Udacity

Deep Reinforcement Leaning Nanodegree

Project – Navigation Report

Training

For each training session, we construct the agent with different parameters and we run the Deep-Q-Network procedure dqn as follows:

agent = Agent(state_size=37, action_size=4, seed=1, fc1_units=fc1_nodes, fc2_units=fc2_nodes)

*scores, episodes = **dqn**(n_episodes = 2000, eps_start = epsilon_start, train_numb=i)*

The obtained weights for each session are saved into a file.

Deep-Q-Network algorithm

The Deep-Q-Network procedure dqn performs the external loop (by episodes_) till the number of episodes reached the maximal number of episodes nepisodes = 2000 or the completion criteria is executed. For the completion criteria, we check

np.mean(scores_window) >= 13,

where scores___window is the array of the type deque realizing the shifting window of length <= 100. The element scores___window_[i] contains the _score achieved by the algorithm on the episode i.

In the internal loop, dqn gets the current action from the agent. By this action dqn gets state and reward from Unity environment env. Then, the agent accept params state_, _action_, _reward_, _next___state_, _done to the next training step. The variable score accumulates obtained rewards.

Mechanisms of Agent

The class Agent is class implementing the following mechanisms:

- **Two Q-Networks (local and target) using the simple neural network.**

self.qnetwork_local = QNetwork(state_size, action_size, seed, fc1_units, fc2_units).to(device)

self.qnetwork_target = QNetwork(state_size, action_size, seed, fc1_units, fc2_units).to(device)

- **Replay memory (using the class ReplayBuffer)**

```
self.experience = namedtuple("Experience", field_names=["state", "action", "reward", "next_state", "done"])

...

e = self.experience(state, action, reward, next_state, done)

self.memory.append(e)
```

- **Epsilon-greedy mechanism**

```
if random.random() > eps:
    return np.argmax(action_values.cpu().data.numpy())
else:
    return random.choice(np.arange(self.action_size))
```

The epsilon become a bit smaller with each episode:

*eps = max(eps_end, eps_decay*eps),*

where eps_end=0.01, eps_decay = 0.996.

- **Q-learning, i.e., using the max value for all possible actions**
- **Computing the loss function by MSE loss**
- **Minimize the loss by gradient descend mechanism using the ADAM optimizer**

Model Q-Network

Both Q-Networks (local and target) are implemented by the class QNetwork. This class implements the simple neural network

with 3 fully-connected layers and 2 rectified nonlinear layers. The class QNetwork is realized in the framework of package PyTorch.

The number of neurons of the fully-connected layers are as follows:

- Layer fc1, number of neurons: state___size x _fc1___units_,
- Layer fc2, number of neurons: fc1___units x _fc2___units_,
- Layer fc3, number of neurons: fc2___units x _action___size_,

where state___size = 37, action___size = 8, fc1___units and fc2___units are the input params.

Training and Testing

We run 8 training sessions with different parameters _fc1___units_, _fc2___units_, _eps___start_, and we save obtained weights by the function of PyTorch:

torch.save(agent.qnetwork_local.state_dict(), 'weights_'+str(train_numb)+'.trn')

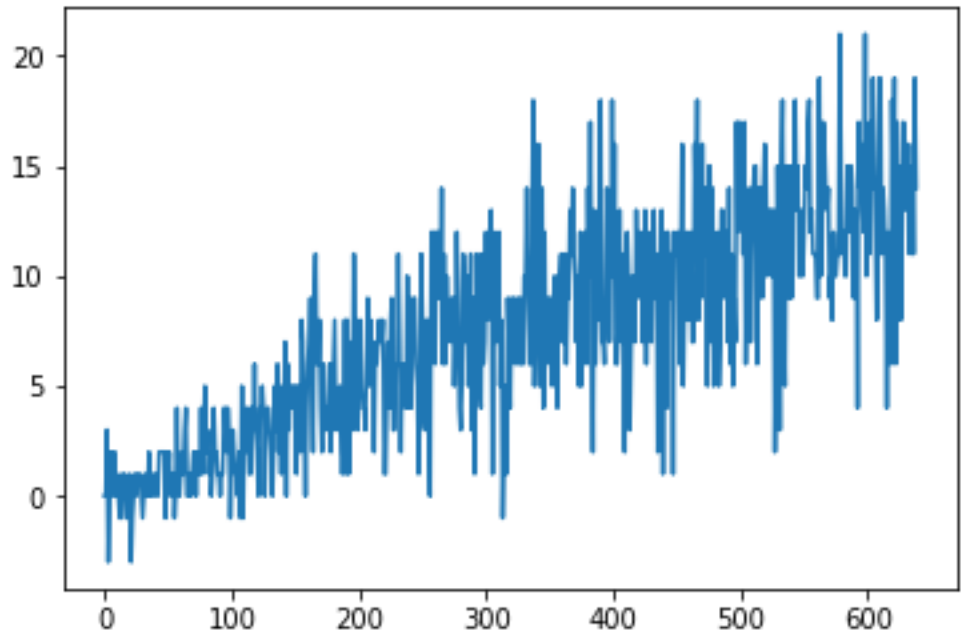
For each session we get following output

