



full thesis

Artificial intelligence in the air

25/09/2020 (sep)

- Introduce the network slicing in 5g and 6g also neural methods
- Introduce the nokia wireless suite

Meeting one



Considering INI

1570641355.pdf

File | D:/COURSE/THESIS/AI/1570641355.pdf

beep Dodge karting HTML iotco arsin SPG MOTOR raspberry pi tech rf ic imedss voltage complex android studio avr sweden

1 of 7

Read aloud Draw Highlight Erase

Spectrum Allocation for Network Slices with Inter-Numerology Interference using Deep Reinforcement Learning

Marco Zambianco, Giacomo Verticale
Dipartimento di Elettronica, Informatica e Bioingegneria
Politecnico di Milano – Italy
{marco.zambianco, giacomo.verticale}@polimi.it

Network slicing

Abstract—Network slicing and mixed-numerology schemes are essential technologies to efficiently accommodate different services in 5G radio access networks (RAN). To fully take advantage of these techniques, the design of spectrum slicing policies needs to account for the limited availability of the radio resources as well as the inter-numerology interference generated by slices employing different numerologies. In this context, we formulate a binary non-convex problem that maximizes the aggregate capacity of multiple network slices. The resulting spectrum allocation minimizes the inter-numerology interference under the frequent channel fluctuations characterizing the various users. To address the computational complexity of the designed objective function, we leverage deep reinforcement learning (DRL) to design a model-free solution computation. In detail, the trained centralized DRL agent exploits the channel fading statistic in order to provide a spectrum allocation that minimizes the inter-numerology interference. Results reveal that the proposed DRL scheme achieves performance that is comparable to the optimal one. It also outperforms a baseline scheme that statically allocate the radio resources.

I. INTRODUCTION

RAN slicing is a network feature that makes it possible to deploy multiple independent virtual networks on top of the same physical network infrastructure. Each virtual entity is commonly denoted as “network slice” and has access to blocks (RB), having heterogeneous subcarrier spacing on the same physical layer. On one hand, this access scheme provides the flexibility to accommodate different radio requirements, on the other hand, the loss of orthogonality between RBs of different numerologies generates interference that hinders the transmission performance [2].

For these reasons, the design of effective RAN slicing policies should jointly account for the dynamic quality of the radio resources together with the inter-numerology interference (INI). However, in our opinion, most work has addressed these issues separately and the analysis of their mutual impact has received little attention. Following these observations, we propose a centralized agent-based allocation of mixed-numerology spectrum slices by leveraging deep reinforcement learning (DRL), which has recently found many applications in the field of wireless communications [3] [4]. In detail, the main contributions of this work are:

- We design a binary non-convex optimization problem for the allocation of radio resources to multiple network slices that are multiplexed on a mixed-numerology physical layer. The objective function maximizes the aggregated capacity of each slice by accounting for the

<https://ieeexplore.ieee.org/abstract/document/8476595>



Nokia

ataeiamirhosein/wireless-suite: v x +

https://github.com/ataeiamirhosein/wireless-suite

beep Dodge karting HTML iotco arsin SPG MOTOR raspberry pi tech rf ic imedss voltage complex android studio avr sweden

Search or jump to... Pull requests Issues Marketplace Explore

ataeiamirhosein / wireless-suite
forked from nokia/wireless-suite

Watch 0 Unstar 1 Fork 23

Code Pull requests Actions Projects Wiki Security Insights Settings

master 2 branches 0 tags Go to file Add file Code

This branch is even with nokia:master. Pull request Compare

alvarovalcarce	Merge pull request nokia#12 from shahab1992/Knapsack-Agent	83860cd on Sep 15	33 commits
config	Create the NomaULTimeFreqResourceAllocation-v0 environment	4 months ago	
wireless	Merge pull request nokia#12 from shahab1992/Knapsack-Agent	last month	
LICENSE	Initial commit	7 months ago	
README.md	-Add UMTS OLPC documentation to README file.	2 months ago	
setup.py	-Add tabular Q-learning agent.	2 months ago	

github

About
www.nokia.com
Readme
BSD-3-Clause License

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)



full thesis

Artificial intelligence in the air

09/19/2020 (oct)

- Read the gym and sacred documentation also deep learning
- Run the nokia wireless suite with observer

Meeting **two**



Wireless-suite develop

nokia/wireless-suite

https://github.com/nokia/wireless-suite

beep Dodge karting HTML iotco arsin SPG MOTOR raspberry pi tech rf ic imedss voltage complex android studio avr sweden

Evaluation

The simulated environment can be chosen by setting "env": "TimeFreqResourceAllocation-v0" or "env": "NomaULTimeFreqResourceAllocation-v0" in `config/config_environment.json`. The script `wireless/scripts/launch_agent.py` runs 16 episodes with a maximum of 65536 time steps each, and collects the reward obtained by the agent on each time step. The result is calculated as the average reward obtained in all time steps on all episodes.

How to contribute

There are two main ways of contributing to Wireless Suite:

- Implementing new problems:** This version of Wireless Suite contains two problems implementation. New problems can be easily added as simple variations of the existing ones (e.g. by changing their parameters), or by introducing fully new problem implementations (e.g. Adaptive Modulation and Coding, Open Loop Power Control, Handover optimization, etc).
- Implementing new agents:** Ideally, new agent contributions shall perform better than the default ones.

References

1. [Open AI Gym Documentation](#)

2. [How to create new environments for Gym](#)

3. [Sacred Documentation](#)

