Q-learning Baseline

Objective:

We want to use reinforcement learning to automatically learn how to control the microgrid:

- Generation
- Storage
- Consumption

Data:

- the load
- the pv generation
- the battery levels
- consumption level
- costing to buy electricity from the grid

the agent's action:

- consume from the grid
- export energy to the grid
- charge pv generation into the battery
- discharge pv generation into the battery

Testing: strategy 1, strategy 2, RL agent

Duration: 4 days - 1 month

1 timestep = 1 hour

Inputs: data above

State →(net load, state of charge)

Actions the agent take during that state

What result do we want from the experiment?

For daily usage of strategy 1&2 and rl , how much would be my bill for the day, costing of using the strategy

Task Breakdown

10 microgrids are being generated. In order to avoid the result to be the same as previous

```
#generate microgrids
env = mg.MicrogridGenerator(nb_microgrid=10)
env.generate_microgrid(verbose= True)
mg0 = env.microgrids[0]
mg1 = env.microgrids[1]
mg2 = env.microgrids[2]
mg3 = env.microgrids[3]
mg4 = env.microgrids[4]
mg5 = env.microgrids[5]
mg6 = env.microgrids[6]
mg9 = env.microgrids[9]
```

	load	cost_loss_load	cost_overgeneration	cost_co2	PV_rated_power
0	960	10	1	0.1	969.60
1	60363	10	1	0.1	30785.13
2	43725	10	1	0.1	54219.00
3	78147	10	1	0.1	38292.03
4	49813	10	1	0.1	73225.11
5	28560	10	1	0.1	30559.20
6	49216	10	1	0.1	53645.44
7	11798	10	1	0.1	10028.30
8	64065	10	1	0.1	34595.10
9	98996	10	1	0.1	136614.48

The potential microgrids to be experimented are:

- microgrid 2: {'PV': 1, 'battery': 1, 'genset': 1, 'grid': 0}
- microgrid 4: {'PV': 1, 'battery': 1, 'genset': 0, 'grid': 1}
- microgrid 5: {'PV': 1, 'battery': 1, 'genset': 1, 'grid': 0}
- microgrid 6: {'PV': 1, 'battery': 1, 'genset': 0, 'grid': 1}

- microgrid 9: `{'PV': 1, 'battery': 1, 'genset': 1, 'grid': 1}`

lets experiment with the same architecture first: microgrid 4 and microgrid 6

microgrid 4: RL

this is the pv generation and load forecast of microgrid 4

when running the simulation for microgrid 4 it seems that the system is not able to run beyond 4 days.

```
testing_Q_Learning(mg4,Q1, 24)
```

```
t -     STATE   -  ACTION - COST
================================
0  - (7926, 0.2) import 824.1 €
1  - (7784, 0.2) import 1659.8 €
2  - (7914, 0.2) import 2506.9 €
3  - (7784, 0.2) import 3334.9 €
4  - (8287, 0.2) discharge 4216.4 €
5  - (8903, 0.2) import 5424.1 €
6  - (10527, 0.2) import 6863.1 €
7  - (9195, 0.2) discharge 8127.0 €
8  - (196, 0.2) import 8153.8 €
9  - (-11138, 0.2) discharge 8153.8 €
10 - (-18508, 0.2) discharge 8153.8 €
11 - (-13489, 0.2) discharge 8153.8 €
12 - (-22013, 0.2) discharge 8153.8 €
13 - (669, 0.2) import 8245.5 €
14 - (-8427, 0.2) discharge 8245.5 €
15 - (1384, 0.2) discharge 8393.3 €
16 - (9316, 0.2) discharge 9358.7 €
17 - (11140, 0.2) import 10781.6 €
18 - (11271, 0.2) discharge 12195.5 €
19 - (10395, 0.2) discharge 13491.9 €
20 - (10578, 0.2) discharge 14811.1 €
21 - (9696, 0.2) import 16023.7 €
22 - (9093, 0.2) discharge 17166.2 €
23 - (8205, 0.2) discharge 18220.2 €
```

```
testing_Q_Learning(mg0,Q1, 48)
```