1. fc1_units: 54 , fc2_units: 46

- train_numb: 0 eps_start: 0.99

- Episode: 639, elapsed: 0:14:11.673203, Avg.Score: 13.00, score 14.0, How many scores >= 13: 53, eps.: 0.08

- terminating at episode : 639 ave reward reached +13 over 100 episodes

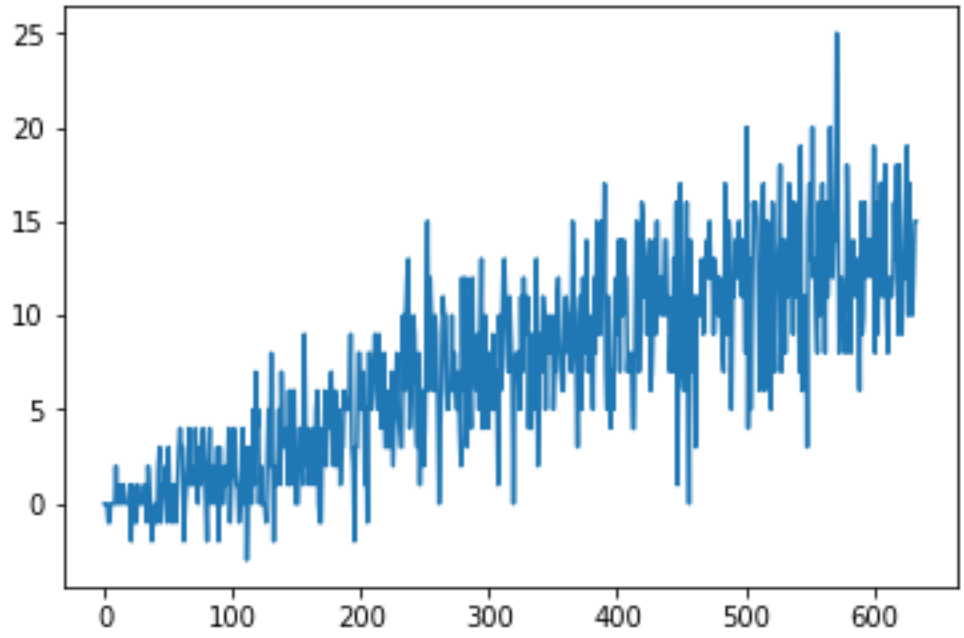


2. fc1_units: 118 , fc2_units: 118

- train_numb: 1 eps_start: 0.992

- Episode: 634, elapsed: 0:14:09.634947, Avg.Score: 13.01, score 15.0, How many scores >= 13: 53, eps.: 0.08

