

Navigation

June 27, 2020

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 -q install ./python
```

```
tensorflow 1.7.1 has requirement numpy>=1.13.3, but you'll have numpy 1.12.1 which is incompatible
ipython 6.5.0 has requirement prompt-toolkit<2.0.0,>=1.0.15, but you'll have prompt-toolkit 3.0.
```

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]: from unityagents import UnityEnvironment
import numpy as np

# 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]
        brain = env.brains[brain_name]

```

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 [5]: 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 environment
            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 st
            if done:                                     # exit loop if episode finished
                break

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

Score: 0.0

1.0.4 4. Training the agent

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 [6]: from dq_n_agent import Agent
        # init the agent
        agent = Agent(state_size=state_size, action_size=action_size, seed=0)
```

1.0.5 5. Learning Algorithm

The DQN algorithm is used to train the agent. It uses a fully connected neural network (NN) layers as a function approximator for the Q function. The NN consists of two hidden layers, each with 64 units. The input layer receives the states with 37 units and the output layer has four units for the possible actions.

```

QNetwork(
    (fc1): Linear(in_features=37, out_features=64, bias=True)
    (fc2): Linear(in_features=64, out_features=64, bias=True)
    (fc3): Linear(in_features=64, out_features=4, bias=True)
)

```

The agents uses two identical architecture NNs to learn through interactions with the environment. The first, which is referred to as local QNetwork, is used to train the agent. The second, which is the target QNetwork, is used only in the error calculation to make the training process more stable.

The algorithm performs and repeats two main tasks, which are: 1. The sampling process: - The agent chooses an action from the state using a given policy. - The action is chosen using the ϵ – greedy algorithm where the next best action has the highest probability to be selected and low ϵ probability for all other actions. - The agent takes the action and receives new data from the environment. The data is in form of tuples (next_state, reward, done). - Store its experience tuple in a replay memory. - Set state to the next state. 2. The learning process: - Obtain a random batch of tuples from the replay memory. - Use the target QNetwork to get the target value. $target = rewards + (\gamma * Q_targets_next * (1 - done))$ - Use the local QNetwork to get the expected value. - Calculate the MSE between both returned values. - Do an optimization step to change the weights of the local network. - Do a soft update

Near the end and if the agent learned successfully, it will choose the best possible action at each state.

The parameters I chose for the DQN are as follows:

```

BUFFER_SIZE = int(1e5) # replay buffer size
BATCH_SIZE = 64        # minibatch size
GAMMA = 0.99           # discount factor
TAU = 1e-3             # for soft update of target parameters
LR = 5e-4              # learning rate
UPDATE_EVERY = 4       # how often to update the network

```

And for the ϵ -greedy algorithm:

```

eps_start = 1.0        # starting value of epsilon
eps_end = 0.01         # minimum value of epsilon
eps_decay = 0.995      # multiplicative factor (per episode) for decreasing epsilon

```

```

In [7]: from collections import deque
import matplotlib.pyplot as plt
import torch

```

```

def dqn(n_episodes=2000, max_t=1000, eps_start=1.0, eps_end=0.01, eps_decay=0.995):
    """Deep Q-Learning.

    Params
    =====
    n_episodes (int): maximum number of training episodes
    max_t (int): maximum number of timesteps per episode
    eps_start (float): starting value of epsilon, for epsilon-greedy action selection

```

```

    eps_end (float): minimum value of epsilon
    eps_decay (float): multiplicative factor (per episode) for decreasing epsilon
"""
best_score = 13.0
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]

        next_state = env_info.vector_observations[0]
        reward = env_info.rewards[0]
        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_window)))
    if i_episode % 100 == 0:
        print('\rEpisode {} \tAverage Score: {:.2f}'.format(i_episode, np.mean(scores_window)))
    if np.mean(scores_window) >= best_score:
        print('\nEnvironment solved in {:d} episodes! \tAverage Score: {:.2f}'.format(i_episode, np.mean(scores_window)))
        best_score = np.mean(scores_window)
        torch.save(agent.qnetwork_local.state_dict(), 'checkpoint.pth')
return scores

```

