Navigation

October 1, 2019

1 Navigation

You are welcome to use this coding environment to train your agent for the project. Follow the instructions below to get started!

1.0.1 1. Start the Environment

Run the next code cell to install a few packages. This line will take a few minutes to run!

```
In [1]: !pip install unityagents
        !pip -q install ./python
        !pip install numpy==1.13.3
        from unityagents import UnityEnvironment
        import numpy as np
        import random
        import torch
        from collections import deque
        import matplotlib.pyplot as plt
        from dqn_agent import Agent
        %matplotlib inline
```

```
Collecting unityagents
  Downloading https://files.pythonhosted.org/packages/82/42/c337c5ba34c72a2e01fab4abe113ba9f282c
    100% || 81kB 3.8MB/s ta 0:00:011
Collecting tensorflow==1.7.1 (from unityagents)
  Downloading https://files.pythonhosted.org/packages/66/83/35c3f53129dfc80d65ebbe07ef0575263c3c
    100% || 48.1MB 869kB/s eta 0:00:01
Requirement already satisfied: numpy>=1.11.0 in /opt/conda/lib/python3.6/site-packages (from uni
Requirement already satisfied: pytest>=3.2.2 in /opt/conda/lib/python3.6/site-packages (from uni
Requirement already satisfied: pyyaml in /opt/conda/lib/python3.6/site-packages (from unityagent
Collecting jupyter (from unityagents)
  Downloading https://files.pythonhosted.org/packages/83/df/0f5dd132200728a86190397e1ea87cd76244
Collecting grpcio==1.11.0 (from unityagents)
  Downloading https://files.pythonhosted.org/packages/c8/b8/00e703183b7ae5e02f161dafacdfa8edbd72
    100% || 8.8MB 4.8MB/s eta 0:00:01
```

```
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.6/site-packages (from unitya
Requirement already satisfied: Pillow>=4.2.1 in /opt/conda/lib/python3.6/site-packages (from uni
Collecting protobuf==3.5.2 (from unityagents)
  Downloading https://files.pythonhosted.org/packages/9b/0d/6e9cf54be86c742a375a0ca0addadb582c89
    100% || 6.4MB 10.1MB/s ta 0:00:01
Collecting docopt (from unityagents)
  Downloading https://files.pythonhosted.org/packages/a2/55/8f8cab2afd404cf578136ef2cc5dfb50baa1
Collecting termcolor>=1.1.0 (from tensorflow==1.7.1->unityagents)
  Downloading https://files.pythonhosted.org/packages/8a/48/a76be51647d0eb9f10e2a4511bf3ffb8cc1e
Collecting gast>=0.2.0 (from tensorflow==1.7.1->unityagents)
  Downloading https://files.pythonhosted.org/packages/1f/04/4e36c33f8eb5c5b6c622a1f4859352a6acca
Collecting absl-py>=0.1.6 (from tensorflow==1.7.1->unityagents)
  Downloading https://files.pythonhosted.org/packages/3c/0d/7cbf64cac3f93617a2b6b079c0182e4a83a3
    100% || 112kB 20.2MB/s ta 0:00:01
Collecting astor>=0.6.0 (from tensorflow==1.7.1->unityagents)
  Downloading https://files.pythonhosted.org/packages/d1/4f/950dfae467b384fc96bc6469de25d832534f
Requirement already satisfied: six>=1.10.0 in /opt/conda/lib/python3.6/site-packages (from tenso
Requirement already satisfied: wheel>=0.26 in /opt/conda/lib/python3.6/site-packages (from tenso
Collecting tensorboard<1.8.0,>=1.7.0 (from tensorflow==1.7.1->unityagents)
  Downloading https://files.pythonhosted.org/packages/0b/ec/65d4e8410038ca2a78c09034094403d23122
    100% || 3.1MB 11.1MB/s ta 0:00:01
Requirement already satisfied: more-itertools>=4.0.0; python_version > "2.7" in /opt/conda/lib/p
Requirement already satisfied: pluggy!=0.10,<1.0,>=0.9 in /opt/conda/lib/python3.6/site-packages
Requirement already satisfied: py>=1.5.0 in /opt/conda/lib/python3.6/site-packages (from pytest>
