Machine Learning

Slither

Summary: Design your own network

Contents

Chapter I

Foreword

Meh

Chapter II

Introduction

Now that you have an understanding of how to use the tools at your disposal it is time to master them. It Will be your job to decide what information is relevant to your network and what decisions it should be able to make.

You will have to design the entire system from the ground up, and it is up to you to train your model and test it.

Chapter III

Goals

The goal of slither is to be able to decipher what are relevant inputs and outputs for a neural network. What does your network need to know in order to learn most efficiently. How will you structure it to manipulate the data into something useful? Lastly, How will you test how well your model is doing in order to train it up.

Chapter IV

General Instructions

You will be given a database of clothing and you will need to create a neural network that, after some training, will be able to identify what types of clothing are in any given image with some accuracy.

Your program will be written in the latest version of Python3.

Allowed Python libraries are:

- Tensorflow
- Any library that allows you to draw graphics for the game

Your work will be tested with a test database of images during corrections. You may not use this database to train your images. You may only use it to test your trained model.

Chapter V

Mandatory Part

V.1 The Game

Create a program that will run a game of snake with one of your models controlling it. It will take one argument which will be the model's file which it will load.

The game must:
- Open a window where you can watch the game play out.
- Play out on a 20 x 20 grid
- Food must spawn on the map in random locations.
- Be able to speed up so that the simulation can be run faster then you press the right arrow key
- Be able to slow down again when you press the left arrow key
- Every frame the snake moves forward one block in whatever direction it is going
- If the snake moves onto a piece of food it grows by one.
- When the snake eats food another piece of food spawns. The food cannot spawn on top of the snake. If there is no space left the snake wins, and the program ends.
- When the snake dies you should print out the length of the snake and end the program
- The snake cannot go off the screen. If it does, it dies.
- The snake cannot run into its tail. If it does, it dies.
- The snake cannot go more than 100 frames without eating food or else it dies.

V.2 The Snake

Use whatever tools you have to create a train a model to be loaded into your snake game. It is up to you to decide what information the model gets from the game and how it makes decisions.

You may structure you game program however you like to provide your model with the information it needs. And it can intemperate the output of the model however you see fit.

Chapter VI

Bonus Part

VI.1 Showing off

Create a model that attempts to do something over then win. Perhaps it attempts to form a letter out of its body or to reach a specific length then die.

VI.2 Easy on the eyes

Make your game look good.

Chapter VII

Turn-in and peer-evaluation

Turn in your work using your GiT repository, as usual. Only the work that's in your repository will be graded during the evaluation.

Your game will be in a folder called “Game” and will start in a file called ‘main.py’ within that directory.

All models must be stored in a folder called “models”

You must also push a diagram which shows the structure of your neural network.

You may not make your snake move on a predetermined path! It must think for itself!

During defence you must be able to show that your model is making the decisions and that you haven't just coded the AI into the game itself.