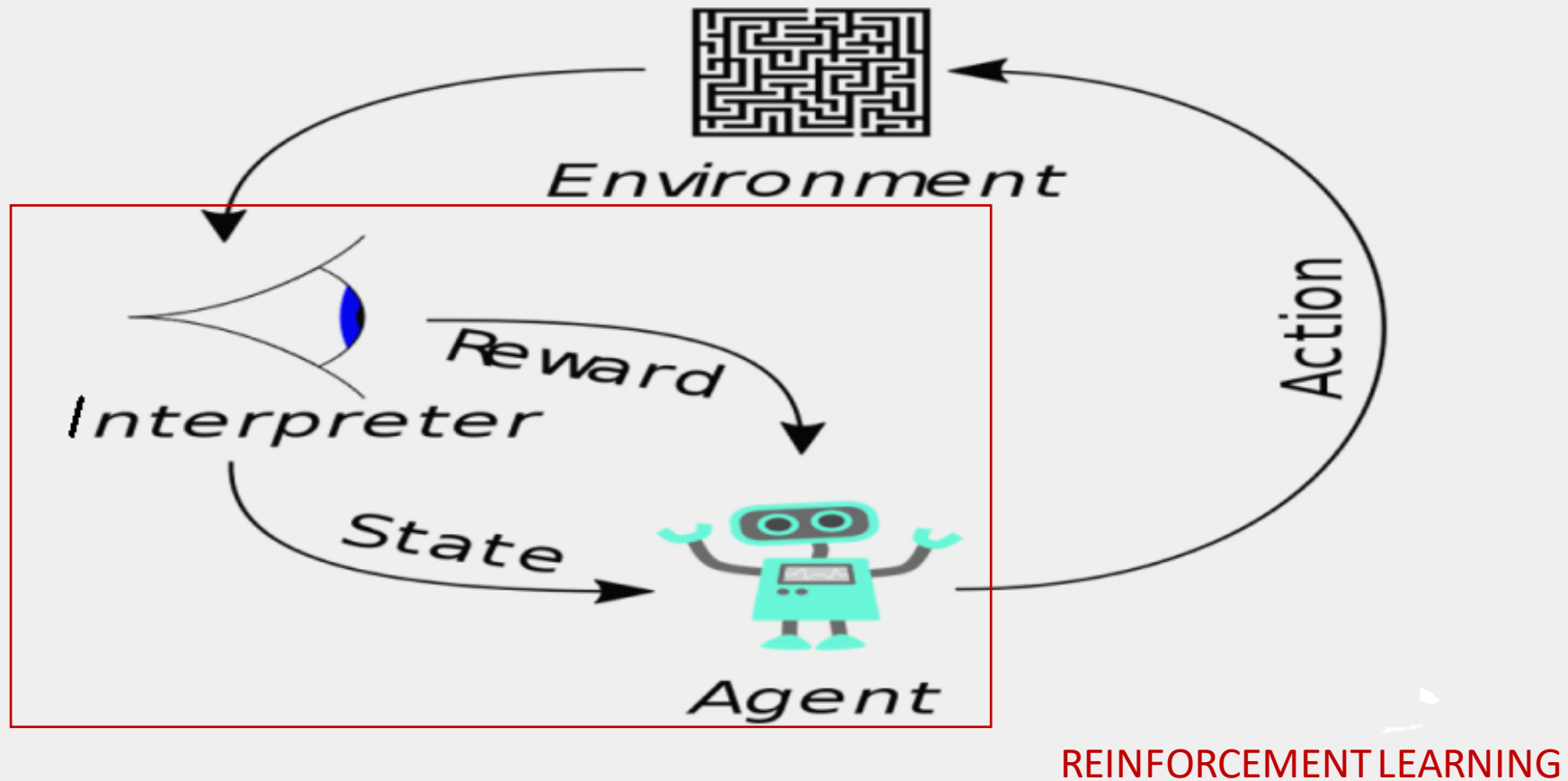
License

This project is licensed under the BSD 3-Clause license - see the [LICENSE](#)

GYM ENVIRONMENT

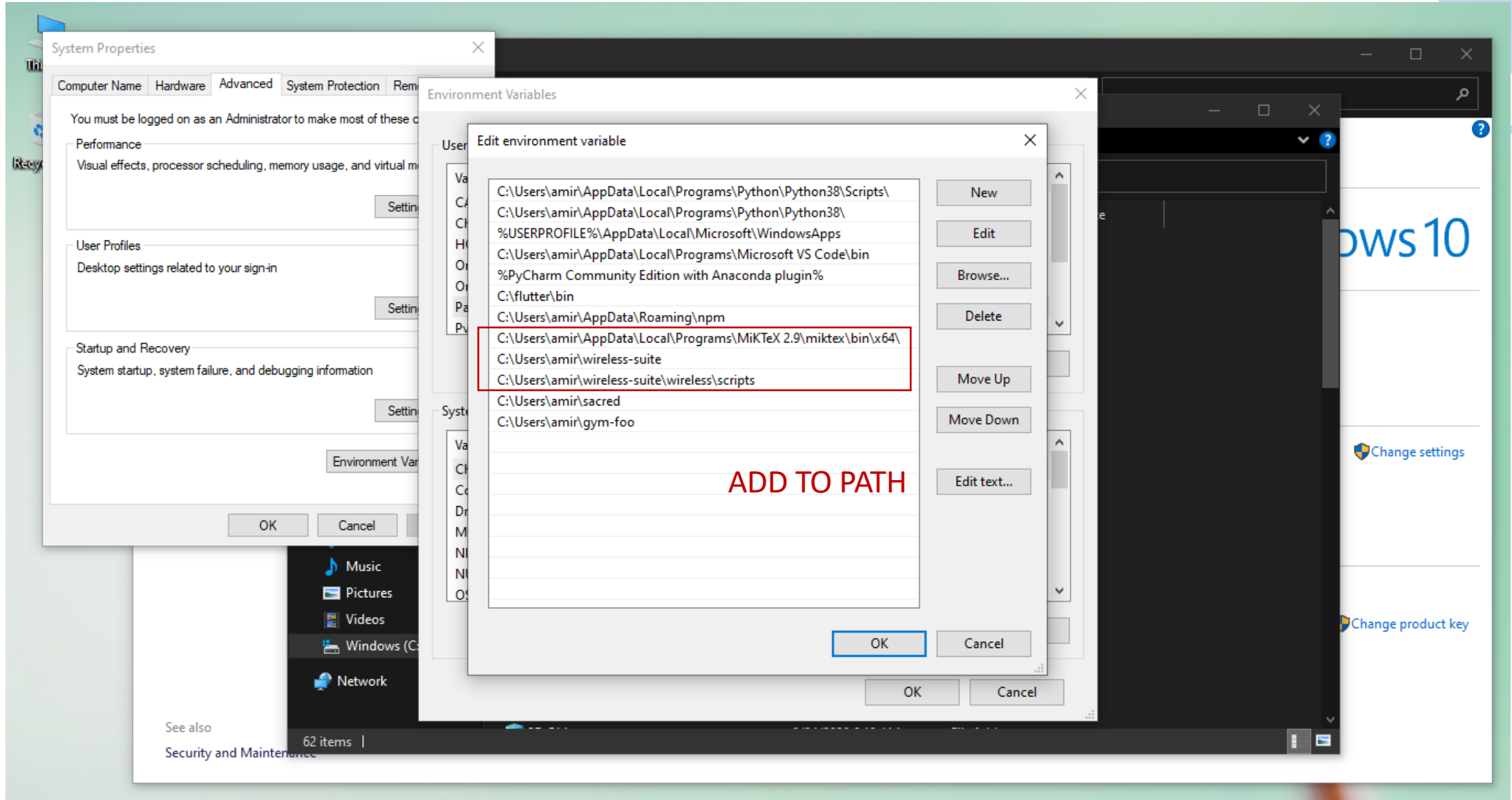
SACRED CONFIGURATION

Machine learning



<https://youtu.be/kopoLzvh5jY>

Package should be add to path list



The image shows a Windows 10 desktop with several overlapping windows. The 'System Properties' window is open, with the 'Advanced' tab selected. The 'Environment Variables' dialog box is also open, showing a list of environment variables. The 'Path' variable is selected, and the 'Edit environment variable' dialog box is open, showing a list of paths. The paths listed are:

- C:\Users\amir\AppData\Local\Programs\Python\Python38\Scripts\
- C:\Users\amir\AppData\Local\Programs\Python\Python38\
- %USERPROFILE%\AppData\Local\Microsoft\WindowsApps
- C:\Users\amir\AppData\Local\Programs\Microsoft VS Code\bin
- %PyCharm Community Edition with Anaconda plugin%
- C:\flutter\bin
- C:\Users\amir\AppData\Roaming\npm
- C:\Users\amir\AppData\Local\Programs\MiKTeX 2.9\miktex\bin\x64\
- C:\Users\amir\wireless-suite
- C:\Users\amir\wireless-suite\wireless\scripts
- C:\Users\amir\sacred
- C:\Users\amir\gym-foo

The path **C:\Users\amir\wireless-suite\wireless\scripts** is highlighted with a red box. The text **ADD TO PATH** is written in red below the list. The 'OK' button is highlighted with a blue border.



Should be import mongo observer

```
1  """
2  © 2020 Nokia
3  Licensed under the BSD 3 Clause license
4  SPDX-License-Identifier: BSD-3-Clause
5  """
6  import gym
7  import json
8
9  from sacred import Experiment
10 from sacred.observers import MongoObserver
11
12 from wireless.agents.bosch_agent import BoschAgent
13 from wireless.agents.time_freq_resource_allocation_v0.round_robin_agent import *
14 from wireless.agents.time_freq_resource_allocation_v0.proportional_fair import *
15 from wireless.agents.noma_ul_time_freq_resource_allocation_v0.noma_ul_proportional_fair import *
16
17 # Load agent parameters
18 with open('../../config/config_agent.json') as f:
19     ac = json.load(f)
20
21 # Configure experiment
22 with open('../../config/config_sacred.json') as f:
23     sc = json.load(f) # Sacred Configuration
24     ns = sc["sacred"]["n_metrics_points"] # Number of points per episode to log in Sacred
25     ex = Experiment(ac["agent"]["agent_type"], save_git_info=False)
26     ex.add_config(sc)
27     ex.add_config(ac)
28
29 mongo_db_url = f'mongodb://{sc["sacred"]["sacred_user"]}:{sc["sacred"]["sacred_pwd"]}@" + \
30 f'f'{sc["sacred"]["sacred_host"]}:{sc["sacred"]["sacred_port"]}/{sc["sacred"]["sacred_db"]}'
31 ex.observers.append(MongoObserver(url=mongo_db_url, db_name=sc["sacred"]["sacred_db"])) # Uncomment to sa
32
33 # Load environment parameters
34 with open('../../config/config_environment.json') as f:
```