Requirement already satisfied: attrs>=17.4.0 in /opt/conda/lib/python3.6/site-packages (from pyt
Requirement already satisfied: atomicwrites>=1.0 in /opt/conda/lib/python3.6/site-packages (from
Requirement already satisfied: setuptools in /opt/conda/lib/python3.6/site-packages (from pytest
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.6/site-packages (from pytest>=3
Requirement already satisfied: nbconvert in /opt/conda/lib/python3.6/site-packages (from jupyter
Requirement already satisfied: notebook in /opt/conda/lib/python3.6/site-packages (from jupyter-
Collecting jupyter-console (from jupyter->unityagents)
  Downloading https://files.pythonhosted.org/packages/cb/ee/6374ae8c21b7d0847f9c3722dcdfac986b8e
Requirement already satisfied: ipywidgets in /opt/conda/lib/python3.6/site-packages (from jupyte
Collecting qtconsole (from jupyter->unityagents)
  Downloading https://files.pythonhosted.org/packages/d0/fc/4936c60be8d56acedfefcc13cf9f3d881e40
    100% || 122kB 19.2MB/s ta 0:00:01
Requirement already satisfied: ipykernel in /opt/conda/lib/python3.6/site-packages (from jupyter
Requirement already satisfied: python-dateutil>=2.0 in /opt/conda/lib/python3.6/site-packages (f
Requirement already satisfied: pytz in /opt/conda/lib/python3.6/site-packages (from matplotlib->
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.6/site-packages/cycler-0.1
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/lib/python
Requirement already satisfied: bleach==1.5.0 in /opt/conda/lib/python3.6/site-packages (from ten
Requirement already satisfied: markdown>=2.6.8 in /opt/conda/lib/python3.6/site-packages (from t
Requirement already satisfied: html5lib==0.9999999 in /opt/conda/lib/python3.6/site-packages (fr
Requirement already satisfied: werkzeug>=0.11.10 in /opt/conda/lib/python3.6/site-packages (from
Requirement already satisfied: mistune>=0.8.1 in /opt/conda/lib/python3.6/site-packages (from nb
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.6/site-packages (from nbconvert-
Requirement already satisfied: pygments in /opt/conda/lib/python3.6/site-packages (from nbconver
```

```
Requirement already satisfied: traitlets>=4.2 in /opt/conda/lib/python3.6/site-packages (from nb
Requirement already satisfied: jupyter_core in /opt/conda/lib/python3.6/site-packages (from nbco
Requirement already satisfied: nbformat>=4.4 in /opt/conda/lib/python3.6/site-packages (from nbc
Requirement already satisfied: entrypoints>=0.2.2 in /opt/conda/lib/python3.6/site-packages (fro
Requirement already satisfied: pandocfilters>=1.4.1 in /opt/conda/lib/python3.6/site-packages (f
Requirement already satisfied: testpath in /opt/conda/lib/python3.6/site-packages (from nbconver
Requirement already satisfied: defusedxml in /opt/conda/lib/python3.6/site-packages (from nbconv
Requirement already satisfied: tornado>=4 in /opt/conda/lib/python3.6/site-packages (from notebo
Requirement already satisfied: pyzmq>=17 in /opt/conda/lib/python3.6/site-packages (from noteboo
Requirement already satisfied: ipython_genutils in /opt/conda/lib/python3.6/site-packages (from
Requirement already satisfied: jupyter_client>=5.2.0 in /opt/conda/lib/python3.6/site-packages (
Requirement already satisfied: Send2Trash in /opt/conda/lib/python3.6/site-packages (from notebo
Requirement already satisfied: terminado>=0.8.1 in /opt/conda/lib/python3.6/site-packages (from
Requirement already satisfied: prometheus_client in /opt/conda/lib/python3.6/site-packages (from