```
scores = dqn()
```

Episode 100	Average Score: 0.95	
Episode 200	Average Score: 4.79	
Episode 300	Average Score: 7.36	
Episode 400	Average Score: 10.17	
Episode 496	Average Score: 13.01	
Environment solved in 396 episodes!		Average Score: 13.01
Episode 497	Average Score: 13.01	
Environment solved in 397 episodes!		Average Score: 13.01
Episode 498	Average Score: 13.09	
Environment solved in 398 episodes!		Average Score: 13.09

Episode 499	Average Score: 13.09	
Environment solved in 399 episodes!		Average Score: 13.09
Episode 500	Average Score: 13.24	
Environment solved in 400 episodes!		Average Score: 13.24
Episode 503	Average Score: 13.28	
Environment solved in 403 episodes!		Average Score: 13.28
Episode 504	Average Score: 13.33	
Environment solved in 404 episodes!		Average Score: 13.33
Episode 505	Average Score: 13.35	
Environment solved in 405 episodes!		Average Score: 13.35
Episode 511	Average Score: 13.37	
Environment solved in 411 episodes!		Average Score: 13.37
Episode 514	Average Score: 13.41	
Environment solved in 414 episodes!		Average Score: 13.41
Episode 515	Average Score: 13.42	
Environment solved in 415 episodes!		Average Score: 13.42
Episode 516	Average Score: 13.43	
Environment solved in 416 episodes!		Average Score: 13.43
Episode 538	Average Score: 13.43	
Environment solved in 438 episodes!		Average Score: 13.43
Episode 566	Average Score: 13.46	
Environment solved in 466 episodes!		Average Score: 13.46
Episode 567	Average Score: 13.48	
Environment solved in 467 episodes!		Average Score: 13.48
Episode 568	Average Score: 13.52	
Environment solved in 468 episodes!		Average Score: 13.52
Episode 577	Average Score: 13.53	
Environment solved in 477 episodes!		Average Score: 13.53
Episode 578	Average Score: 13.53	
Environment solved in 478 episodes!		Average Score: 13.53
Episode 579	Average Score: 13.55	
Environment solved in 479 episodes!		Average Score: 13.55
Episode 586	Average Score: 13.61	
Environment solved in 486 episodes!		Average Score: 13.61
Episode 587	Average Score: 13.65	
Environment solved in 487 episodes!		Average Score: 13.65
Episode 588	Average Score: 13.72	
Environment solved in 488 episodes!		Average Score: 13.72
Episode 589	Average Score: 13.76	
Environment solved in 489 episodes!		Average Score: 13.76
Episode 590	Average Score: 13.81	
Environment solved in 490 episodes!		Average Score: 13.81
Episode 596	Average Score: 13.83	
Environment solved in 496 episodes!		Average Score: 13.83
Episode 599	Average Score: 13.86	
Environment solved in 499 episodes!		Average Score: 13.86
Episode 600	Average Score: 13.75	

Episode 610	Average Score: 13.90	
Environment solved in 510 episodes!		Average Score: 13.90
Episode 612	Average Score: 13.95	
Environment solved in 512 episodes!		Average Score: 13.95
Episode 613	Average Score: 13.96	
Environment solved in 513 episodes!		Average Score: 13.96
Episode 621	Average Score: 13.99	
Environment solved in 521 episodes!		Average Score: 13.99
Episode 622	Average Score: 14.07	
Environment solved in 522 episodes!		Average Score: 14.07
Episode 623	Average Score: 14.08	
Environment solved in 523 episodes!		Average Score: 14.08
Episode 624	Average Score: 14.09	
Environment solved in 524 episodes!		Average Score: 14.09
Episode 625	Average Score: 14.13	
Environment solved in 525 episodes!		Average Score: 14.13
Episode 626	Average Score: 14.19	
Environment solved in 526 episodes!		Average Score: 14.19
Episode 627	Average Score: 14.23	
Environment solved in 527 episodes!		Average Score: 14.23
Episode 629	Average Score: 14.24	
Environment solved in 529 episodes!		Average Score: 14.24
Episode 635	Average Score: 14.27	
Environment solved in 535 episodes!		Average Score: 14.27
Episode 636	Average Score: 14.32	