- terminating at episode : 634 ave reward reached +13 over 100 episodes

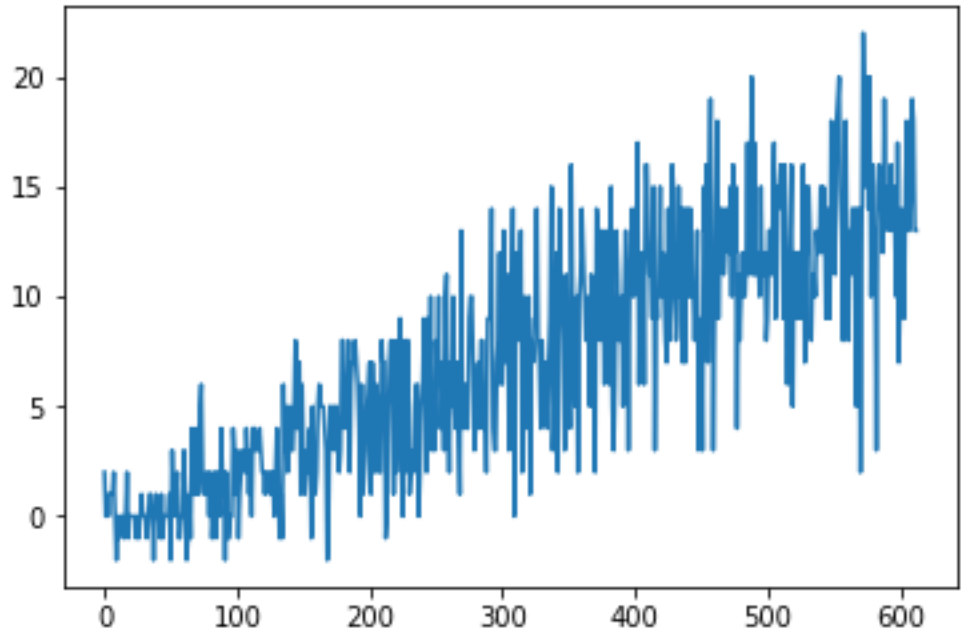


3. fc1_units: 54 , fc2_units: 46

- train_numb: 2 eps_start: 0.993

- Episode: 613, elapsed: 0:13:40.427706, Avg.Score: 13.00, score 13.0, How many scores >= 13: 59, eps.: 0.09

- terminating at episode : 613 ave reward reached +13 over 100 episodes

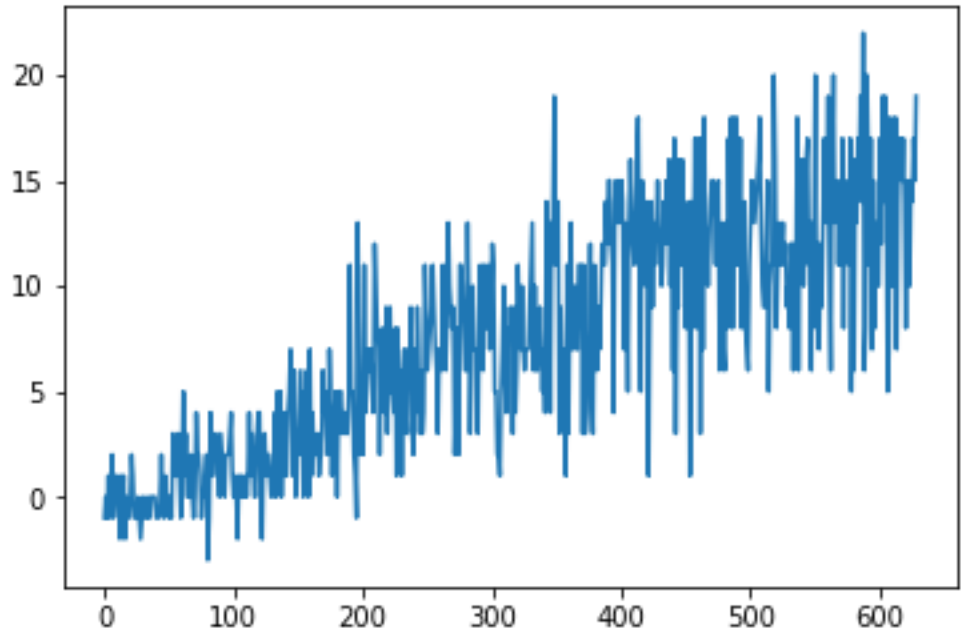


4. fc1_units: 54 , fc2_units: 46

- train_numb: 3 eps_start: 0.988

- Episode: 629, elapsed: 0:14:01.435002, Avg.Score: 13.00, score 19.0, How many scores >= 13: 58, eps.: 0.08