Collecting prompt-toolkit<2.1.0,>=2.0.0 (from jupyter-console->jupyter->unityagents)
  Downloading https://files.pythonhosted.org/packages/f7/a7/9b1dd14ef45345f186ef69d175bdd2491c40
    100% || 337kB 20.1MB/s ta 0:00:01
Requirement already satisfied: ipython in /opt/conda/lib/python3.6/site-packages (from jupyter-c
Collecting widgetsnbextension~=3.0.0 (from ipywidgets->jupyter->unityagents)
  Downloading https://files.pythonhosted.org/packages/8d/f2/c8bcccccbed39d51d3e237fb0c0f0c9bbc84
    100% || 2.2MB 16.6MB/s ta 0:00:01
Requirement already satisfied: MarkupSafe>=0.23 in /opt/conda/lib/python3.6/site-packages (from
Requirement already satisfied: decorator in /opt/conda/lib/python3.6/site-packages (from traitle
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /opt/conda/lib/python3.6/site-packages
Requirement already satisfied: backcall in /opt/conda/lib/python3.6/site-packages (from ipython-
Requirement already satisfied: pickleshare in /opt/conda/lib/python3.6/site-packages (from ipyth
Requirement already satisfied: pexpect; sys_platform != "win32" in /opt/conda/lib/python3.6/site
Requirement already satisfied: simplegeneric>0.8 in /opt/conda/lib/python3.6/site-packages (from
Requirement already satisfied: jedi>=0.10 in /opt/conda/lib/python3.6/site-packages (from ipytho
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/lib/python3.6/site-packages (from p
Building wheels for collected packages: docopt, termcolor, gast, absl-py
  Running setup.py bdist_wheel for docopt ... done
  Stored in directory: /root/.cache/pip/wheels/9b/04/dd/7daf4150b6d9b12949298737de9431a324d4b797
  Running setup.py bdist_wheel for termcolor ... done
  Stored in directory: /root/.cache/pip/wheels/7c/06/54/bc84598ba1daf8f970247f550b175aaaee85f68b
  Running setup.py bdist_wheel for gast ... done
  Stored in directory: /root/.cache/pip/wheels/59/38/c6/234dc39b4f6951a0768fbc02d5b7207137a5b1d9
  Running setup.py bdist_wheel for absl-py ... done
  Stored in directory: /root/.cache/pip/wheels/9a/1e/7a/456008eb5e47fd5de792c6139df6d5b3d5f71d51
Successfully built docopt termcolor gast absl-py
tensorflow 1.7.1 has requirement numpy>=1.13.3, but you'll have numpy 1.12.1 which is incompatib
ipython 6.5.0 has requirement prompt-toolkit<2.0.0,>=1.0.15, but you'll have prompt-toolkit 2.0.
Installing collected packages: protobuf, grpcio, termcolor, gast, absl-py, astor, tensorboard, t
  Found existing installation: protobuf 3.5.1
    Uninstalling protobuf-3.5.1:
      Successfully uninstalled protobuf-3.5.1
  Found existing installation: tensorflow 1.3.0
    Uninstalling tensorflow-1.3.0:
```

```
      Successfully uninstalled tensorflow-1.3.0
  Found existing installation: prompt-toolkit 1.0.15
    Uninstalling prompt-toolkit-1.0.15:
      Successfully uninstalled prompt-toolkit-1.0.15
  Found existing installation: widgetsnbextension 3.1.0
    Uninstalling widgetsnbextension-3.1.0:
      Successfully uninstalled widgetsnbextension-3.1.0
Successfully installed absl-py-0.8.0 astor-0.8.0 docopt-0.6.2 gast-0.3.2 grpcio-1.11.0 jupyter-1
Collecting numpy==1.13.3
  Downloading https://files.pythonhosted.org/packages/57/a7/e3e6bd9d595125e1abbe162e323fd2d06f6f
    100% || 17.0MB 2.5MB/s eta 0:00:01
Installing collected packages: numpy
  Found existing installation: numpy 1.12.1
    Uninstalling numpy-1.12.1:
      Successfully uninstalled numpy-1.12.1
Successfully installed numpy-1.13.3
```