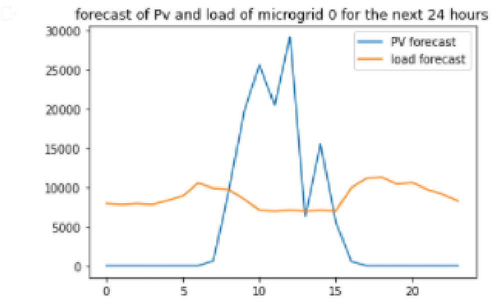
```
t -     STATE   -  ACTION - COST
================================
0  - (299, 0.2) discharge 73.0 €
1  - (300, 0.2) discharge 147.2 €
2  - (264, 0.2) discharge 212.4 €
3  - (264, 0.2) discharge 277.4 €
4  - (281, 0.2) discharge 346.7 €
5  - (311, 0.2) discharge 423.0 €
6  - (518, 0.2) import 550.8 €
7  - (639, 0.2) discharge 708.9 €
8  - (597, 0.2) discharge 898.0 €
9  - (377, 0.2) import 1017.4 €
10 - (203, 0.2) import 1081.9 €
11 - (262, 0.2) discharge 1164.8 €
12 - (158, 0.2) discharge 1262.4 €
13 - (421, 0.2) import 1521.9 €
14 - (293, 0.2) discharge 1703.0 €
15 - (404, 0.2) import 1952.1 €
16 - (498, 0.2) import 2257.7 €
17 - (548, 0.2) discharge 2590.5 €
18 - (689, 0.2) discharge 2800.9 €
19 - (720, 0.2) import 3020.2 €
20 - (775, 0.2) import 3256.4 €
21 - (718, 0.2) import 3425.1 €
22 - (600, 0.2) import 3566.5 €
23 - (434, 0.2) import 3670.0 €
24 - (272, 0.2) import 3736.7 €
25 - (253, 0.2) import 3799.6 €
26 - (235, 0.2) discharge 3858.0 €
27 - (233, 0.2) import 3915.9 €
28 - (249, 0.2) discharge 3977.8 €
29 - (318, 0.2) import 4057.2 €
30 - (565, 0.2) import 4198.3 €
31 - (721, 0.2) discharge 4378.8 €
32 - (463, 0.2) discharge 4527.2 €
33 - (365, 0.2) import 4643.8 €
34 - (162, 0.2) discharge 4695.6 €
35 - (156, 0.2) discharge 4745.5 €
36 - (226, 0.2) import 4885.4 €
37 - (299, 0.2) discharge 5070.6 €
38 - (416, 0.2) discharge 5327.9 €
39 - (384, 0.2) discharge 5564.8 €
40 - (485, 0.2) import 5863.1 €
41 - (551, 0.2) import 6200.1 €
42 - (700, 0.2) discharge 6417.7 €
43 - (729, 0.2) import 6644.7 €
44 - (789, 0.2) import 6889.7 €
45 - (774, 0.2) discharge 7075.6 €
46 - (622, 0.2) discharge 7224.6 €
47 - (445, 0.2) import 7331.9 €
```

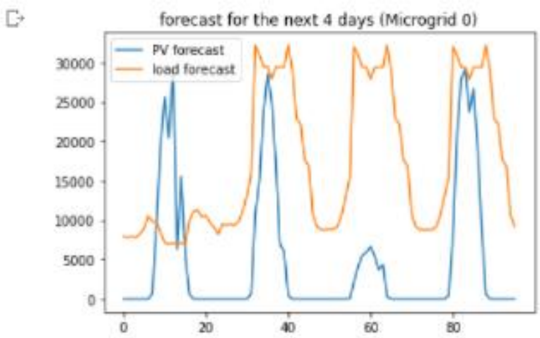
When comparing the performance with microgrid 1, the values that microgrid 4 is dealing with are higher. During the hours where the pv generation peaks so high and is able to satisfy the load, the agent didn't take this chance to charge the energy into the battery causing the system to lose that energy

As you can see in the state for the hours 9 to 12, the negative value indicates that it is able to satisfy the load and it went higher than the load. However the agent didn't choose to charge into the battery causing it to lose energy and the system fails after a day

In system 1, since the values are smaller, it can last for a higher amount of days.

Lets implement our strategy into microgrid 4 and lets see if the system can survive more than a day.



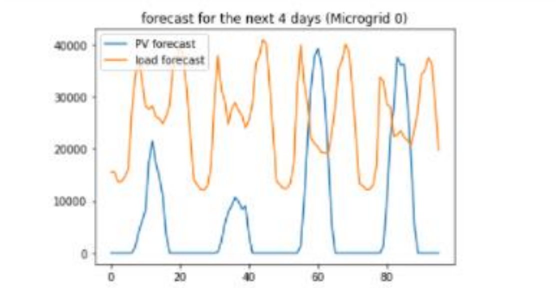


As we can see on the first day the energy were excessive. However in the next 3 days. The load seems to not be able to satisfy the load fully.