IMPORT MONGODB AS OBSERVER

NO NEW NOTIFICATIONS

Ln 10, Col 43 Spaces: 4 UTF-8 LF Python



Configuration of mongodb

The screenshot displays the MongoDB Compass interface on the left and a Notepad window showing the `mongod.cfg` configuration file on the right. The Notepad window is titled `mongod.cfg - Notepad` and shows the following configuration options:

```
# mongod.conf

# for documentation of all options, see:
# http://docs.mongodb.org/manual/reference/configuration-options/

# Where and how to store data.
storage:
  dbPath: C:\data\db
  journal:
    enabled: true

# engine:
# mmapv1:
# wiredTiger:

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: C:\Program Files\MongoDB\Server\4.4\log\mongod.log

# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1

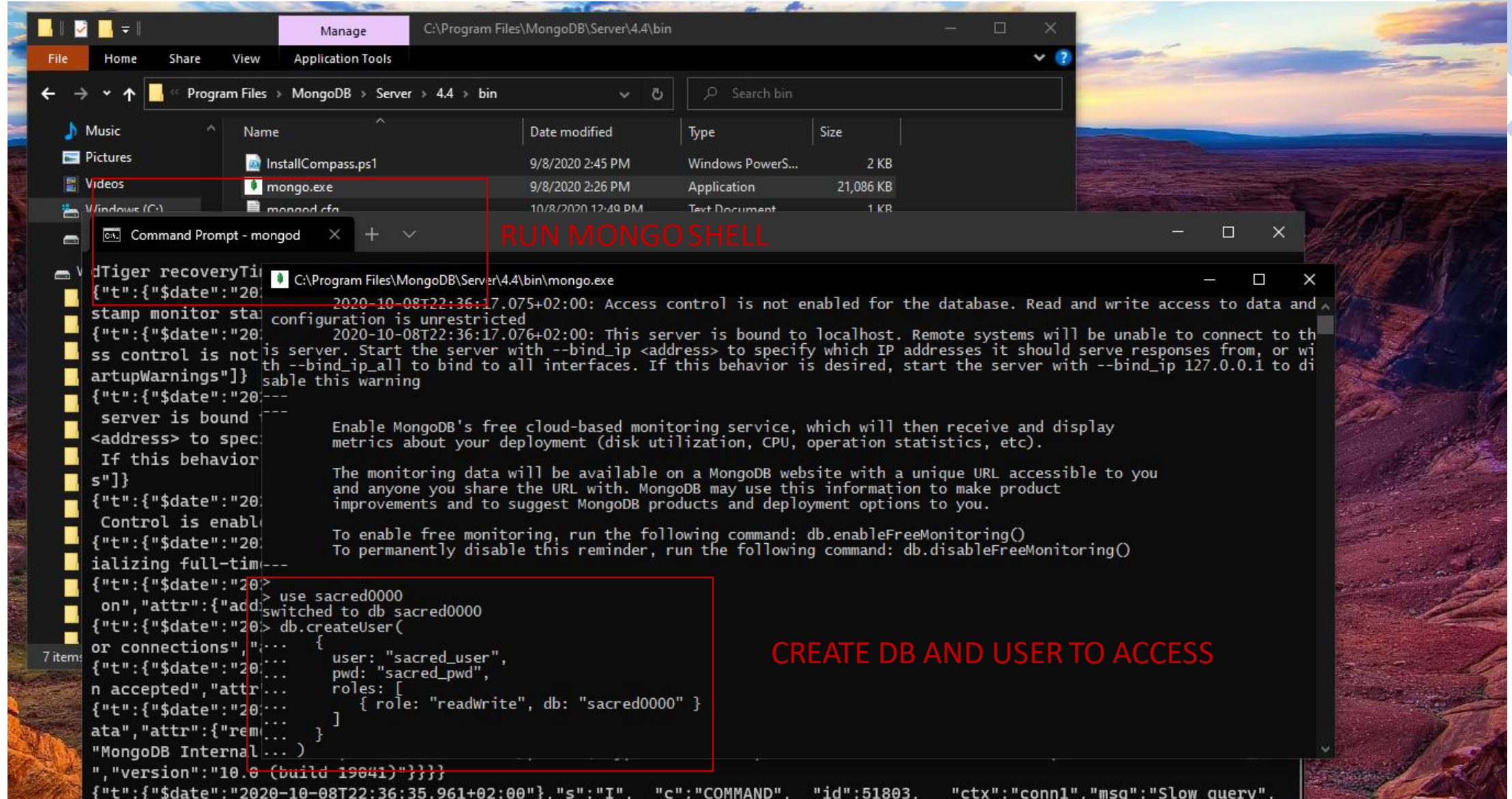
#processManagement:
```

Two red boxes highlight specific sections of the configuration file:

- A red box around the `storage` section is labeled **DATA BASE PATH**.
- A red box around the `net` section is labeled **PORT AND HOST**.

The MongoDB Compass interface on the left shows the connection details for `127.0.0.1:27017` and the `sacred0000` database. The `metrics` collection is selected, showing a table with columns `SIZE` and `AVG.SIZE`.

Running mongodb and it's shell



The screenshot shows a Windows File Explorer window at the path `C:\Program Files\MongoDB\Server\4.4\bin`. It lists files `InstallCompass.ps1`, `mongo.exe`, and `mongod.cfg`. Overlaid on this is a Command Prompt window titled `Command Prompt - mongod`. The prompt shows the execution of `C:\Program Files\MongoDB\Server\4.4\bin\mongo.exe`. The output displays MongoDB startup logs, including a warning about access control and a message about free cloud-based monitoring. The user enters `use sacred0000` and `db.createUser` to create a user. The prompt then shows the user being added to the `sacred0000` database with the `readWrite` role. The prompt ends with a `Slow query` message.

RUN MONGOSHELL

CREATE DB AND USER TO ACCESS



See the data in GUI mongodb compass

MongoDB Compass - 127.0.0.1:27017

Connect View Help

Local

4 DBS 6 COLLECTIONS

☆ FAVORITE

HOST
127.0.0.1:27017

CLUSTER
Standalone

EDITION
MongoDB 4.4.1 Community

Filter your data

> admin

> config

> local

▼ sacred0000

- fs.chunks
- fs.files
- metrics
- runs
- user

CREATE COLLECTION

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
fs.chunks	1	6.5 KB	6.5 KB	2	40.0 KB	
fs.files	1	191.0 B	191.0 B	2	40.0 KB	
metrics	113	3.8 KB	426.4 KB	1	36.0 KB	
runs	1	44.8 KB	44.8 KB	1	20.0 KB	
user	1	43.0 B	43.0 B	1	20.0 KB	

COMPASS GUI

> _MongoSH Beta





REWARD VALUE AS RESULT



Evaluate all the datas that are saved

MongoDB Compass - 127.0.0.1:27017/sacred0000.metrics

Connect View Collection Help

Local

4 DBS 6 COLLECTIONS

☆ FAVORITE

HOST
127.0.0.1:27017

CLUSTER
Standalone

EDITION
MongoDB 4.4.1 Community

Filter your data

- admin
- config
- local
- sacred0000
 - fs.chunks
 - fs.files
 - metrics
 - runs
 - user

> _MongoSH Beta

sacred0000.metrics Documents

sacred0000.metrics

DOCUMENTS 113 TOTAL SIZE 426.4KB AVG. SIZE 3.8KB INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER

ADD DATA

VIEW

Displaying documents 1 - 20 of 113

values: Array

```
> {
  "_id": ObjectId("5f7efe40fbd7f0788373ab47"),
  "name": "Episode 2. UE 16. CQI vs time step",
  "run_id": 1,
  "steps": Array,
  "timestamps": Array,
  "values": Array
}
```

```
> {
  "_id": ObjectId("5f7efe40fbd7f0788373ab49"),
  "name": "Episode 2. UE 16. Buffer occupancy [bits] vs time step",
  "run_id": 1,
  "steps": Array,
  "timestamps": Array,
  "values": Array
    0: 0
    1: 0
    2: 40093
    3: 0
    4: 0
    5: 0
    6: 31201
    7: 0
    8: 33349
    9: 0
    10: 0
  }