Environment solved in 536 episodes!		Average Score: 14.32
Episode 638	Average Score: 14.34	
Environment solved in 538 episodes!		Average Score: 14.34
Episode 639	Average Score: 14.38	
Environment solved in 539 episodes!		Average Score: 14.38
Episode 640	Average Score: 14.48	
Environment solved in 540 episodes!		Average Score: 14.48
Episode 641	Average Score: 14.53	
Environment solved in 541 episodes!		Average Score: 14.53
Episode 643	Average Score: 14.54	
Environment solved in 543 episodes!		Average Score: 14.54
Episode 644	Average Score: 14.61	
Environment solved in 544 episodes!		Average Score: 14.61
Episode 647	Average Score: 14.70	
Environment solved in 547 episodes!		Average Score: 14.70
Episode 648	Average Score: 14.75	
Environment solved in 548 episodes!		Average Score: 14.75
Episode 656	Average Score: 14.76	
Environment solved in 556 episodes!		Average Score: 14.76
Episode 657	Average Score: 14.78	
Environment solved in 557 episodes!		Average Score: 14.78
Episode 659	Average Score: 14.82	
Environment solved in 559 episodes!		Average Score: 14.82

Episode 660	Average Score: 14.87	
Environment solved in 560 episodes!		Average Score: 14.87
Episode 662	Average Score: 14.90	
Environment solved in 562 episodes!		Average Score: 14.90
Episode 664	Average Score: 14.91	
Environment solved in 564 episodes!		Average Score: 14.91
Episode 671	Average Score: 14.94	
Environment solved in 571 episodes!		Average Score: 14.94
Episode 672	Average Score: 14.95	
Environment solved in 572 episodes!		Average Score: 14.95
Episode 673	Average Score: 14.95	
Environment solved in 573 episodes!		Average Score: 14.95
Episode 674	Average Score: 14.99	
Environment solved in 574 episodes!		Average Score: 14.99
Episode 675	Average Score: 14.99	
Environment solved in 575 episodes!		Average Score: 14.99
Episode 676	Average Score: 15.01	
Environment solved in 576 episodes!		Average Score: 15.01
Episode 691	Average Score: 15.02	
Environment solved in 591 episodes!		Average Score: 15.02
Episode 700	Average Score: 14.97	
Episode 701	Average Score: 15.04	
Environment solved in 601 episodes!		Average Score: 15.04
Episode 702	Average Score: 15.04	
Environment solved in 602 episodes!		Average Score: 15.04
Episode 706	Average Score: 15.04	
Environment solved in 606 episodes!		Average Score: 15.04
Episode 708	Average Score: 15.09	
Environment solved in 608 episodes!		Average Score: 15.09
Episode 709	Average Score: 15.15	
Environment solved in 609 episodes!		Average Score: 15.15
Episode 714	Average Score: 15.16	
Environment solved in 614 episodes!		Average Score: 15.16
Episode 715	Average Score: 15.18	
Environment solved in 615 episodes!		Average Score: 15.18
Episode 717	Average Score: 15.21	
Environment solved in 617 episodes!		Average Score: 15.21
Episode 718	Average Score: 15.21	
Environment solved in 618 episodes!		Average Score: 15.21
Episode 719	Average Score: 15.23	
Environment solved in 619 episodes!		Average Score: 15.23
Episode 720	Average Score: 15.29	
Environment solved in 620 episodes!		Average Score: 15.29
Episode 745	Average Score: 15.37	
Environment solved in 645 episodes!		Average Score: 15.37
Episode 746	Average Score: 15.42	
Environment solved in 646 episodes!		Average Score: 15.42
Episode 752	Average Score: 15.42	

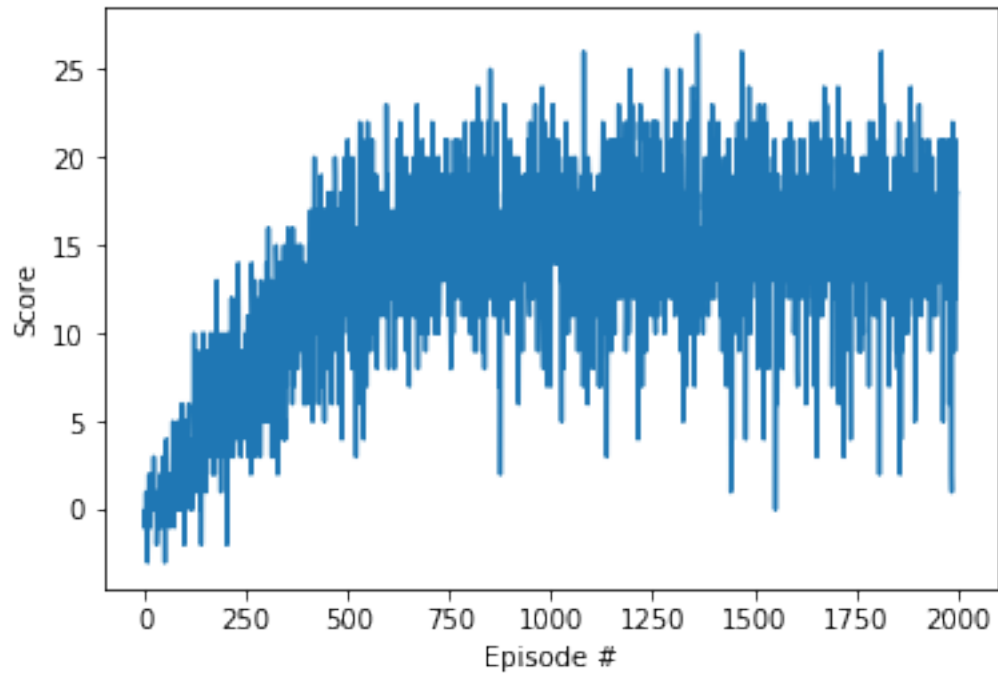
Environment solved in 652 episodes!	Average Score: 15.42
Episode 777	Average Score: 15.42
Environment solved in 677 episodes!	Average Score: 15.42
Episode 778	Average Score: 15.44
Environment solved in 678 episodes!	Average Score: 15.44
Episode 779	Average Score: 15.51
Environment solved in 679 episodes!	Average Score: 15.51
Episode 780	Average Score: 15.52
Environment solved in 680 episodes!	Average Score: 15.52
