Research

Depending on Mr Christoph email, using notebooks may be a problem in the professional life because Google colab manage all our dependencies and resolve the issues encountred.

From here, I started to ask, since I'm used to deal only with notebooks, why developping in vscode (or any IDE) is better than notebooks ? why should I Switch from Jupyter Notebook to Scripts ?

Answers I found : based on preference data from user reviews. The Jupyter Notebook rates 4.5/5 stars with 200 reviews. By contrast, **Visual Studio rates 4.5/5 stars with 3,215 reviews**.

- Ideal for Reproducibility. With classes and functions, we could make the code general enough so that it will be able to work with other data.
- Easy to Debug.
- Ideal for Production

References :

Google scholar

Google

Stackoverflow

GeeksForGeeks

Pytorch official website

Stable baselines3 github repository

Solution and Algorithm explanation

- Create an AI dedicated in hiding
- The AI would hide from a turret placed at the center (n=0) of the map with the most realistic behavior. Indeed, the AI needs to hide to the shortest hideout.

➔ Starting from **n=0 (center),** the algorithm will :

1. Calculate the length of all placements possibilities to hide
2. Choose the minimum length because we need the shortest hideout
3. The agent will be awarded if it **hides with the shortes hideout (hide=True and hideout=min) else it will receive a penalty**

Basics :

Is Q-learning a deep learning?

Deep Q Learning uses the Q-learning idea and takes it one step further. Instead of using a Q-table, we use a Neural Network that takes a state and approximates the Q-values for each action based on that state.

alpha: is the learning rate, set generally between 0 and 1. Setting the alpha value to 0 means that the Q-values are never updated, thereby nothing is learned. If we set the alpha to a high value such as 0.9, it means that the learning can occur quickly.

Gamma : varies from 0 to 1. If Gamma is closer to zero, the agent will tend to consider only immediate rewards. If Gamma is closer to one, the agent will consider future rewards with greater weight, willing to delay the reward. Max[Q(s', A)] gives a maximum value of Q for all possible actions in the next state.

Gym provides different game environments which we can plug into our code and test an agent. The library takes care of API for providing all the information that our agent would require, like possible actions, score, and current state. We just need to focus just on the algorithm part for our agent.

Pygame Snippets is a vscode snippet that makes it easy to create games in python using pygame.

Stable Baselines3 (SB3) is a set of reliable implementations of reinforcement learning algorithms in PyTorch. It is the next major version of Stable Baselines.

Proximal Policy Optimization, or PPO, is a policy gradient method for reinforcement learning. The motivation was to have an algorithm with the data efficiency and reliable performance of TRPO, while using only first-order optimization.

PPO algorithm is a deep reinforcement learning algorithm with outstanding performance, especially in continuous control tasks. But the performance of this method is still affected by its exploration ability

It can be observed that PPO provides a better convergence and performance rate than other techniques but is sensitive to changes. DQN alone is unstable and gives poor convergence, hence requires several add-ons.

The differences between tabular Q-learning (TQL), deep Q-learning (DQL), and deep Q-network (DQN) :

	Tabular Q-learning (TQL)	**Deep Q-learning (DQL)**	**Deep Q-network (DQN)**
Is it an RL algorithm?	Yes	Yes	No (unless you use DQN to refer to DQL, which is done often!)
Does it use neural networks?	No. It uses a table.	Yes	No. DQN is the neural network.
Is it a model?	No	No	Yes (but usually not in the RL sense)
Can it deal with continuous state spaces?	No (unless you discretize them)	Yes	Yes (in the sense that it can get real-valued inputs for the states)
Can it deal with continuous action spaces?	Yes (but maybe not a good idea)	Yes (but maybe not a good idea)	Yes (but only the sense that it can produce real-valued outputs for actions).
Does it converge?	Yes	Not necessarily	Not necessarily
Is it an online learning algorithm?	Yes	No, if you use experience replay	No, but it can be used in an online learning setting

[illegible]

Hadil BEN AMOR

10/30/2022