available to the public to refine and implement and learn from the results of playing the game. Stable Baselines is a fork that aims at making Reinforcement Learning accessible to a broad audience.

OpenAI Gym:

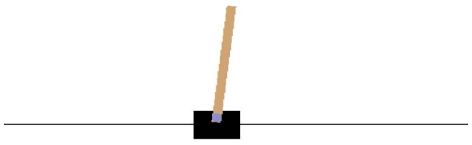
Gym is a toolkit for developing and comparing reinforcement learning algorithms. It makes no assumptions about the structure of your agent, and is compatible with any numerical computation library, such as TensorFlow or Theano.
Gym attempts to fix two issues in Reinforcement Learning, which are

1. Lack of standard benchmarks
2. Lack of standardization of environments used in publications

Cartpole game implementation:

In this I create the environment in gym by using make() present in gym library, and pass the cartpole game name as the parameter to that. Once I have the environment to run my game in, I create the model using the environment in DQN and then start the game and make it learn from the process of Q-learning. This is done for the number of timestamps specified as parameter to the learn function. It plays the game for the specified number of iterations and then from the Q-Matrix it will try to predict the possibilities of movement of the cartpole. These outputs are rendered into an image frame and stored in a images list. At the end of certain number of iterations the image frames are combined into a video and saved as before in case of tom and jerry game.

Output:

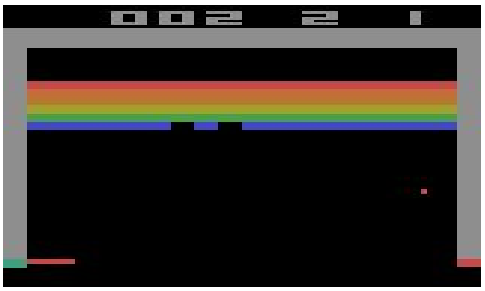


Atari Implementation with Stable Baselines:

In this I implement the Atari Breakout game using the same method as detailed above. Except the creation of the environment which is different in this case, we use the entirely same code to implement the breakout game in atari environment.

We use the Atari Environment by using the make_atari() method which creates an environment and implements the game passed as argument to it. In this case it is the BreakoutNoFrameskip-v4.

This again generated image frames which are rendered by the renderer function which has been used previously. These image frames are appended into a list and then the images are combined together to generate a GIF image using ImageIO library. We then save the image.



References:

https://towardsdatascience.com/stable-baselines-a-fork-of-openai-baselines-reinforcement-learning-made-easy-df87c4b2fc82
https://medium.com/@jonathan_hui/rl-dqn-deep-q-network-e207751f7ae4
https://stable-baselines.readthedocs.io/en/master/modules/dqn.html
https://gym.openai.com/
https://becominghuman.ai/lets-build-an-atari-ai-part-0-intro-to-rl-9b2c5336e0ec