The environment is already saved in the Workspace and can be accessed at the file path provided below. Please run the next code cell without making any changes.

```
In [2]: # please do not modify the line below
        env = UnityEnvironment(file_name="/data/Banana_Linux_NoVis/Banana.x86_64")
```

```
INFO:unityagents:
'Academy' started successfully!
Unity Academy name: Academy
        Number of Brains: 1
        Number of External Brains : 1
        Lesson number : 0
        Reset Parameters :

Unity brain name: BananaBrain
        Number of Visual Observations (per agent): 0
        Vector Observation space type: continuous
        Vector Observation space size (per agent): 37
        Number of stacked Vector Observation: 1
        Vector Action space type: discrete
        Vector Action space size (per agent): 4
        Vector Action descriptions: , , ,
```

Environments contain ***brains*** which are responsible for deciding the actions of their associated agents. Here we check for the first brain available, and set it as the default brain we will be controlling from Python.

```
In [3]: # get the default brain
        brain_name = env.brain_names[0]
        # print(brain_name)
```

```
# academy_name = env.academy_name[1]
# print(academy_name)
brain = env.brains[brain_name]
# print(brain)
```

1.0.2 2. Examine the State and Action Spaces

Run the code cell below to print some information about the environment.

```
In [4]: # reset the environment
        env_info = env.reset(train_mode=True)[brain_name]

        # number of agents in the environment
        print('Number of agents:', len(env_info.agents))

        # number of actions
        action_size = brain.vector_action_space_size
        print('Number of actions:', action_size)

        # examine the state space
        state = env_info.vector_observations[0]
        print('States look like:', state)
        state_size = len(state)
        print('States have length:', state_size)
```

```
Number of agents: 1
Number of actions: 4
States look like: [ 1.          0.          0.          0.          0.84408134  0.          0.
  1.          0.          0.0748472   0.          1.          0.          0.
  0.25755     1.          0.          0.          0.          0.74177343
  0.          1.          0.          0.          0.25854847  0.          0.
  1.          0.          0.09355672  0.          1.          0.          0.
  0.31969345  0.          0.        ]
States have length: 37
```

1.0.3 3. Take Random Actions in the Environment

In the next code cell, you will learn how to use the Python API to control the agent and receive feedback from the environment.

Note that **in this coding environment, you will not be able to watch the agent while it is training**, and you should set `train_mode=True` to restart the environment.

```
In [ ]: # env_info = env.reset(train_mode=True)[brain_name] # reset the environment
        # state = env_info.vector_observations[0]            # get the current state
        # score = 0                                          # initialize the score
        # while True:
        #     action = np.random.randint(action_size)        # select an action
        #     env_info = env.step(action)[brain_name]        # send the action to the environmen
```

```
#     next_state = env_info.vector_observations[0]   # get the next state
#     reward = env_info.rewards[0]                   # get the reward
#     done = env_info.local_done[0]                  # see if episode has finished
#     score += reward                                # update the score
#     state = next_state                             # roll over the state to next time
#     if done:                                       # exit loop if episode finished
#         break

# print("Score: {}".format(score))
```

When finished, you can close the environment.

```
In [ ]: # env.close()
```

1.0.4 4. It's Your Turn!

Now it's your turn to train your own agent to solve the environment! A few **important notes**: - When training the environment, set `train_mode=True`, so that the line for resetting the environment looks like the following:

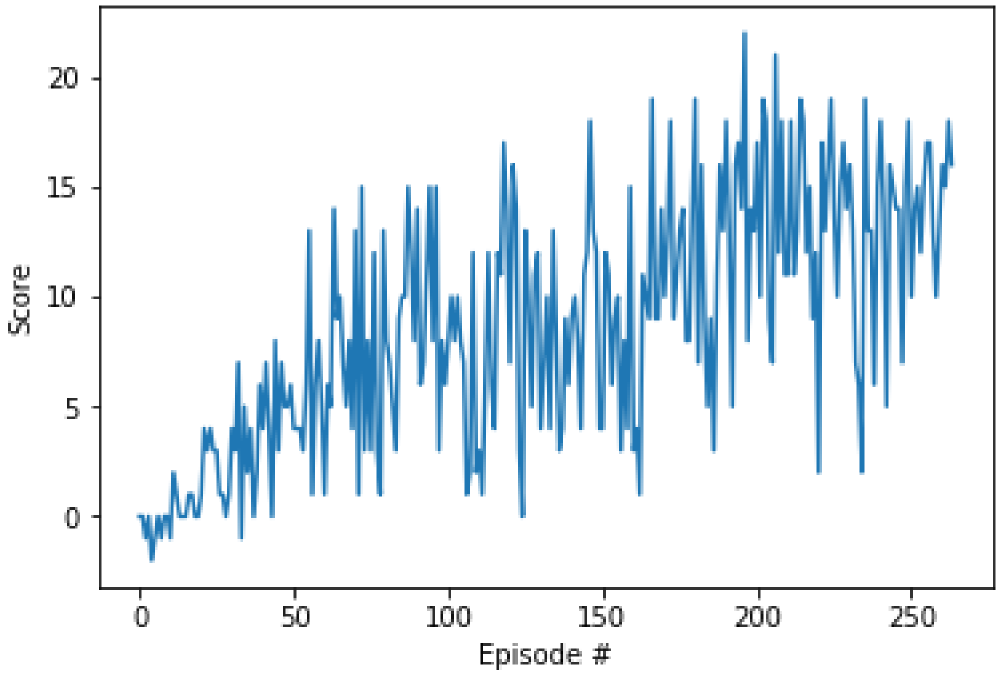
```
env_info = env.reset(train_mode=True)[brain_name]
```

- To structure your work, you're welcome to work directly in this Jupyter notebook, or you might like to start over with a new file! You can see the list of files in the workspace by clicking on ***Jupyter*** in the top left corner of the notebook.
- In this coding environment, you will not be able to watch the agent while it is training. However, ***after training the agent***, you can download the saved model weights to watch the agent on your own machine!

```
In [5]: def dqn(n_episodes=5000, max_t=10000, eps_start=1.0, eps_end=0.01, eps_decay=0.97):
            scores = []                        # list containing scores from each episode
            scores_window = deque(maxlen=100)  # last 100 scores
            eps = eps_start                    # initialize epsilon
            for i_episode in range(1, n_episodes+1):
                env_info = env.reset(train_mode=True)[brain_name]
                state = env_info.vector_observations[0]
                score = 0
                for t in range(max_t):
                    action = agent.act(state, eps)
                    env_info = env.step(action)[brain_name]        # send the action to the envi
                    next_state = env_info.vector_observations[0]   # get the next state
                    reward = env_info.rewards[0]                   # get the reward
                    done = env_info.local_done[0]

                    agent.step(state, action, reward, next_state, done)
                    state = next_state
                    score += reward
                    if done:
                        break
```

```
        scores_window.append(score)       # save most recent score
        scores.append(score)              # save most recent score
        eps = max(eps_end, eps_decay*eps) # decrease epsilon
        print('\rEpisode {}\tAverage Score: {:.2f}'.format(i_episode, np.mean(scores_win
        if i_episode % 100 == 0:
            print('\rEpisode {}\tAverage Score: {:.2f}'.format(i_episode, np.mean(scores
        if np.mean(scores_window)>=13.0:
            print('\nEnvironment solved in {:d} episodes!\tAverage Score: {:.2f}'.format
            torch.save(agent.qnetwork_local.state_dict(), 'checkpoint.pth')
            break
    return scores

agent = Agent(state_size=37, action_size=4, seed=0)
scores = dqn()

# plot the scores
fig = plt.figure()
ax = fig.add_subplot(111)
plt.plot(np.arange(len(scores)), scores)
plt.ylabel('Score')
plt.xlabel('Episode #')
plt.show()
```

```
Episode 100        Average Score: 4.76
Episode 200        Average Score: 9.51
Episode 264        Average Score: 13.00
Environment solved in 164 episodes!       Average Score: 13.00
```



```
In [ ]:
```