- On day 2 and day 4, it seems that the pv consumption is very high but not enough to fulfill the load yet, with the help of energy from the grid and the battery, it may be able to reach and satisfy the goal. Therefore, our next step is to change the strategy just for the first day and lets see how the algorithm will perform.

Microgrid 4: simulate for 4 days

Cost of the strategy for different number of days



The data forecast we are using are above

Cost

	S1	S2	S3	RL
Day1	63273.5 €	4828151.0 €	63273.5 €	63273.5 €
Day2	131322.9 €	9269465.9 €	131322.9 €	131322.9€
Day3	3716709.1 €	11611311.9 €	3727856.€	184391.8 €
Day4	8360144.7 €	12737542.5	8371291.€	234125.8 €
Average				
Total				

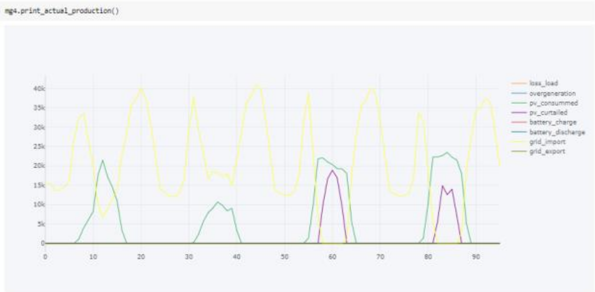
Net load

	S1	S2	S3	RL
Day1				
Day2				
Day3				
Day4				
Average				
Total				

The net load of all the strategies are the same

RL Performance

Actual production

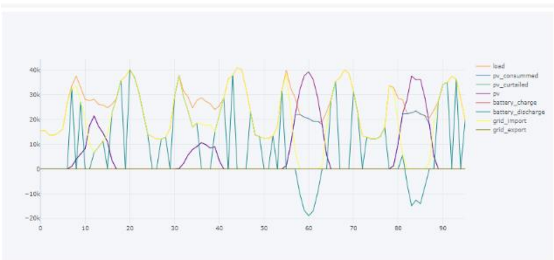


In day 3 and day 4 is the only time the pv generated is high enough and able to satisfy the load.

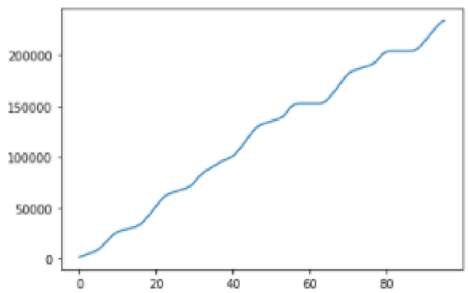
- The purple line shows the amount of pv that is taken in
- The green line shows the amount of pv that is being consumed to satisfy the load.
- The yellow line indicates the amount of energy imported from the grid.

Controls

Now lets see the controls that the agent took during the simulation



In day 3 and day 4, we can see the battery is discharging negative value. This is not good because it shows that the system loses a lot of energy.



It seems that the cost keeps rising up and up over the days.

Microgrid 4: our strategies

We have run the microgrid with a new time-series forecast and based on what we learn, we'll implement a strategy

Strategy 1: change during training

- In this strategy, to avoid energy loss, we will use charging actions every daytime, during the hours where the pv source are at its highest.
- However, because pv generation is available, it doesn't always mean that it is high and if it is worth charging into the battery. Therefore, we only change the agent's training.
- During 100 episodes of training, we'll tell the agent to charge during the hours of high pv generation,
- The hours we choose to charge is from 9am to 4pm everyday
- During testing, based on what the agent has learn from the modified training, the agent can make his own decision

Strategy 3: striving for net load 0

In this strategy, we don't indicate the agents decision based on the time of the day

Instead, we modify the agents action based on the scenarios. In previous experiment, the agent can only do 1 action at 1 time. In this experiment, we combine the actions. We enable the agent to do 2 actions during the same timestep

Action space of the agent:

Action 0: battery charge

Action 1: battery discharge

Action 2: grid import

Action 3: grid export

Scenario 1

IF netload >=0

THEN: Agent's action is **discharging from the battery** and **importing energy from the grid**

Available actions: 1, 2

Scenario 2

IF netload < 0

THEN: Agent's action is to **charge into the battery** and **export energy into the grid**

Available actions: 0,3

For both scenarios, these actions are taken at once in the same timestep.

In this strategy, we are targeting a net load of 0. We want the agent to strive in satisfying a net load of zero

By charging into the battery and exporting energy into the grid when the netload is negative, then we can ensure that unused energy will not become wasted. Therefore, in the long run, it will help us save more money.