Episode 781	Average Score: 15.52
Environment solved in 681 episodes!	Average Score: 15.52
Episode 782	Average Score: 15.54
Environment solved in 682 episodes!	Average Score: 15.54
Episode 783	Average Score: 15.54
Environment solved in 683 episodes!	Average Score: 15.54
Episode 794	Average Score: 15.55
Environment solved in 694 episodes!	Average Score: 15.55
Episode 798	Average Score: 15.57
Environment solved in 698 episodes!	Average Score: 15.57
Episode 799	Average Score: 15.60
Environment solved in 699 episodes!	Average Score: 15.60
Episode 800	Average Score: 15.57
Episode 803	Average Score: 15.60
Environment solved in 703 episodes!	Average Score: 15.60
Episode 804	Average Score: 15.65
Environment solved in 704 episodes!	Average Score: 15.65
Episode 805	Average Score: 15.68
Environment solved in 705 episodes!	Average Score: 15.68
Episode 807	Average Score: 15.68
Environment solved in 707 episodes!	Average Score: 15.68
Episode 809	Average Score: 15.70
Environment solved in 709 episodes!	Average Score: 15.70
Episode 810	Average Score: 15.74
Environment solved in 710 episodes!	Average Score: 15.74
Episode 821	Average Score: 15.80
Environment solved in 721 episodes!	Average Score: 15.80
Episode 826	Average Score: 15.81
Environment solved in 726 episodes!	Average Score: 15.81
Episode 828	Average Score: 15.82
Environment solved in 728 episodes!	Average Score: 15.82
Episode 829	Average Score: 15.88
Environment solved in 729 episodes!	Average Score: 15.88
Episode 830	Average Score: 15.88
Environment solved in 730 episodes!	Average Score: 15.88
Episode 831	Average Score: 15.89
Environment solved in 731 episodes!	Average Score: 15.89
Episode 835	Average Score: 15.90
Environment solved in 735 episodes!	Average Score: 15.90

Episode 836	Average Score: 15.91	
Environment solved in 736 episodes!		Average Score: 15.91
Episode 839	Average Score: 15.92	
Environment solved in 739 episodes!		Average Score: 15.92
Episode 848	Average Score: 15.92	
Environment solved in 748 episodes!		Average Score: 15.92
Episode 849	Average Score: 15.93	
Environment solved in 749 episodes!		Average Score: 15.93
Episode 850	Average Score: 15.98	
Environment solved in 750 episodes!		Average Score: 15.98
Episode 851	Average Score: 16.05	
Environment solved in 751 episodes!		Average Score: 16.05
Episode 853	Average Score: 16.14	
Environment solved in 753 episodes!		Average Score: 16.14
Episode 854	Average Score: 16.22	
Environment solved in 754 episodes!		Average Score: 16.22
Episode 857	Average Score: 16.22	
Environment solved in 757 episodes!		Average Score: 16.22
Episode 858	Average Score: 16.24	
Environment solved in 758 episodes!		Average Score: 16.24
Episode 859	Average Score: 16.31	
Environment solved in 759 episodes!		Average Score: 16.31
Episode 900	Average Score: 15.75	
Episode 1000	Average Score: 15.70	
Episode 1100	Average Score: 15.24	
Episode 1200	Average Score: 15.56	
Episode 1300	Average Score: 15.90	
Episode 1400	Average Score: 15.80	
Episode 1500	Average Score: 15.75	
Episode 1600	Average Score: 15.63	
Episode 1700	Average Score: 15.58	
Episode 1800	Average Score: 15.85	
Episode 1900	Average Score: 15.38	
Episode 2000	Average Score: 15.51	

<matplotlib.figure.Figure at 0x7f87d5223b38>

Clearly, the agent was able to solve the environment in episode 396. But it achieved its best score in episode 759.

```
In [8]: # 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()
```



When finished, you can close the environment.

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

Although the model has a great score and performance, but it can be further improved by trying other algorithms like: - Double DQN - Prioritized Experience Replay - Dueling DQN - Pixel Based Learning

```
In [ ]:
```