Baseline Model vs current Q-Learning model

1. Time-series data

1.1 explanation for randomness of data

as we know Pymgrid data is randomly generated but it doesn't mean that it is unpredictable.

When the algorithm is runned. It will always start with the time series data that we are using for this experiment.

In order to change the time series data that we want to use simply reload the code for forecasting pv and load.

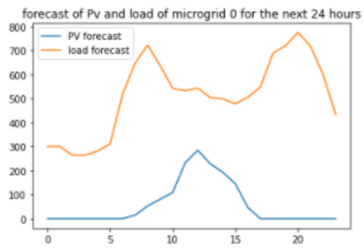
If we want to go back to the old dataset, simply restart the runtime.

1.2 data forecast

The current data forecast in (8:16 pm, 2/09/2021)

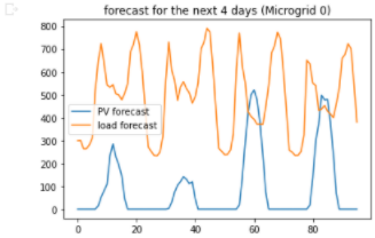
forecast for next 24 hours

```
[9] plt.title("forecast of Pv and load of microgrid 0 for the next 24 hours")
    plt.plot(mg0.forecast_pv(), label ="PV forecast")
    plt.plot(mg0.forecast_load(), label = "load forecast")
    plt.legend()
    plt.show()
```



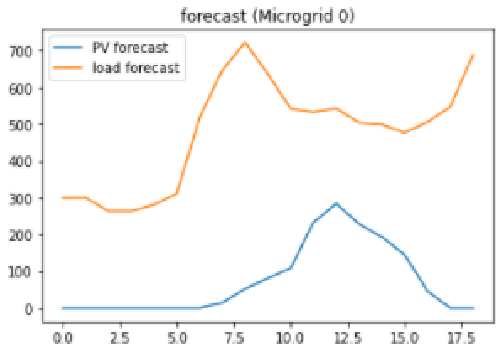
forecast for next 4 days

```
[12] plt.title("forecast for the next 4 days (Microgrid 0)")
     #change the horizon time
     mg0.set_horizon(24*4)
     #dont forget to change back the horizon at the right value afterward
     plt.plot(mg0.forecast_pv(), label ="PV forecast")
     plt.plot(mg0.forecast_load(), label ="load forecast")
     plt.legend()
     plt.show()
```

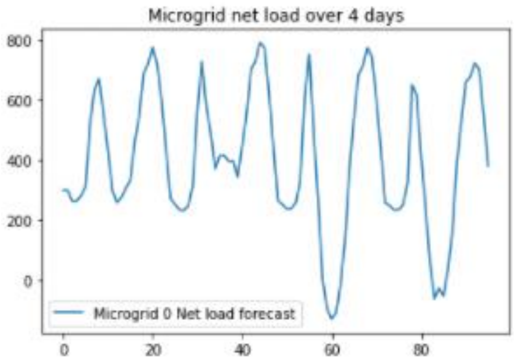


The data seems consistent with its sunrise peak hour time (the time where PV forecast is high).

The hours of pv availability are from 7am up to 5pm.



The net load forecast of the microgrid for 4 days



2. Baseline experiment

I conducted an experiment to determine a proper baseline for the model.

First, we assume, since the sunlight is only present during daytime, PV source will only be available at a specific time in the day. Based on the forecast, the PV is only available and receivable from 7AM up to 5PM.

Therefore, there are 2 experiment we will use for our baseline. We will also make a comparison with the original model.

1. Experiment 1: During the training, the agent charges into the battery during the hours where PV source is available (7am to 5pm). However, during the testing stage, the change in Q-table values because of this decision during training has an impact to the decision that the agent will pick in the next state.
2. Experiment 2: The agent charges to the battery during pv availability hours absolutely without considering the consequences. We demand the agent to have blind obedience.
3. Original Model: the agent will run normally and make its decision the normal way.

So now let the experiment begin!!