```

STORED DATA



Gym environment

gym/creating-environments.md x +

https://github.com/openai/gym/blob/master/docs/creating-environments.md

- Create a new repo called gym-foo, which should also be a PIP package.
- A good example is <https://github.com/openai/gym-soccer>.
- It should have at least the following files:

```
gym-foo/  
  README.md  
  setup.py  
  gym_foo/  
    __init__.py  
  envs/  
    __init__.py  
    foo_env.py  
    foo_extrahard_env.py
```
- gym-foo/setup.py should have:

```
from setuptools import setup  
  
setup(name='gym_foo',  
      version='0.0.1',  
      install_requires=['gym'] # And any other dependencies foo needs  
)
```
- gym-foo/gym_foo/__init__.py should have:

MAKE ENVIRONMENT



Test open ai gym

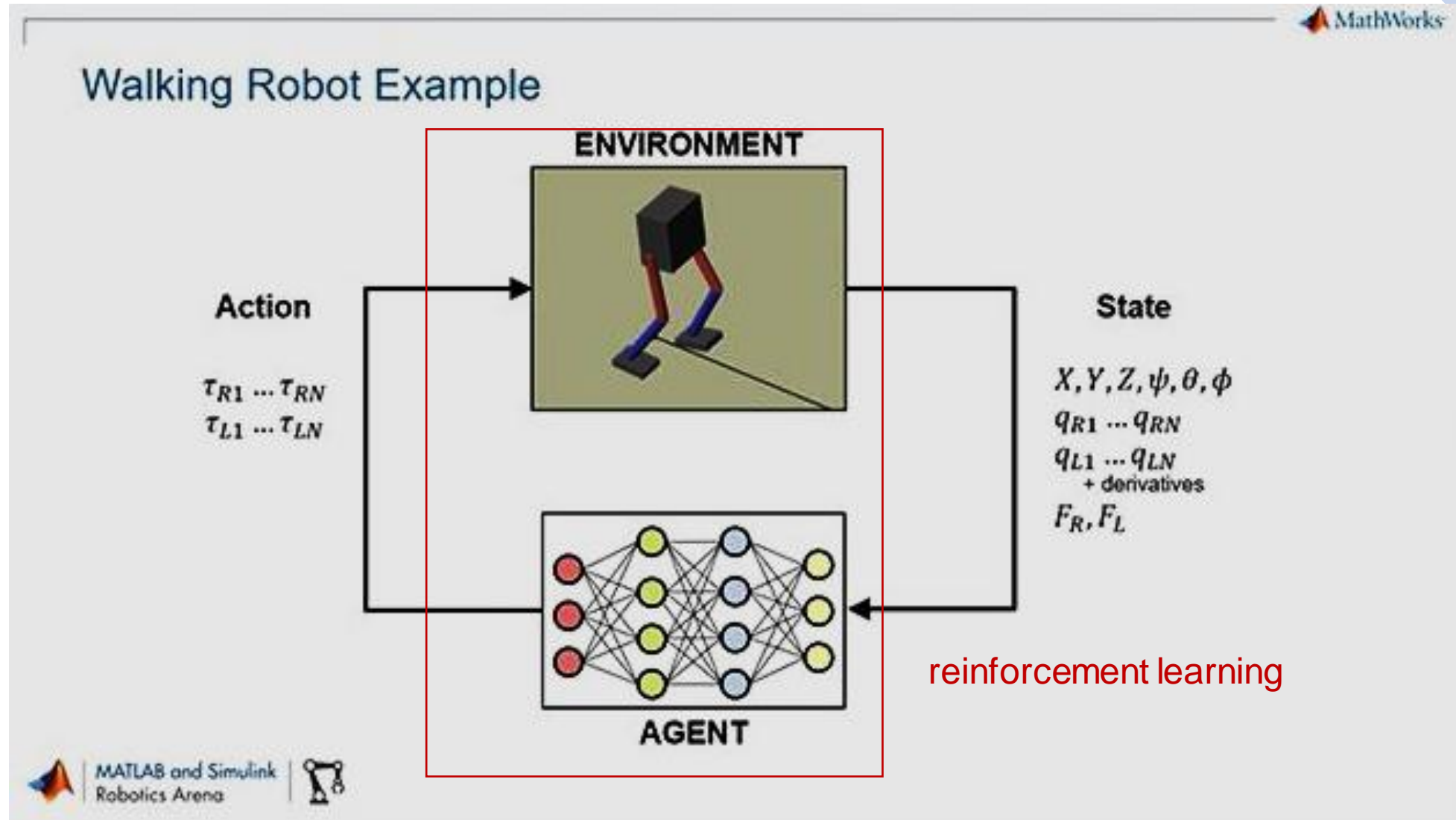
test.py

```
1 import gym
2
3 env = gym.make('CartPole-v0')
4 env.reset()
5
6 for _ in range(1000):
7     env.render()
8     env.step(env.action_space.sample()) # take action
9 env.close()
10
```

GYM ENVIRONMENT EXAMPLE

Terminal output:

```
PS D:\COURSE\THESIS\code gym> cd 'd:\COURSE\THESIS\code gym' & python 'c:\Users\amir\AppData\Roaming\Python\Python38\site-packages\gym\logger.py:30: UserWarning: WARN: You are calling 'step()' even though this environment has already returned done = True. You should always call 'reset()' once you receive 'done = True' -- any further steps are undefined behavior.
warnings.warn(colorize('%s: %s'%( 'WARN', msg % args), 'yellow'))
PS D:\COURSE\THESIS\code gym> cd 'd:\COURSE\THESIS\code gym'; & 'python' 'c:\Users\amir\AppData\Roaming\Python\Python38\site-packages\gym\logger.py:30: UserWarning: WARN: You are calling 'step()' even though this environment has already returned done = True. You should always call 'reset()' once you receive 'done = True' -- any further steps are undefined behavior.
warnings.warn(colorize('%s: %s'%( 'WARN', msg % args), 'yellow'))
```

