Project 2: Continous Control - Report

Introduction

The goal of this project ist o train an agent to solve the Reacher enviroment. In this environment, a double-jointed arm can move to target locations. A reward of +0.1 is provided for each step that the agent's hand is in the goal location. Thus, the goal of your agent is to maintain its position at the target location for as many time steps as possible. The observation space consists of 33 variables corresponding to position, rotation, velocity, and angular velocities of the arm. Each action is a vector with four numbers, corresponding to torque applicable to two joints. Every entry in the action vector should be a number between -1 and 1.

Model

I used an actor and a critic model. Both are Deep Neural Networks (DNN) with 2 hidden layers. Both of them are activated with the relu function, in order to introduce non-linearity into the model. Further, both DNNs have an input layer with 33 neurons, matching the state space.

The hidden layers of the actor model have 400 and 300 neurons. The output of the actor model has 4 neurons, matching the actions space.

The hidden layers of the critic model have a baseline of 128 neurons each. However, the input of the second hidden layer gets compounded with the 4 output neurons of the actor network resulting in 132 neurons for the second hidden layer. The output layer of the critic network has 1 neuron representing the target value.

Algorithm

To train the agent, I used the DDPG algorithm, which implements a deterministic policy for a continuous action space. DDPG can be seen as an actor-critical method because it uses the same structure as actor-critical methods, but the underlying mechanism is similar to the DQN algorithm. Therefore, it can also be seen as an extension of DQN to continuous actionspace. I based my implementation on the ddpg-pendulum example from the udacity github:

https://github.com/udacity/deep-reinforcement-learning/tree/master/ddpg-pendulum.

To make the code work, I changed the way data is exchanged with the environment, since there are differences between openai-gym environments and unity environments. I also extended the code to work with multiple agents in parallel. This was done by adding the experience of all agents to the replay buffer and extending the OUNoise class to generate process noise for multiple agents. From there I tried different strategies to make the agent learn. I started with the suggestion of not training every run, but ended up reversing this. The improvements that made a difference in the end are the following:

1) I changed the process noise from a uniform distribution to a normal distribution. The reason was that the uniform distribution samples between [0,1] and introduces a bias because negative numbers are not considered. I also implemented a noise reduction. So the influence of the process noise decreases over time.
2) I reduced the number of neurons in the critic network to avoid overfitting. I also added a batch normalization layer after the input layer to make training faster and more stable. The reason why this is effective is still under discussion. The authors of the original paper claim that batch normalization can mitigate the problem of internal covariance shift, but recent papers claim different reasons. Nevertheless, in my case it had a big impact on the learning speed.

I didn't change the rest of the implementation, as it already gave sufficient results. The following hyperparameters were used for training:

- Replay buffer size: 1e5
- Minibatch size: 128
- Discount factor = 0.99
- Soft update of target parameters: 1e-3
- Learning rate of the actor: 1e-3
- Learning rate of the critic: 1e-3
- L2 weight decay: 0
-