- terminating at episode : 629 ave reward reached +13 over 100 episodes

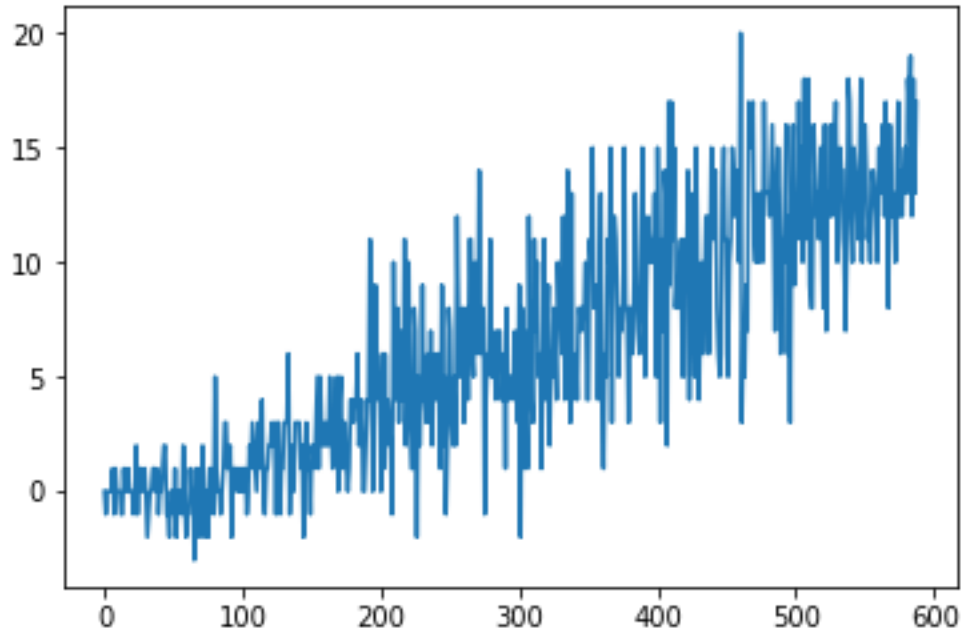


5. fc1_units: 54 , fc2_units: 54

- train_numb: 4 eps_start: 0.991

- Episode: 588, elapsed: 0:13:10.050657, Avg.Score: 13.00, score 17.0, How many scores >= 13: 57, eps.: 0.09

- terminating at episode : 588 ave reward reached +13 over 100 episodes

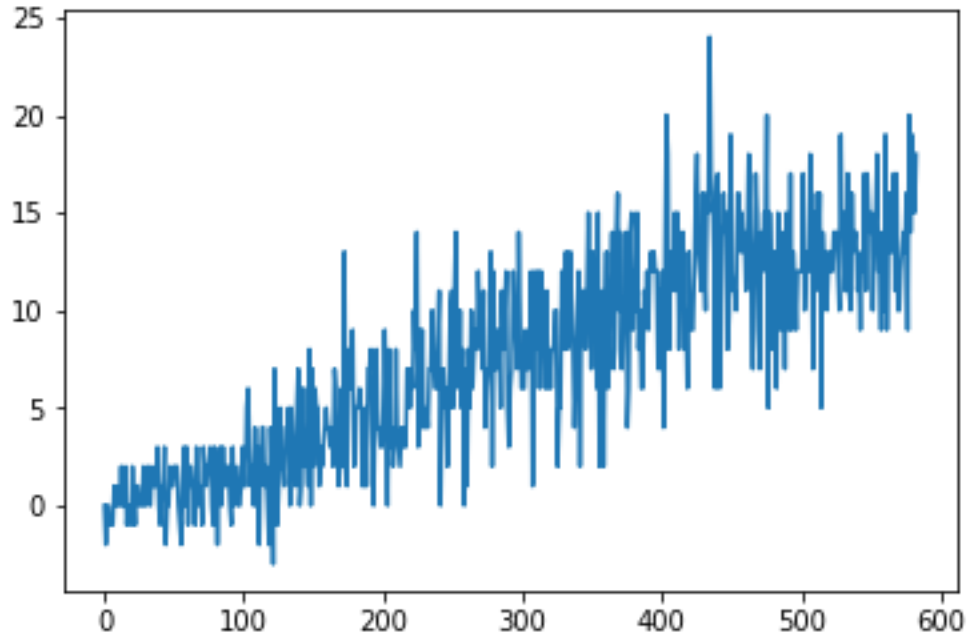


6. fc1_units: 54 , fc2_units: 46

- train_numb: 5 eps_start: 0.992

- Episode: 583, elapsed: 0:13:04.717450, Avg.Score: 13.05, score 18.0, How many scores >= 13: 55, eps.: 0.10