<https://it.mathworks.com/products/reinforcement-learning.html>



full thesis

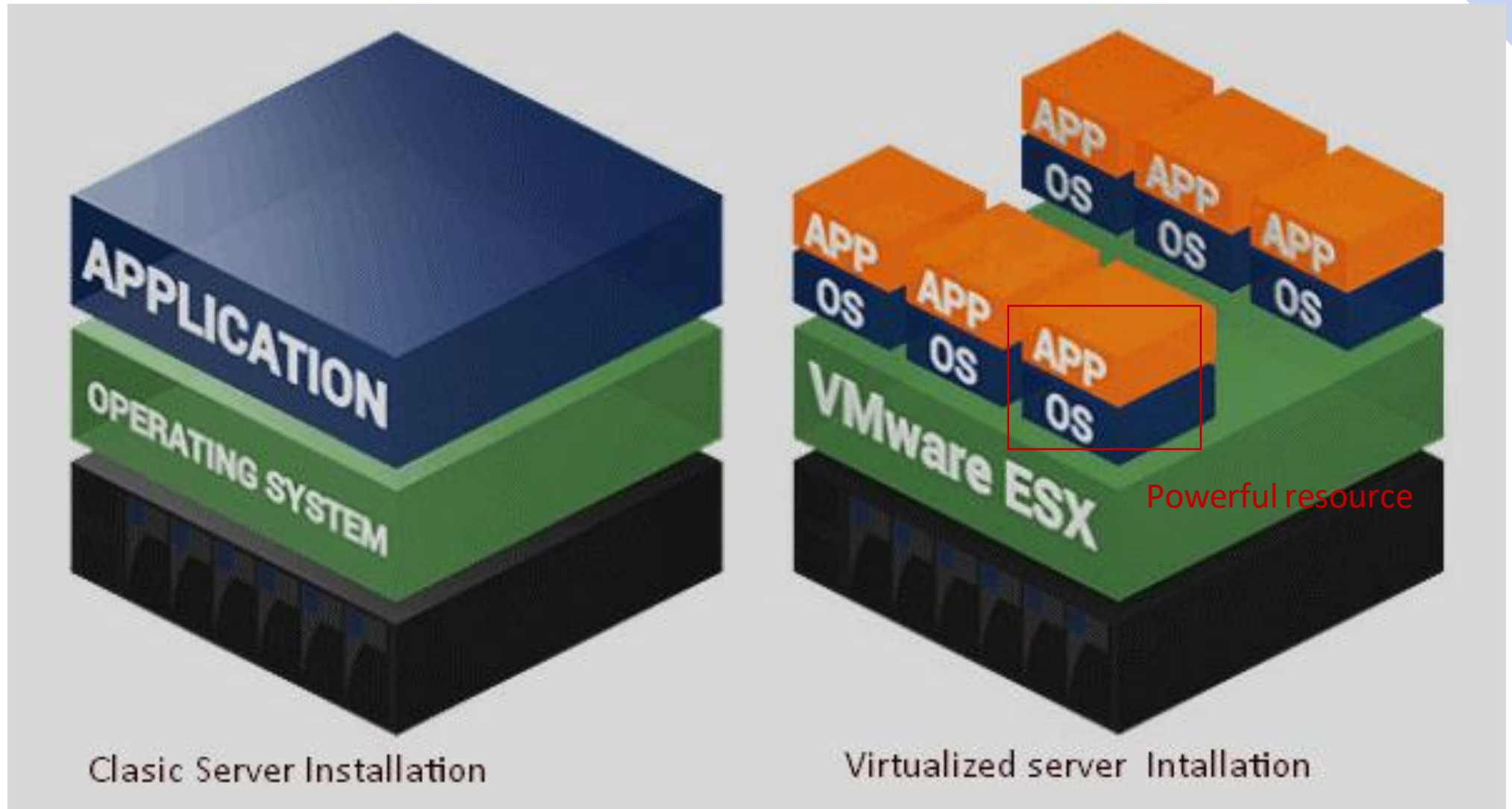
Artificial intelligence in the air

26/10/2020 (oct)

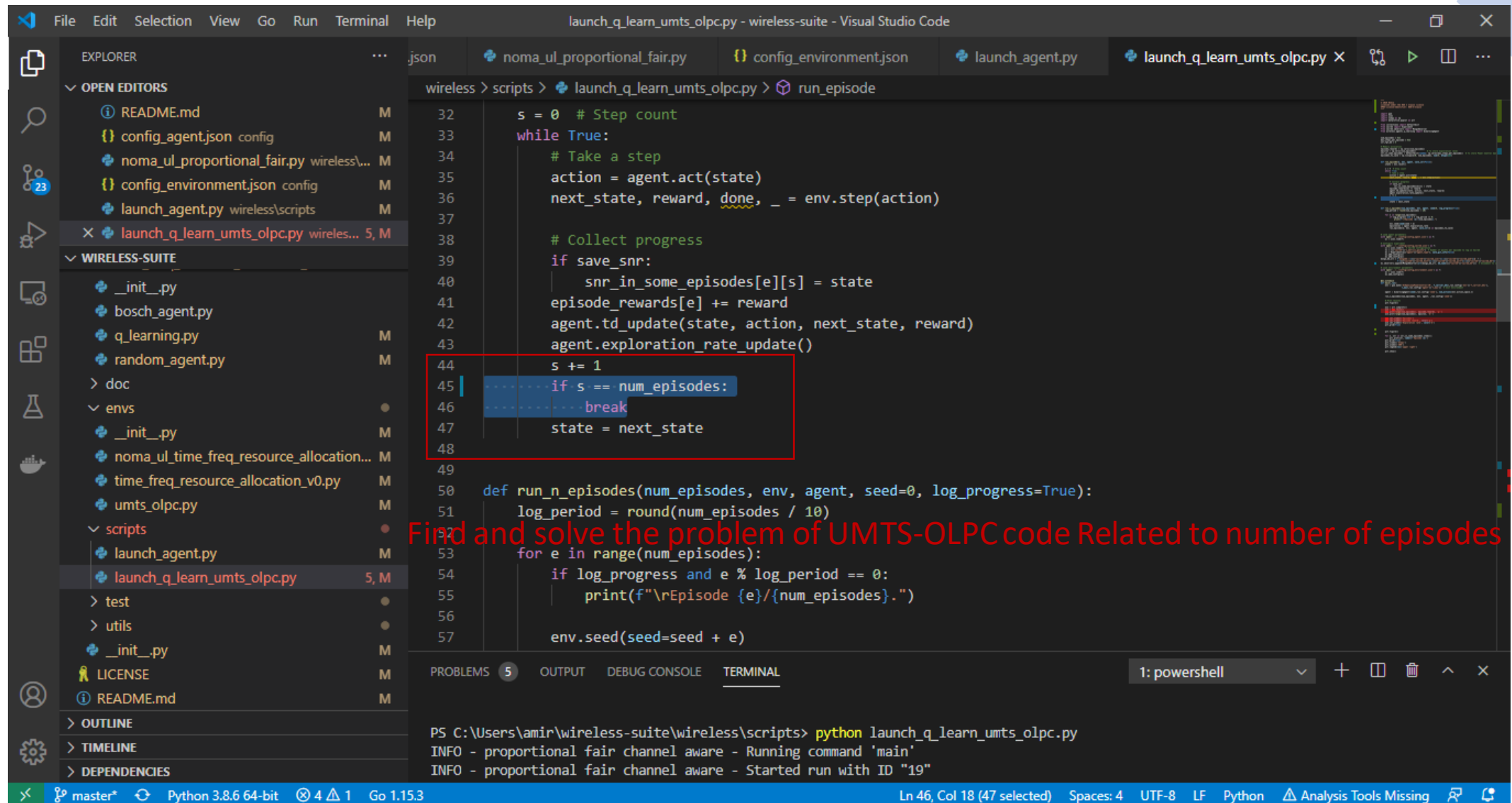
- Exploit all the specification of q-learning python program
 - improve the time efficiency

Meeting **three**

Virtually run on top of powerful HW



Q-learn



```
File Edit Selection View Go Run Terminal Help
launch_q_learn_umts_olpc.py - wireless-suite - Visual Studio Code

EXPLORER
OPEN EDITORS
  README.md M
  config_agent.json config M
  noma_ul_proportional_fair.py wireless... M
  config_environment.json config M
  launch_agent.py wireless/scripts M
  launch_q_learn_umts_olpc.py wireless... 5, M
WIRELESS-SUITE
  __init__.py
  bosch_agent.py
  q_learning.py M
  random_agent.py M
  doc
  envs
    __init__.py M
    noma_ul_time_freq_resource_allocation... M
    time_freq_resource_allocation_v0.py M
    umts_olpc.py M
  scripts
    launch_agent.py M
    launch_q_learn_umts_olpc.py 5, M
  test
  utils
  __init__.py M
  LICENSE M
  README.md M
  OUTLINE
  TIMELINE
  DEPENDENCIES

wireless > scripts > launch_q_learn_umts_olpc.py > run_episode
32 s = 0 # Step count
33 while True:
34     # Take a step
35     action = agent.act(state)
36     next_state, reward, done, _ = env.step(action)
37
38     # Collect progress
39     if save_snr:
40         snr_in_some_episodes[e][s] = state
41         episode_rewards[e] += reward
42         agent.td_update(state, action, next_state, reward)
43         agent.exploration_rate_update()
44     s += 1
45     if s == num_episodes:
46         break
47     state = next_state
48
49
50 def run_n_episodes(num_episodes, env, agent, seed=0, log_progress=True):
51     log_period = round(num_episodes / 10)
52     for e in range(num_episodes):
53         if log_progress and e % log_period == 0:
54             print(f"\rEpisode {e}/{num_episodes}.")
55         env.seed(seed=seed + e)
56
57     env.seed(seed=seed + e)
```

