# CSCI-4800/5800-003: AI with Reinforcement Learning

# Term Project Report

# Project Title:

OpenAI Gym Space Invaders Agents

# Team members:

Ian Anderson

Problem Statement / Tasks of the project:

The goal of this project is to use the reinforcement learning techniques that we have discussed in class in order to train an agent to play the Space invaders (image based) game and maximize its score using the environment provided by the OpenAI gym.

# Methodology:

The overall method that I used to tackle the problem is Deep Q learning. When combined with a convolutional neural network and the pyTorch library, I was able to do Q evaluations at each step (steps consist of a 3 frame transition) and then store each evaluation into a dictionary.

RMSprop is an optimizer function that is used to calculate the weights for each step. The implementation here takes the square root of the gradient average before adding epsilon (note that TensorFlow interchanges these two operations). The effective learning rate is thus where**α** is the scheduled learning rate and **v** is the weighted moving average of the squared gradient. The algorithm originates from G. Hinton [1] [2]. Afterwards the mean squared average is used to calculate the loss function.

.After enough training the agent exhibits improved gameplay patterns that allow it to consistently achieve higher scores.

# Evaluation:

I used the in-game score to evaluate my progress. Throughout my testing I kept track of a running average of scores from the past 10 episodes. If the running average was consistently getting higher that previous episodes, then I concluded a test as a success. Orignally my goal was to have the agent get past the first level, however this did not happen although the closest I got was a score of 960 with only 4 “space invaders” remaining.

# Experiment Results and Discussion:

An early test where epsilon depreciation was too slow

Chart, scatter chart

Description automatically generated

Chart, line chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

The last image shows the largest test run that I performed that had 150 episodes. As you can see there are still problems with my algorithm and more fine tuning is needed.

# Prior/Related works around the universe:

Phil Tabor’s guides on reinforcement learning using OpenAI gym really helped me get started on the right foot his github repository can be found here <https://github.com/philtabo>.

This video helped me understand how to implement Q learning on OpenAi gym:

https://www.youtube.com/watch?v=wN3rxIKmMgE&list=PLX1ZmajyPhv31rKKdSpGKeO0ug3FFZcoE&index=1&t=25s

# Limitations and Future work

My biggest limitation seemed to be my GPU not being to handle tests larger that 50 episodes at once as this would cause overheating. I would like to figure out a way to save my state dictionaries from one test to carry them over to the next so that I could mitigate this issue. I could not get the agent to converge during my testing and I believe that if I was able to run larger tests that I could eventually get the agent to beat the level consistently.

*.*

# List of References

*(*Enumerated list of references you cite in the project report. If you add a figure in the report that you copied from some website, put a number, say [1] at the caption of the figure and add a reference entry [12] here in this section containing the web address.*)*

[1] https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\_slides\_lec6.pdf

[2] https://arxiv.org/pdf/1308.0850v5.pdf