Guide to rebuilding the Project:

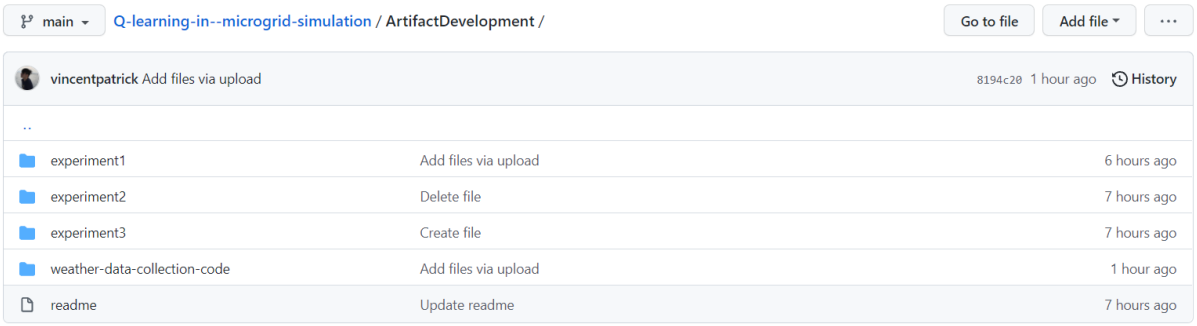
By: Vincenntius Patrick Tunas

In this document, steps on how the project can be rebuild is shown.

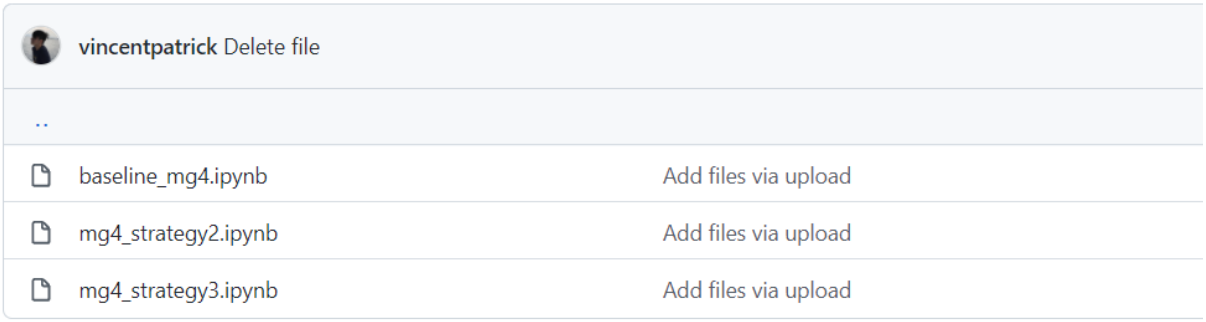
Step1: choosing the experiment and the strategy to be run

Click on the artefact developed section in the github link: https://github.com/vincentpatrick/Q-learning-in--microgrid-simulation.git

There are 3 experiments to choose from



Inside each experiments, there are strategies to choose from. Simply download the strategy that you want to use.

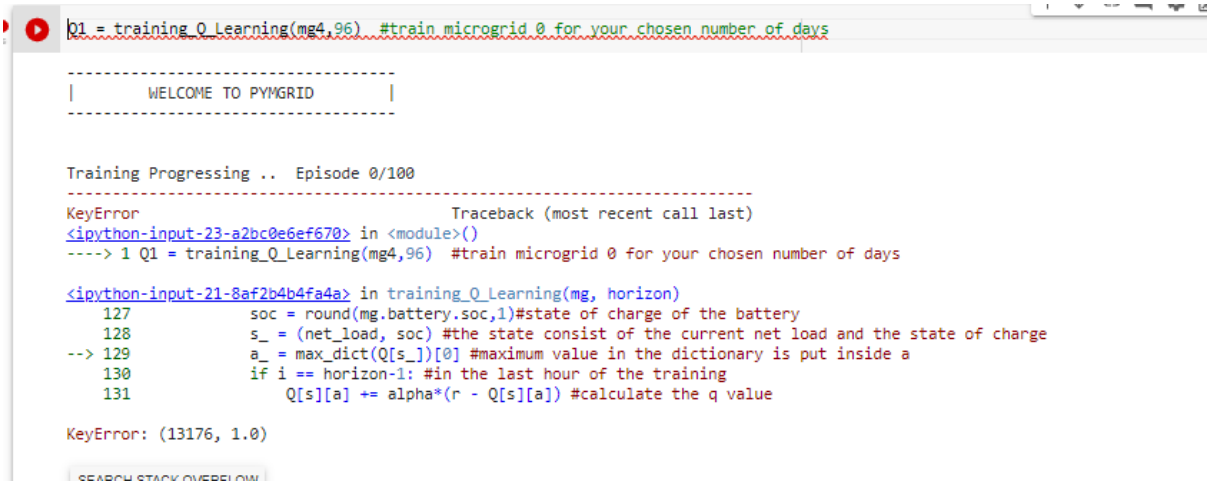


Step 2: running the notebook in colab

This python notebook can be run inside google colab. Simply run the notebook inside google colab.

Every day, the dataset changes when using the pymgrid package. Therefore, when you run the algorithm you may not receive the same result as me.

During training section, if you encounter this error, you need to lower down the number of horizon that is set in the agents training.



```
Q1 = training_Q_Learning(mg4,96)  #train microgrid 0 for your chosen number of days

-------------------------------------
|        WELCOME TO PYMGRID         |
-------------------------------------


Training Progressing ..  Episode 0/100
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-23-a2bc0e6ef670> in <module>()
----> 1 Q1 = training_Q_Learning(mg4,96)  #train microgrid 0 for your chosen number of days

<ipython-input-21-8af2b4b4fa4a> in training_Q_Learning(mg, horizon)
    127             soc = round(mg.battery.soc,1)#state of charge of the battery
    128             s_ = (net_load, soc) #the state consist of the current net load and the state of charge
--> 129             a_ = max_dict(Q[s_])[0] #maximum value in the dictionary is put inside a
    130             if i == horizon-1: #in the last hour of the training
    131                 Q[s][a] += alpha*(r - Q[s][a]) #calculate the q value

KeyError: (13176, 1.0)
```

The number of horizons is at the right parameter of the training_Q_learning function.

By Increasing the number of episodes for the agent to train can improve the agent's performance. The number of episodes can be changed inside the training function. The name of the variable is nb_episode

```
def training_Q_Learning(mg,horizon):

    #initialize variables
    nb_action = 4
    Q = init_qtable(mg,nb_action)
    nb_state = len(Q)
    nb_episode = 100
    alpha = 0.1
    epsilon = 0.99
    gamma = 0.99
```