Find and solve the problem of UMTS-OLPC code Related to number of episodes

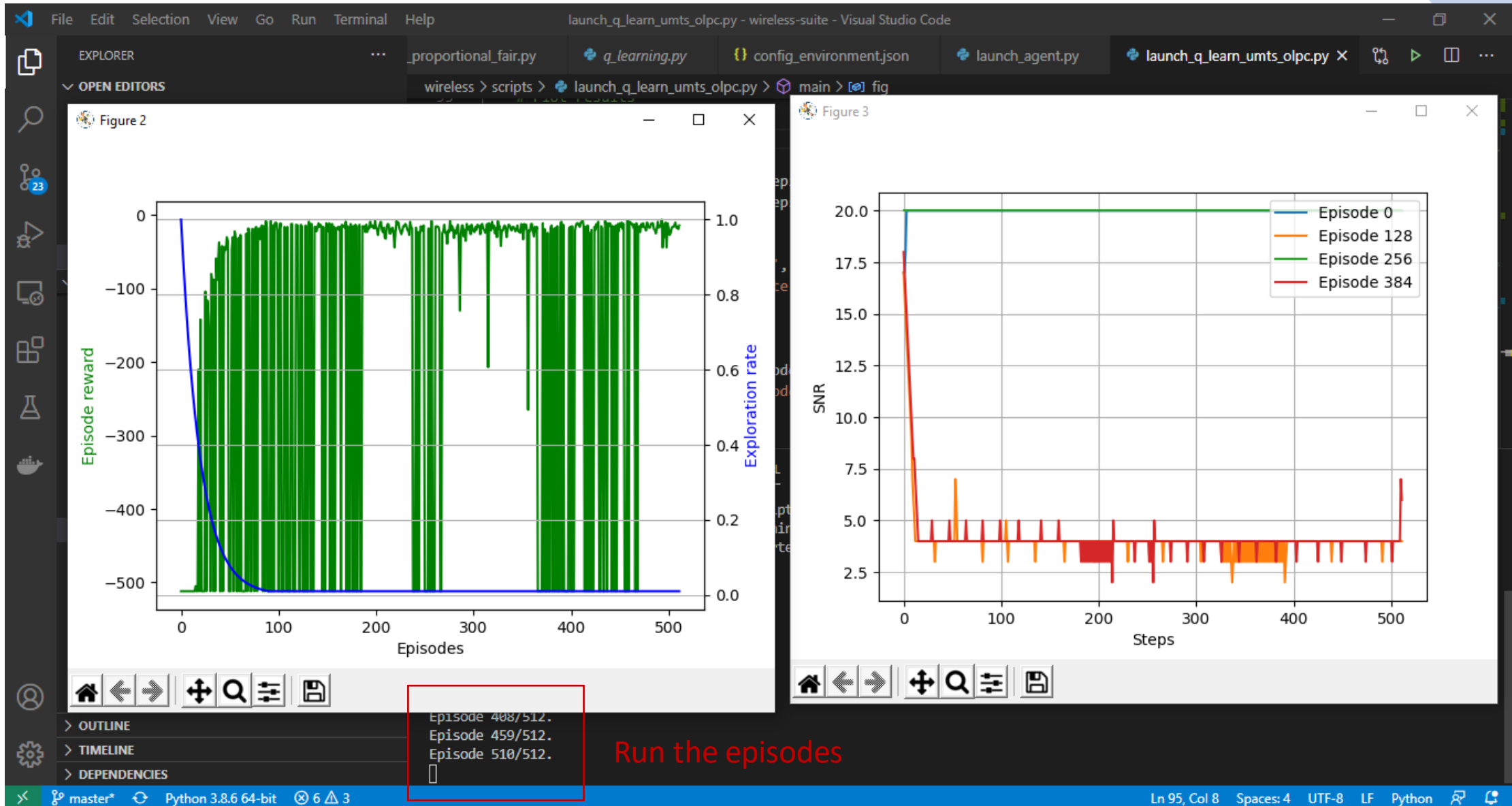
1: powershell

PS C:\Users\amir\wireless-suite\wireless\scripts> python launch_q_learn_umts_olpc.py
INFO - proportional fair channel aware - Running command 'main'
INFO - proportional fair channel aware - Started run with ID "19"

master* Python 3.8.6 64-bit 4 1 Go 1.15.3 Ln 46, Col 18 (47 selected) Spaces: 4 UTF-8 LF Python Analysis Tools Missing

<https://www.youtube.com/watch?v=e3L4VocZnnQ>

Output plots





full thesis

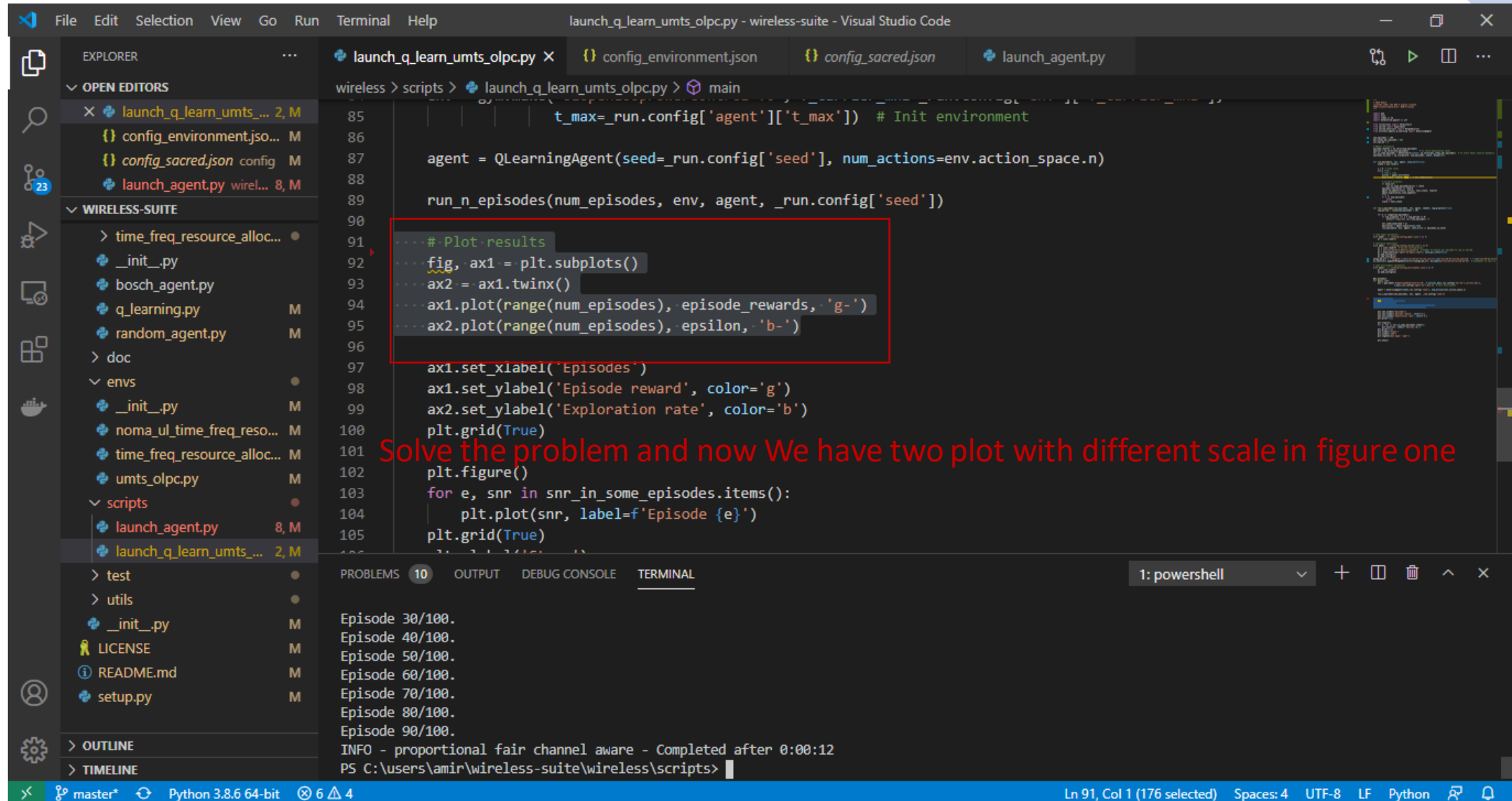
Artificial intelligence in the air

10/11/2020 (nov)

- Solve the problems of q-learning python program
 - change the configuration and compare result