- terminating at episode : 583 ave reward reached +13 over 100 episodes

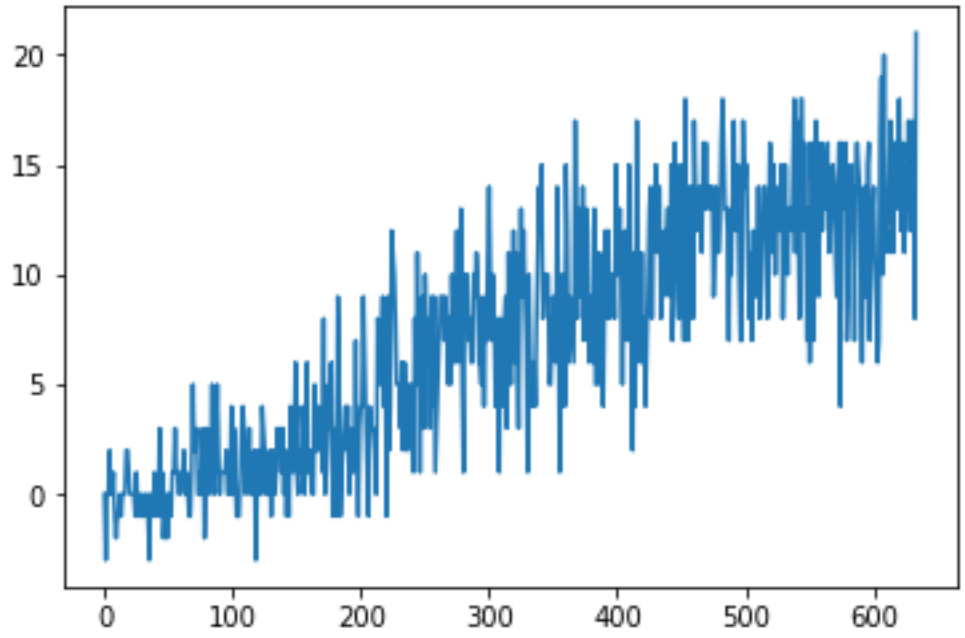


7. fc1_units: 118 , fc2_units: 110

- train_numb: 6 eps_start: 0.988

- Episode: 634, elapsed: 0:14:14.761383, Avg.Score: 13.05, score 21.0, How many scores >= 13: 60, eps.: 0.08