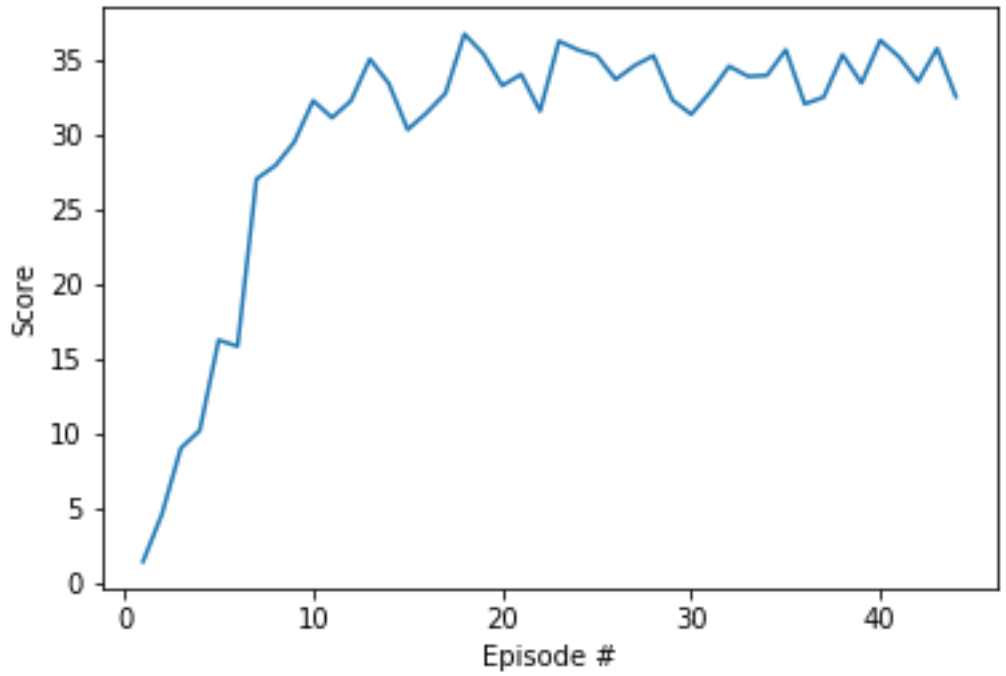
Results

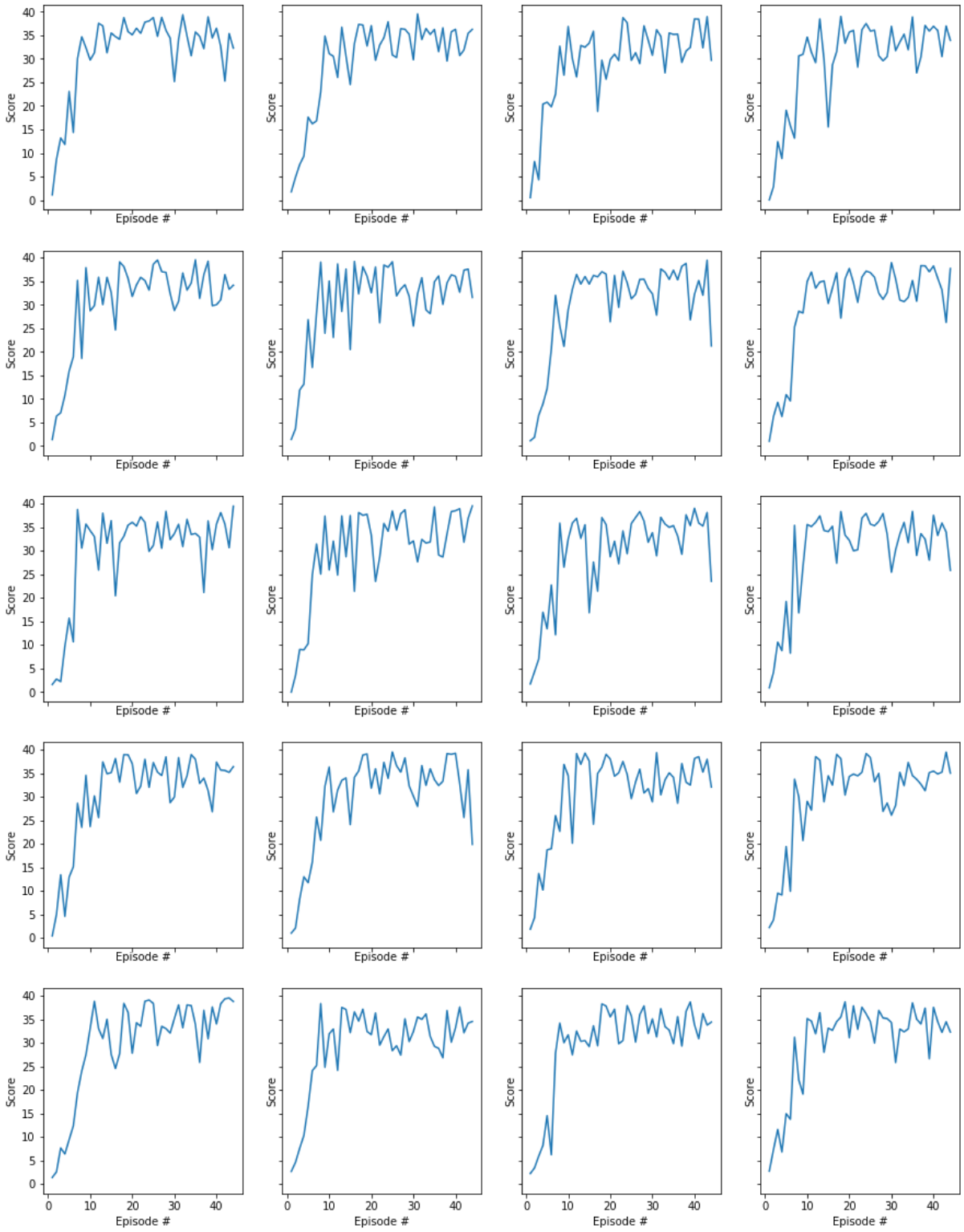
With the described network and training algorithm, an average score of 30 over all 20 Agents over 100 consecutive episodes was reached after 44 episodes as shown in the output below.

```
Episode 1       Average Score: 1.37     Score min: 0.00 Score max: 2.75 Score avg: 1.37
Episode 2       Average Score: 2.95     Score min: 1.87 Score max: 8.64 Score avg: 4.54
Episode 3       Average Score: 4.96     Score min: 2.21 Score max: 13.73        Score avg: 8.96
Episode 4       Average Score: 6.25     Score min: 4.62 Score max: 20.39        Score avg: 10.14
Episode 5       Average Score: 8.24     Score min: 9.43 Score max: 26.87        Score avg: 16.18
Episode 6       Average Score: 9.50     Score min: 6.23 Score max: 24.91        Score avg: 15.77
Episode 7       Average Score: 11.99    Score min: 12.17        Score max: 38.82        Score avg: 26.97
Episode 8       Average Score: 13.97    Score min: 16.84        Score max: 39.09        Score avg: 27.86
Episode 9       Average Score: 15.69    Score min: 19.12        Score max: 37.93        Score avg: 29.43
Episode 10      Average Score: 17.34    Score min: 23.68        Score max: 36.88        Score avg: 32.20
Episode 11      Average Score: 18.59    Score min: 20.17        Score max: 38.85        Score avg: 31.06
Episode 12      Average Score: 19.72    Score min: 24.16        Score max: 39.21        Score avg: 32.20
Episode 13      Average Score: 20.90    Score min: 28.61        Score max: 38.48        Score avg: 34.98
Episode 14      Average Score: 21.79    Score min: 28.01        Score max: 39.31        Score avg: 33.35
Episode 15      Average Score: 22.35    Score min: 15.52        Score max: 37.64        Score avg: 30.26
Episode 16      Average Score: 22.92    Score min: 20.48        Score max: 39.23        Score avg: 31.39
Episode 17      Average Score: 23.49    Score min: 18.84        Score max: 39.14        Score avg: 32.69
Episode 18      Average Score: 24.22    Score min: 27.21        Score max: 39.03        Score avg: 36.66
Episode 19      Average Score: 24.81    Score min: 25.69        Score max: 39.12        Score avg: 35.33
Episode 20      Average Score: 25.23    Score min: 26.39        Score max: 38.04        Score avg: 33.21
Episode 21      Average Score: 25.64    Score min: 23.51        Score max: 38.05        Score avg: 33.96
Episode 22      Average Score: 25.91    Score min: 26.20        Score max: 37.26        Score avg: 31.51
Episode 23      Average Score: 26.36    Score min: 30.50        Score max: 38.86        Score avg: 36.18
Episode 24      Average Score: 26.74    Score min: 29.41        Score max: 39.23        Score avg: 35.60