Meeting **four**

Fix the problem of plots



```

85 |         t_max=_run.config['agent']['t_max']) # Init environment
86 |
87 |     agent = QLearningAgent(seed=_run.config['seed'], num_actions=env.action_space.n)
88 |
89 |     run_n_episodes(num_episodes, env, agent, _run.config['seed'])
90 |
91 |     ...#Plot results
92 |     fig, ax1 = plt.subplots()
93 |     ax2 = ax1.twinx()
94 |     ax1.plot(range(num_episodes), episode_rewards, 'g-')
95 |     ax2.plot(range(num_episodes), epsilon, 'b-')
96 |
97 |     ax1.set_xlabel('Episodes')
98 |     ax1.set_ylabel('Episode reward', color='g')
99 |     ax2.set_ylabel('Exploration rate', color='b')
100 |     plt.grid(True)
101 |     plt.figure()
102 |     for e, snr in snr_in_some_episodes.items():
103 |         plt.plot(snr, label=f'Episode {e}')
104 |     plt.grid(True)
105 |

```

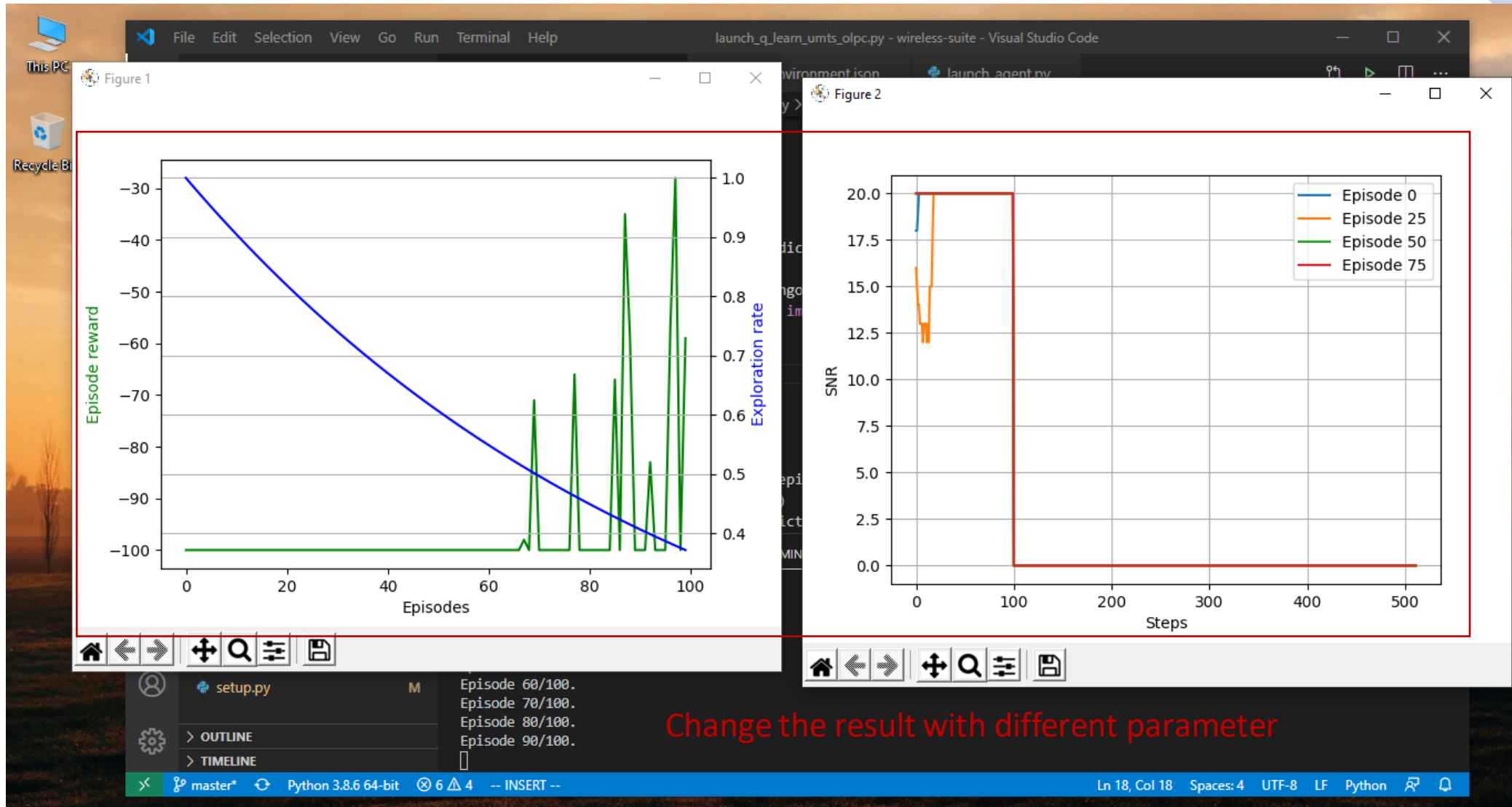
Solve the problem and now We have two plot with different scale in figure one

```

Episode 30/100.
Episode 40/100.
Episode 50/100.
Episode 60/100.
Episode 70/100.
Episode 80/100.
Episode 90/100.
INFO - proportional fair channel aware - Completed after 0:00:12
PS C:\users\amir\wireless-suite\wireless\scripts>

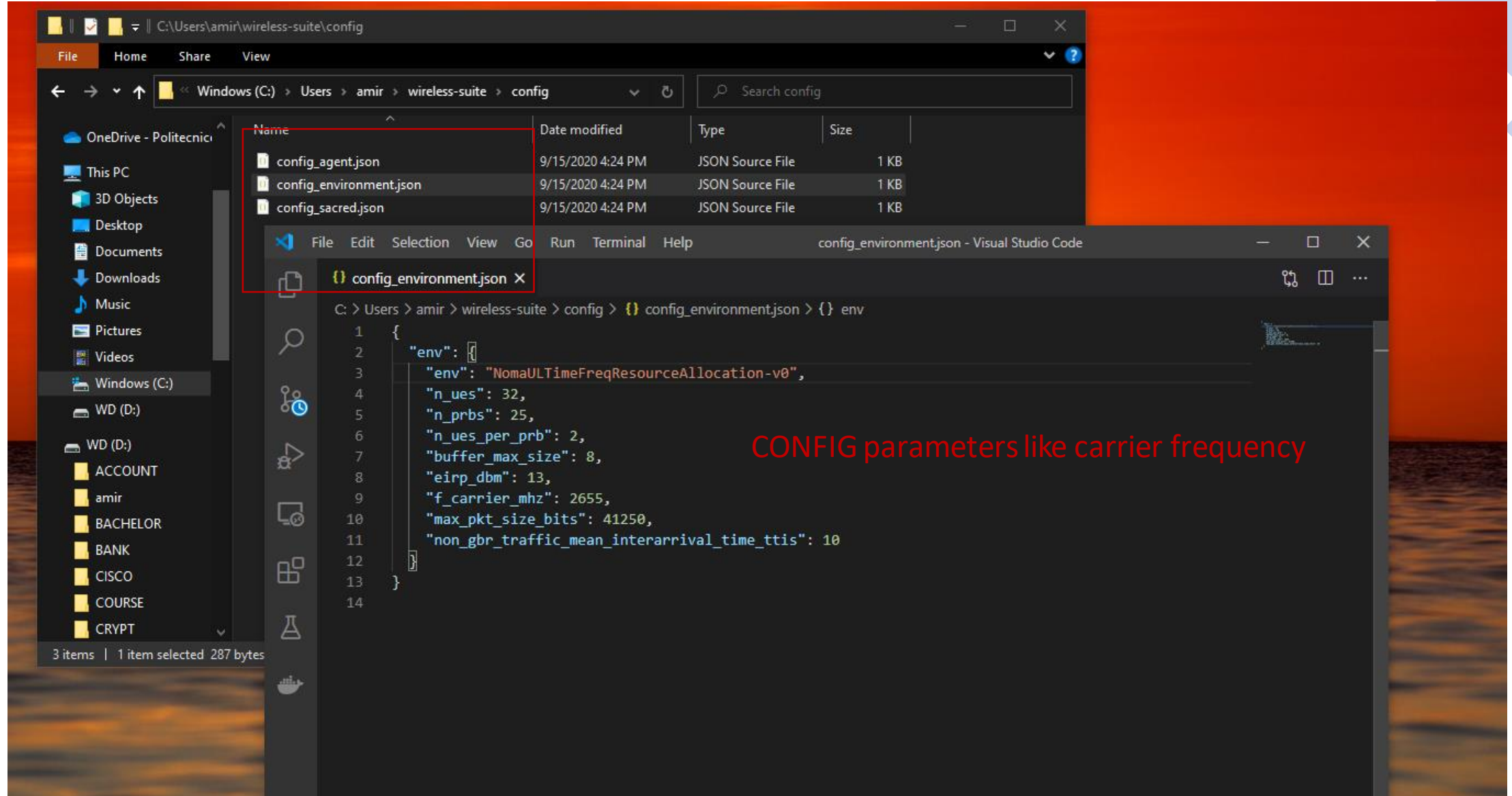
```

