CMSC 409: Artificial Intelligence

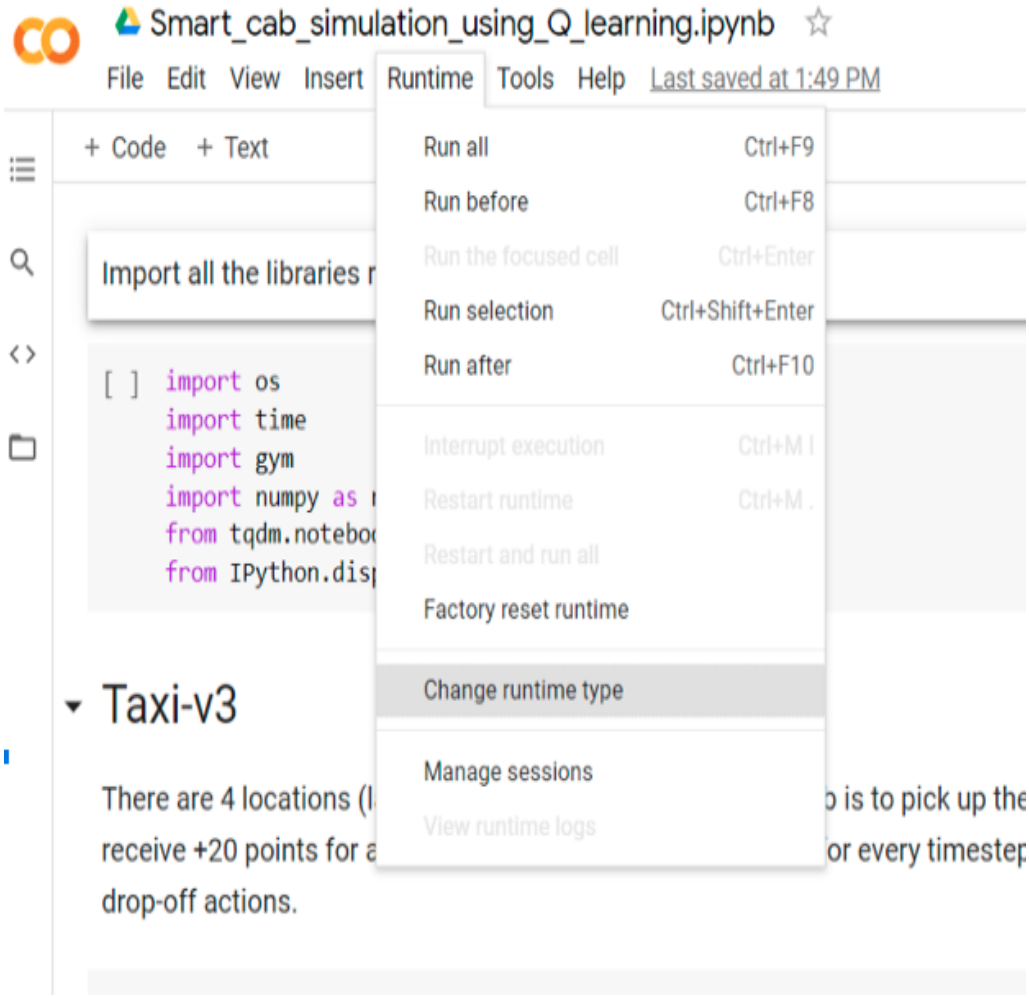
Fall 2023, Instructor: Dr. Milos Manic, http://www.people.vcu.edu/~mmanic

Project 3 - INSTRUCTIONS

Instructions to run the code:

Upload the two notebook files "*Smart_cab_brute_force.ipynb"* and "*Smart_cab_simulation_using_Q_learning.ipynb*" to your Google Drive, then open those notebook files using Google Colab (right mouse click).

Go to the Runtime tab → change run time type

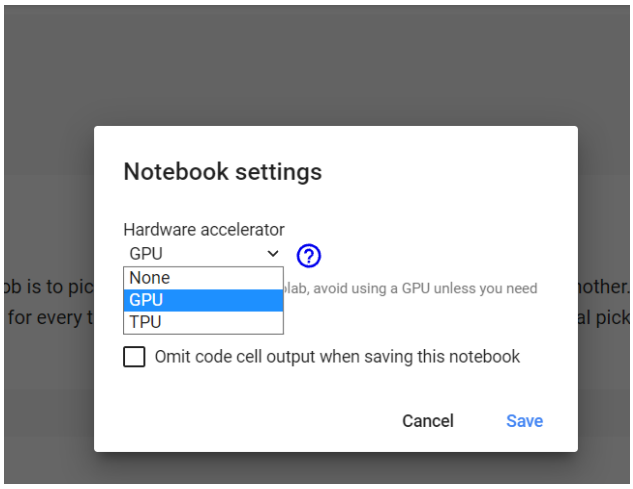


Go to the Hardware accelerator → GPU/TPU (This option is to accelerate the training process).

CMSC 409: Artificial Intelligence

Fall 2023, Instructor: Dr. Milos Manic, http://www.people.vcu.edu/~mmanic

Project 3 - INSTRUCTIONS



Click Connect button (on the top right) to start a server instance

Read carefully the explanation given within each cell, and run each cell first with original parameters (you will change them later).

$$Q(state, action) \leftarrow (1-\alpha)Q(state, action) + \alpha(reward + \gamma max_a Q(nextstate, allactions))$$

Where:

- α (alpha) is the learning rate ($0 < \alpha \leq 1$) - Just like in supervised learning settings, α is the extent to which our Q-values are being updated in every iteration.
- γ (gamma) is the discount factor ($0 \leq \gamma \leq 1$) - determines how much importance we want to give to future rewards. A high value for the discount factor (close to 1) captures the long-term effective award, whereas, a discount factor of 0 makes our agent consider only immediate reward, hence making it greedy.

```
[ ]  q_table = np.zeros([env.observation_space.n, env.action_space.n])
```

Start training the Q Learning model

```
# Hyperparameters
alpha = 0.1
gamma = 0.99
epsilon = 0.1
```