Episode 25      Average Score: 27.08    Score min: 28.37        Score max: 39.56        Score avg: 35.22
Episode 26      Average Score: 27.33    Score min: 29.40        Score max: 39.54        Score avg: 33.62
Episode 27      Average Score: 27.60    Score min: 27.43        Score max: 38.83        Score avg: 34.55
Episode 28      Average Score: 27.87    Score min: 26.93        Score max: 38.74        Score avg: 35.21
Episode 29      Average Score: 28.02    Score min: 28.71        Score max: 35.25        Score avg: 32.25
Episode 30      Average Score: 28.13    Score min: 25.16        Score max: 39.00        Score avg: 31.29
Episode 31      Average Score: 28.28    Score min: 25.84        Score max: 39.56        Score avg: 32.77
Episode 32      Average Score: 28.48    Score min: 30.46        Score max: 39.39        Score avg: 34.50
Episode 33      Average Score: 28.64    Score min: 27.02        Score max: 38.09        Score avg: 33.83
Episode 34      Average Score: 28.79    Score min: 28.14        Score max: 39.01        Score avg: 33.89
Episode 35      Average Score: 28.99    Score min: 29.28        Score max: 39.61        Score avg: 35.60
Episode 36      Average Score: 29.07    Score min: 25.86        Score max: 36.16        Score avg: 31.98
Episode 37      Average Score: 29.16    Score min: 21.17        Score max: 38.34        Score avg: 32.43
Episode 38      Average Score: 29.32    Score min: 29.53        Score max: 39.29        Score avg: 35.28
Episode 39      Average Score: 29.43    Score min: 26.64        Score max: 39.05        Score avg: 33.38
Episode 40      Average Score: 29.60    Score min: 30.04        Score max: 39.28        Score avg: 36.24
Episode 41      Average Score: 29.73    Score min: 30.71        Score max: 38.97        Score avg: 35.12
Episode 42      Average Score: 29.82    Score min: 25.27        Score max: 39.38        Score avg: 33.48
Episode 43      Average Score: 29.96    Score min: 26.28        Score max: 39.56        Score avg: 35.70
Episode 44      Average Score: 30.01    Score min: 19.92        Score max: 39.56        Score avg: 32.45

Environment solved in 44 episodes!      Average Score: 30.01
```

A plot of the progression of the average score over all agents can be found below, followed by the score of each agent separately.




Score
Episode #

Future Work

For the future, it would be fun to implement some of the other actor-critic Methods like A3C, A2C or GAE to compare their performance to the DDPG algorithm. When it comes to improving my implementation of the DDPG, i would like to implement a prioritized replay buffer, as this seems to be the most promising improvement to speed up the learning process.