3. Baseline model

the experiment will be run in the simulation for 1 day

Original model

In the original model, we do not interfere with the agents training, forcing it to pick the action to charge into our battery from 7am to 5pm. The result is as below:

```
testing_Q_Learning(mg0,Q1, 24) #start q learning for 2 days
```

```
t -     STATE   -  ACTION - COST
================================
0  - (299, 0.2) discharge 73.0 €
1  - (300, 0.2) import 147.2 €
2  - (264, 0.2) import 212.4 €
3  - (264, 0.2) import 277.4 €
4  - (281, 0.2) import 346.7 €
5  - (311, 0.2) import 423.0 €
6  - (518, 0.2) discharge 550.8 €
7  - (633, 0.2) import 707.5 €
8  - (670, 0.2) discharge 919.8 €
9  - (556, 0.2) import 1095.8 €
10 - (434, 0.2) discharge 1233.2 €
11 - (300, 0.2) import 1328.2 €
12 - (259, 0.2) import 1487.9 €
13 - (276, 0.2) discharge 1658.0 €
14 - (306, 0.2) discharge 1846.7 €
15 - (333, 0.2) discharge 2051.9 €
16 - (459, 0.2) import 2333.8 €
17 - (548, 0.2) discharge 2666.6 €
18 - (689, 0.2) discharge 2877.0 €
19 - (720, 0.2) discharge 3096.4 €
20 - (775, 0.2) discharge 3332.6 €
21 - (718, 0.2) discharge 3501.2 €
22 - (600, 0.2) discharge 3642.7 €
23 - (434, 0.2) discharge 3746.2 €
```

the agent used up to 3746.2 euro for cost

Experiment 1

When experiment 1 is initiated for the agent's training, the output is as below:

```
testing_Q_Learning(mg0,Q1, 24)

t -     STATE  -  ACTION - COST
================================
0  - (299, 0.2) import 73.0 €
1  - (300, 0.2) discharge 147.2 €
2  - (264, 0.2) import 212.4 €
3  - (264, 0.2) import 277.4 €
4  - (281, 0.2) import 346.7 €
5  - (311, 0.2) import 423.0 €
6  - (518, 0.2) discharge 550.8 €
7  - (633, 0.2) export 13213.5 €
8  - (670, 0.2) charge 24225.1 €
9  - (556, 0.4) charge 34092.0 €
10 - (434, 0.7) export 42765.6 €
11 - (300, 0.7) charge 48766.0 €
12 - (259, 0.8) charge 53949.7 €
13 - (276, 0.9) charge 57792.4 €
14 - (306, 1.0) discharge 57798.6 €
15 - (333, 0.8) charge 64459.3 €
16 - (459, 1.0) discharge 64485.9 €
17 - (548, 0.7) charge 74270.7 €
18 - (689, 0.9) charge 82620.4 €
19 - (720, 1.0) discharge 82717.3 €
20 - (775, 0.7) discharge 82831.0 €
21 - (718, 0.4) import 82999.7 €
22 - (600, 0.2) import 83141.2 €
23 - (434, 0.4) discharge 83162.3 €
```

the agent used up to 83162.3 euro for cost

Experiment 2

In experiment 2. The agent is blindly obedient to us. It will only charge electricity into the battery from 7AM to 5pm. In this version, the agent will not choose anything else

```
[34] testing_Q_Learning(mg0,Q1, 24)

t -     STATE  -  ACTION - COST
================================
0  - (299, 0.2) import 73.0 €
1  - (300, 0.2) import 147.2 €
2  - (264, 0.2) import 212.4 €
3  - (264, 0.2) import 277.4 €
4  - (281, 0.2) import 346.7 €
5  - (311, 0.2) import 423.0 €
6  - (518, 0.2) import 550.8 €
7  - (633, 0.2) charge 11190.8 €
8  - (670, 0.4) charge 22202.3 €
9  - (556, 0.7) charge 32069.2 €
10 - (434, 0.9) charge 38781.9 €
11 - (300, 1.0) charge 41779.1 €
12 - (259, 1.0) charge 44368.3 €
13 - (276, 1.0) charge 47124.3 €
14 - (306, 1.0) charge 50183.1 €
15 - (333, 1.0) charge 53510.1 €
16 - (459, 1.0) charge 58104.2 €
17 - (548, 1.0) charge 63580.4 €
18 - (689, 1.0) discharge 63668.1 €
19 - (720, 0.7) discharge 63765.0 €
20 - (775, 0.4) discharge 63893.9 €
21 - (718, 0.2) discharge 64062.5 €
22 - (600, 0.2) discharge 64204.0 €
23 - (434, 0.2) import 64307.5 €
```

The agent used up to 64307.5 euro

4. Experiment conclusion

Based on the results above,

it seems that the agent never chooses to charge into the battery at all in the original model. even when it is run in the simulation for 4 days.

This shows that maybe the microgrid data generated by pymgrid here has a high electricity inside the battery and we can use a microgrid with a different dataset in the next run.

The result also show that experiment 2 is able to save more cost than experiment 1. We can assume the reason why is because the agent commits fully with the strategy to charge during daytime

Next step

There are many places where we can improve our experiment

1. A different dataset for the microgrid
2. Use different microgrid architecture:
   - Try working without a grid
   - Use a weak grid
3. Modify the baseline so that it can be run for a longer number of days (approximately 4 days)