Comparing results with change config file



Change the result with different parameter

Parameters that we can change



The screenshot shows a Windows File Explorer window displaying the contents of the `C:\Users\amir\wireless-suite\config` directory. Three JSON files are listed: `config_agent.json`, `config_environment.json`, and `config_sacred.json`. A red box highlights the `config_environment.json` file. Below the File Explorer, a Visual Studio Code window is open, showing the contents of `config_environment.json`. The file contains a JSON object with various parameters, including `env`, `n_ues`, `n_prbs`, `n_ues_per_prb`, `buffer_max_size`, `eirp_dbm`, `f_carrier_mhz`, `max_pkt_size_bits`, and `non_gbr_traffic_mean_interarrival_time_ttis`. A red text overlay on the right side of the VS Code window reads "CONFIG parameters like carrier frequency".

Name	Date modified	Type	Size
<code>config_agent.json</code>	9/15/2020 4:24 PM	JSON Source File	1 KB
<code>config_environment.json</code>	9/15/2020 4:24 PM	JSON Source File	1 KB
<code>config_sacred.json</code>	9/15/2020 4:24 PM	JSON Source File	1 KB

```
1 {  
2   "env": {  
3     "env": "NomaULTimeFreqResourceAllocation-v0",  
4     "n_ues": 32,  
5     "n_prbs": 25,  
6     "n_ues_per_prb": 2,  
7     "buffer_max_size": 8,  
8     "eirp_dbm": 13,  
9     "f_carrier_mhz": 2655,  
10    "max_pkt_size_bits": 41250,  
11    "non_gbr_traffic_mean_interarrival_time_ttis": 10  
12  }  
13 }  
14
```

CONFIG parameters like carrier frequency



full thesis

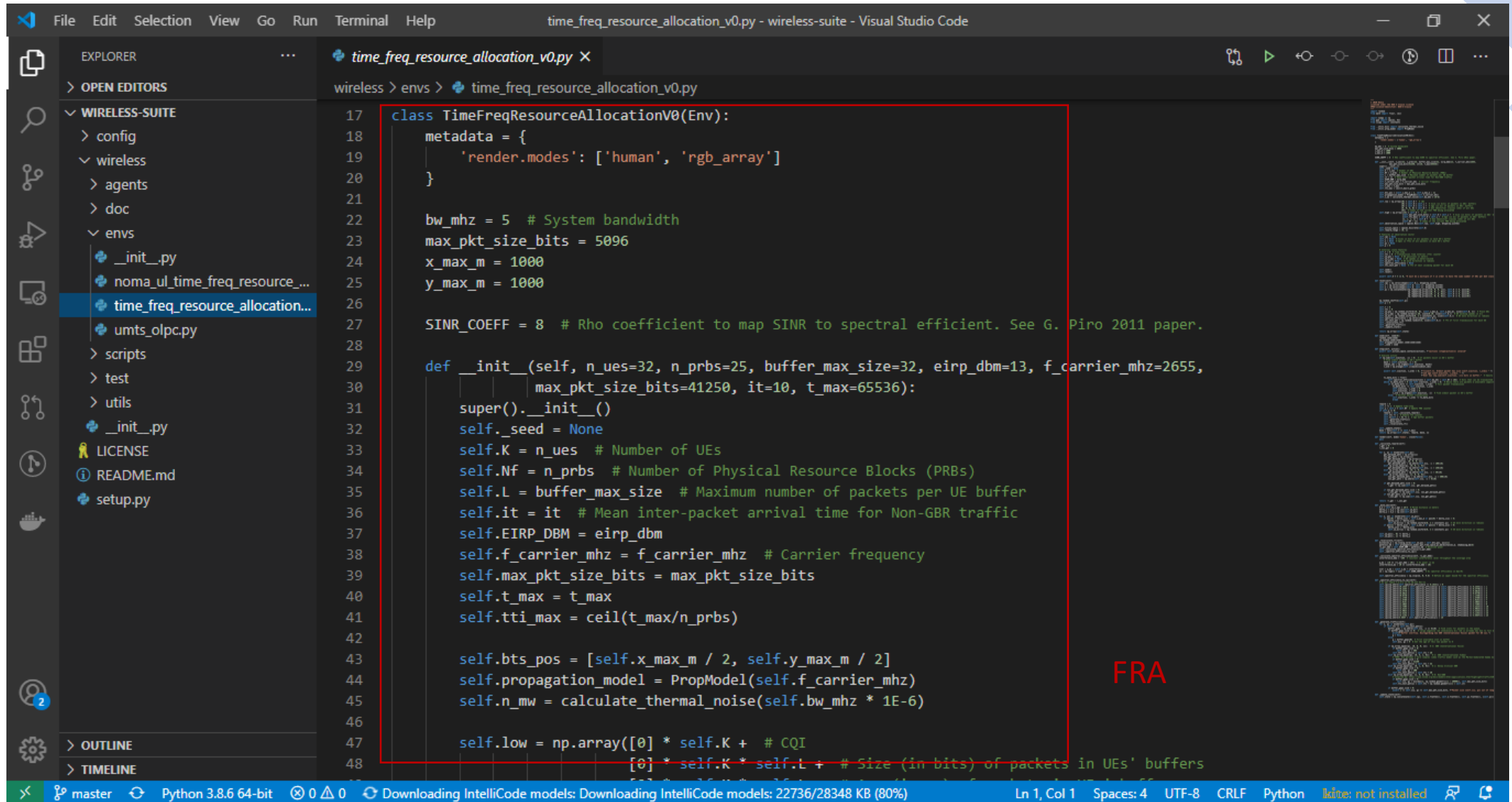
Artificial intelligence in the air

01/12/2020 (dec)

- find split in time frequency file
- implement qlearn in time frequency program

Meeting **five**

Frequency resource allocation



```
class TimeFreqResourceAllocationV0(Env):
    metadata = {
        'render.modes': ['human', 'rgb_array']
    }

    bw_mhz = 5 # System bandwidth
    max_pkt_size_bits = 5096
    x_max_m = 1000
    y_max_m = 1000

    SINR_COEFF = 8 # Rho coefficient to map SINR to spectral efficient. See G. Piro 2011 paper.

    def __init__(self, n_ues=32, n_prbs=25, buffer_max_size=32, eirp_dbm=13, f_carrier_mhz=2655,
                  max_pkt_size_bits=41250, it=10, t_max=65536):
        super().__init__()
        self._seed = None
        self.K = n_ues # Number of UEs
        self.Nf = n_prbs # Number of Physical Resource Blocks (PRBs)
        self.L = buffer_max_size # Maximum number of packets per UE buffer
        self.it = it # Mean inter-packet arrival time for Non-GBR traffic
        self.EIRP_DBM = eirp_dbm
        self.f_carrier_mhz = f_carrier_mhz # Carrier frequency
        self.max_pkt_size_bits = max_pkt_size_bits
        self.t_max = t_max
        self.tti_max = ceil(t_max/n_prbs)

        self.bts_pos = [self.x_max_m / 2, self.y_max_m / 2]
        self.propagation_model = PropModel(self.f_carrier_mhz)
        self.n_mw = calculate_thermal_noise(self.bw_mhz * 1E-6)

        self.low = np.array([0] * self.K + # CQI
                             [0] * self.K * self.L + # Size (in bits) of packets in UEs' buffers
```

FRA