- terminating at episode : 634 ave reward reached +13 over 100 episodes

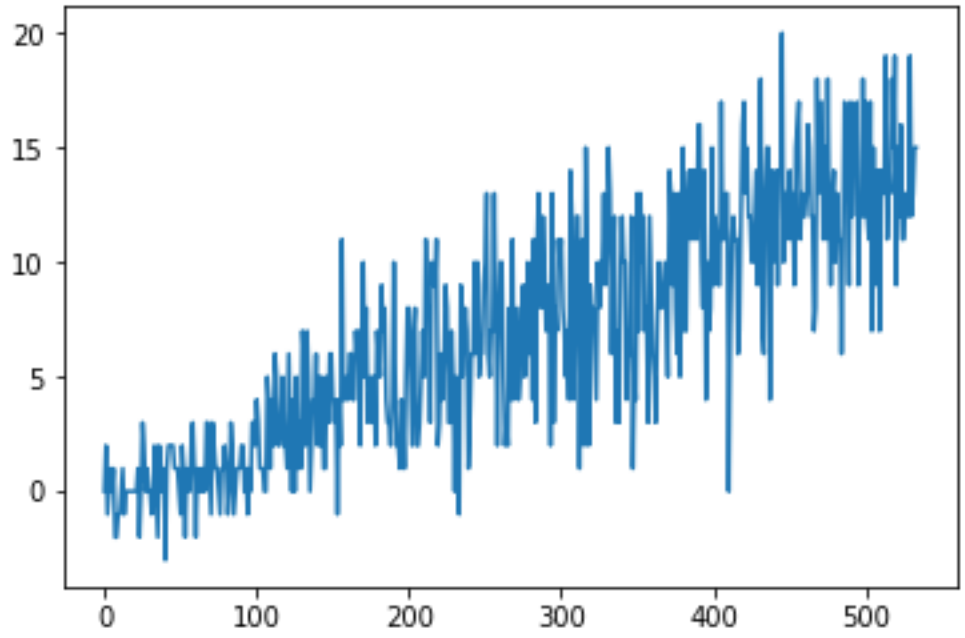


8. fc1_units: 86 , fc2_units: 78

- train_numb: 7 eps_start: 0.991

- Episode: 534, elapsed: 0:11:59.747946, Avg.Score: 13.03, score 15.0, How many scores >= 13: 57, eps.: 0.12

- terminating at episode : 534 ave reward reached +13 over 100 episodes



Test average score

For each testing session, we run checkWeights to get the average score for the given set of parameters. Following are the results of the testing

Train: 0, Test: 0, Episode: 639, fc1_units: 54, fc2_units: 46, eps_start: 0.99, Score: 8.0

Train: 0, Test: 1, Episode: 639, fc1_units: 54, fc2_units: 46, eps_start: 0.99, Score: 7.0

Train: 0, Test: 2, Episode: 639, fc1_units: 54, fc2_units: 46, eps_start: 0.99, Score: 6.0

Train: 0, Test: 3, Episode: 639, fc1_units: 54, fc2_units: 46, eps_start: 0.99, Score: 12.0

Train: 0, Test: 4, Episode: 639, fc1_units: 54, fc2_units: 46, eps_start: 0.99, Score: 10.0

Train: 0, Test: 5, Episode: 639, fc1_units: 54, fc2_units: 46, eps_start: 0.99, Score: 13.0

Average Score: 9.33333333333

Train: 1, Test: 0, Episode: 634, fc1_units: 118, fc2_units: 118, eps_start: 0.992, Score: 12.0

Train: 1, Test: 1, Episode: 634, fc1_units: 118, fc2_units: 118, eps_start: 0.992, Score: 18.0

Train: 1, Test: 2, Episode: 634, fc1_units: 118, fc2_units: 118, eps_start: 0.992, Score: 14.0

Train: 1, Test: 3, Episode: 634, fc1_units: 118, fc2_units: 118, eps_start: 0.992, Score: 15.0

Train: 1, Test: 4, Episode: 634, fc1_units: 118, fc2_units: 118, eps_start: 0.992, Score: 13.0

Train: 1, Test: 5, Episode: 634, fc1_units: 118, fc2_units: 118, eps_start: 0.992, Score: 19.0

Average Score: 15.1666666667

Train: 2, Test: 0, Episode: 613, fc1_units: 54, fc2_units: 46, eps_start: 0.993, Score: 14.0

Train: 2, Test: 1, Episode: 613, fc1_units: 54, fc2_units: 46, eps_start: 0.993, Score: 20.0

Train: 2, Test: 2, Episode: 613, fc1_units: 54, fc2_units: 46, eps_start: 0.993, Score: 19.0

Train: 2, Test: 3, Episode: 613, fc1_units: 54, fc2_units: 46, eps_start: 0.993, Score: 19.0

Train: 2, Test: 4, Episode: 613, fc1_units: 54, fc2_units: 46, eps_start: 0.993, Score: 19.0

Train: 2, Test: 5, Episode: 613, fc1_units: 54, fc2_units: 46, eps_start: 0.993, Score: 15.0

Average Score: 17.6666666667

Train: 3, Test: 0, Episode: 629, fc1_units: 54, fc2_units: 46, eps_start: 0.988, Score: 7.0

Train: 3, Test: 1, Episode: 629, fc1_units: 54, fc2_units: 46, eps_start: 0.988, Score: 12.0

