# A guide to setting up the machine learning algorithms and environments

Author: Jake Ward

## 1 Introduction

This guide will describe the process required to get neural nets trained on your machine using various machine learning algorithms on various environments. The guide assumes you have python installed on an anaconda environment on your system. By the end of this guide, you should be able to run the `Main.py` python script found in the `Machine_Learning` folder in the GitHub repository. This will then provide you with the options necessary to selected a machine learning algorithm, followed by an environment. It should then run and, once complete, allow you to save your neural net for future replaying or for use on the Nao robot. The following machine learning algorithms are available:

- Deep Q-Learning
- Simple Evolutionary
- NEAT (neat-python)
- NEAT (Sam)
- GNARL

The following environments are available:

- OpenAI Gym CartPole-v0
- Pymunk
- 3D Unity

Note that the 3D Unity environment must be used with care, as it utilises key presses that can temporarily lock up your computer.

## 2 Prerequisites

Various python modules / packages must first be installed to allow you to run the machine learning / simulation files. These can be installed with the following command:

```
pip install theano keras tqdm numpy matplotlib pandas pymunk pyglet pynput
```

Note if you have difficulty installing pymunk, please see section 3 of the pymunk guide found in the `Machine_Learning/Guides` folder on GitHub. If any of the modules mentioned above are already installed then you may omit them from the command.

# 3 Keras

In the past we have found difficulty installing TensorFlow and hence using it as a backend for Keras. Furthermore, it was discovered that the Theano module provides a more suitable Keras backend for our needs. However you must change the `load_backend.py` file to use the Theano backend instead of the TensorFlow backend. If you attempt to run the `Main.py` python file, and then enter `1` to select deep Q, you will get an error referring to the `load_backend.py` file. In this error will include the path to this file. Navigate to this path and edit the file. You must set `_BACKEND = ‘theano’`. Note you may also need to copy this line and paste it just before the `if` statements that reference the `_BACKEND` variable. After this, Keras should use the Theano backend and you should be able to successfully run all machine learning scripts.

# 4 Unity

By this point, the cartpole and pymunk environments should function correctly. However the Unity environment utilises key presses to perform actions and reading / writing from a `state.txt` file found in the `Machine_Learning` folder to update the ML algorithms on the current state of the system. As a result of this, you must edit a line in the Unity script to update the path of this state file. This will allow the Unity engine to constantly overwrite this file, allowing the python scripts to read the states from it. Note that it will not be obvious this is a problem until you realise the machine learning algorithms not improving, as they believe the system is unchanging despite performing actions. The file you must edit is found at `Simulation/OSit/Assets/CustomRotationLeft.cs`. You must edit line 371 to use the path to the text file: `Machine_Learning/state.txt`. This path must be correct for your computer. Once done you can test it is working by moving the Unity simulation using the keyboard and checking the `state.txt` file to check it is updating the values. You should now be able to train machine learning algorithms on the Unity simulation.