MecClearen AI Robotics

AIM OF THE PROJECT

This project aims to explore and learn about Artificial Intelligence (AI). These include the Artificial Neural Networks, Deep Q Learning, the different processes of the neural networks, equations involved in machine learning and Python AI libraries (such as PyTorch). Also, this project gives an opportunity to extend my knowledge of Python, which is a powerful script-based programming language.

AIM OF THE PROGRAM

The aim of this program is to successfully train the 'brain' of a robot so that the robot can accomplish round trips from the plastic debris in the oceans to the rubbish collection point on land. The robot must be able to avoid the colonies of fish and rock piles in the ocean.

WHAT THIS PROGRAM DOES

The program demonstrates Deep Q Neural Network, which is a branch of computer science. The program creates an untrained robot which the player will need to train to make it an intelligent, self-operating cleaner. The player can randomly draw piles of rock and fish colonies to create obstacles that the AI will learn to avoid. This program simulates an ocean cleaning robot that does not require human effort, except in the training stage.

THE INTENDED AUDIENCE FOR THIS PROGRAM

The intended audience is people over 13 years of age with interest in information technology so that they can understand the code and logic behind the scenes, but it is suited for children over 9 years of age if they want to run and play with the program for fun.

HOW TO START AND RUN THIS PROGRAM (INSTRUCTIONS)

To start the program, the player must first install Python v2.7, PyTorch, Anaconda, Spyder, Kivy. Please refer to the Appendix - Software Setup Guide. Source code available at: https://github.com/clarenceantonmeryl/MecClearen
Video available at: https://youtu.be/f4rPwyNQfCg

There are six buttons on the map or 'ocean'. At first the robot will start exploring the ocean and start to figure out its path for the round trips. The trainer can select rock or fish and draw random obstacles on the map. The robot will learn to avoid those obstacles over time.

The trainer can press the 'Save Brain' to save the trained brain and to see a graph showing the positive and negative rewards over time. They can analyse this graph to see where the AI made a mistake and whether it learned from it (Note: the maximum reward is positive-0.1). They can also press the 'Load Brain' button to load in their previously saved brain. The 'Clear Map' button clears all the drawings. They can press on 'Score' to get the AI's current score on the console. Also, players can play around with the different values of the speed and softmax temperature by adjusting the sliders to analyse how they affect the robot.

EXPLANATION OF THE SCIENTIFIC CONCEPT WHICH IS SHOWN IN THE PROGRAM

This program is composed of a simple Deep Q Learning Neural Network. There are three main layers to this neural network. The first layer is the input signal, which indicates the amount or density of the obstacles. Then there are hidden neuron layers which connect to the input section with 'weights'. The weights are a measure of how important the input signals are. Finally, the last hidden layer connects to the output layer, which is the action the AI takes. This is called forward-propagation. Then the loss of the action, whether the AI is in a positive or negative state is calculated and is back-propagated through the neural network to adjust the weights. Once the weights are effective, and the AI can overcome the obstacles, it can be considered as 'trained'.

APPENDIX

Software Setup Guide

Anaconda for Python 2.7
(https://www.anaconda.com/products/individual)

Kivy

```
conda install -c conda-forge kivy
```

PyTorch

For Mac & Linux:

```
conda install pytorch==0.3.1 -c pytorch
```

For Windows:

```
conda install -c peterjc123 pytorch-cpu
```

Spyder 3
Launched from Anaconda Navigator

Source code: Download the source code from the GitHub Repository
https://github.com/clarenceantonmeryl/MecClearen

Once downloaded, open all the files in Sypder and open "ocean.py" and press run to execute the program.

BIBLIOGRAPHY

W3schools.com. 2020. *Python Tutorial*.
[online] Available at: <https://www.w3schools.com/python/>
[Accessed between January 2020 and June 2020].

Pythonprogramming.net. 2020. *Python Programming Tutorials*.
[online] Available at: <https://pythonprogramming.net/introduction-deep-learning-neural-network-pytorch/>
[Accessed between March 2020 and July 2020].

Bellman Equation Basics for Reinforcement Learning.
[online] Available at: <https://www.youtube.com/watch?v=14BfO5lMiuk>,
<https://www.youtube.com/watch?v=aNuOLwojyfg>, <https://www.youtube.com/watch?v=aAkFtRxeP7c>
[Accessed between March 2020 and April 2020].

Kivy.org. 2020. *Crash Course — Kivy*
[online] Available at: <https://kivy.org/doc/stable/tutorials/crashcourse.html>
[Accessed between February 2020 and July 2020].

Numpy.org. 2020. Quickstart Tutorial — Numpy Manual.
[online] Available at: <https://numpy.org/devdocs/user/quickstart.html>
[Accessed between January 2020 and July 2020].

Brownlee, J., 2020. A Gentle Introduction To Tensors For Machine Learning With Numpy.
[online] Machine Learning Mastery.
Available at: <https://machinelearningmastery.com/introduction-to-tensors-for-machine-learning/>
[Accessed between March 2020 and May 2020].

Kivy.org. 2020. Game Tutorial — Kivy 1.11.1 Documentation.
[online] Available at: <https://kivy.org/doc/stable/tutorials/pong.html>
[Accessed between March 2020 and April 2020].

Images in the video from iStock photos and Shutterstock
[online] Available at: <https://www.istockphoto.com/> and <https://www.shutterstock.com/>
[Accessed between June 2020 and July 2020].

ACKNOWLEDGEMENTS

I express my sincere thanks to my guide and mentor, Mr Gary Shiell, for guiding me through the guidelines and submissions.

I am deeply grateful to my school IT teacher, Mr Damien Meunier, for introducing me to Python and encouraged me to pursue coding.

I would like to thank my parents for encouraging me along the way.

Finally, I am thankful to the creators of all of the Python libraries I used, especially Kivy and PyTorch, for making the coding process much easier.