Train: 3, Test: 2, Episode: 629, fc1_units: 54, fc2_units: 46, eps_start: 0.988, Score: 18.0

Train: 3, Test: 3, Episode: 629, fc1_units: 54, fc2_units: 46, eps_start: 0.988, Score: 22.0

Train: 3, Test: 4, Episode: 629, fc1_units: 54, fc2_units: 46, eps_start: 0.988, Score: 17.0

Train: 3, Test: 5, Episode: 629, fc1_units: 54, fc2_units: 46, eps_start: 0.988, Score: 17.0

Average Score: 15.5

Train: 4, Test: 0, Episode: 588, fc1_units: 54, fc2_units: 54, eps_start: 0.991, Score: 11.0

Train: 4, Test: 1, Episode: 588, fc1_units: 54, fc2_units: 54, eps_start: 0.991, Score: 8.0

Train: 4, Test: 2, Episode: 588, fc1_units: 54, fc2_units: 54, eps_start: 0.991, Score: 12.0

Train: 4, Test: 3, Episode: 588, fc1_units: 54, fc2_units: 54, eps_start: 0.991, Score: 16.0

Train: 4, Test: 4, Episode: 588, fc1_units: 54, fc2_units: 54, eps_start: 0.991, Score: 14.0

Train: 4, Test: 5, Episode: 588, fc1_units: 54, fc2_units: 54, eps_start: 0.991, Score: 16.0

Average Score: 12.8333333333

Train: 5, Test: 0, Episode: 583, fc1_units: 54, fc2_units: 46, eps_start: 0.992, Score: 15.0

Train: 5, Test: 1, Episode: 583, fc1_units: 54, fc2_units: 46, eps_start: 0.992, Score: 12.0

Train: 5, Test: 2, Episode: 583, fc1_units: 54, fc2_units: 46, eps_start: 0.992, Score: 13.0

Train: 5, Test: 3, Episode: 583, fc1_units: 54, fc2_units: 46, eps_start: 0.992, Score: 16.0

Train: 5, Test: 4, Episode: 583, fc1_units: 54, fc2_units: 46, eps_start: 0.992, Score: 13.0

Train: 5, Test: 5, Episode: 583, fc1_units: 54, fc2_units: 46, eps_start: 0.992, Score: 15.0

Average Score: 14.0

Train: 6, Test: 0, Episode: 634, fc1_units: 118, fc2_units: 110, eps_start: 0.988, Score: 12.0

Train: 6, Test: 1, Episode: 634, fc1_units: 118, fc2_units: 110, eps_start: 0.988, Score: 16.0

Train: 6, Test: 2, Episode: 634, fc1_units: 118, fc2_units: 110, eps_start: 0.988, Score: 14.0

Train: 6, Test: 3, Episode: 634, fc1_units: 118, fc2_units: 110, eps_start: 0.988, Score: 14.0

Train: 6, Test: 4, Episode: 634, fc1_units: 118, fc2_units: 110, eps_start: 0.988, Score: 19.0

Train: 6, Test: 5, Episode: 634, fc1_units: 118, fc2_units: 110, eps_start: 0.988, Score: 20.0

Average Score: 15.8333333333

Train: 7, Test: 0, Episode: 534, fc1_units: 86, fc2_units: 78, eps_start: 0.991, Score: 14.0

Train: 7, Test: 1, Episode: 534, fc1_units: 86, fc2_units: 78, eps_start: 0.991, Score: 16.0

Train: 7, Test: 2, Episode: 534, fc1_units: 86, fc2_units: 78, eps_start: 0.991, Score: 6.0

Train: 7, Test: 3, Episode: 534, fc1_units: 86, fc2_units: 78, eps_start: 0.991, Score: 18.0

Train: 7, Test: 4, Episode: 534, fc1_units: 86, fc2_units: 78, eps_start: 0.991, Score: 12.0

Train: 7, Test: 5, Episode: 534, fc1_units: 86, fc2_units: 78, eps_start: 0.991, Score: 15.0

Average Score: 13.5

Future Improvements Ideas

The future ideas for improving the agent's performance.

Possible improve can be achieved by adding one or more nonlinear (or also linear) layers to the neural network. The starting value of epsilon should be parameter for improvement.