

Project 3

jdbloom

October 2019

1 Introduction

PLEASE CLONE MY GIT REPOSITORY: <https://github.com/jdbloom/JBLOOM.Project.3.git>

PyTorch is a wonderfully complex tool. Much of my time spent on this project was simply due to lack of understanding of the tools available to me. I ran many experiments, however, most were not useful due to simple errors in the algorithm and gained knowledge from browsing forums.

I ran my algorithm on a 64-bit unix system on a Nvidia GTX 1070 mobile. I also ran on a 64-bit unix system on a Nvidia GTX 1080. This report is an overview of my experiments, including how the model is set up, what parameters I used, some charts, and what I think would make the algorithm better.

2 Experimentation

2.1 Set of Experiments

I ran the code experimenting with different values for some of the parameters. Specifically, I played with the amount of frames to run through before learning, the rate of epsilon decay, and the max length of the mini batch.

2.2 What structure you experimented with

My code has 3 layers with around 1.5 million neurons. This was just a basic implementation of a Convolution Neural Network and should be improved upon in the future.

2.3 Hyper-parameters

As mentioned above, I played around with the frames before learning, epsilon decay, and the size of the mini batch. All of the values used in experimentation were only used for a brief period of time, not enough time to realize any learning. Most of my changes were necessitated from reading forums or from realizing errors, some even came from intuition about how the model would be running.

2.4 loss function

I used Smooth L1 Huber loss for my model. I used this because it is a robust loss function. I did notice that this is less influenced by outliers than other loss functions. I would think choosing a loss function will be a design choice relative to the amount of sensitivity required.

3 Special Skills

Unfortunately I did not have time to implement any of the tips offered in *Rainbow: Combining Improvements in Reinforcement Learning*. I will continue to work on this algorithm and possibly implement some of the suggestions.

4 Visualization

