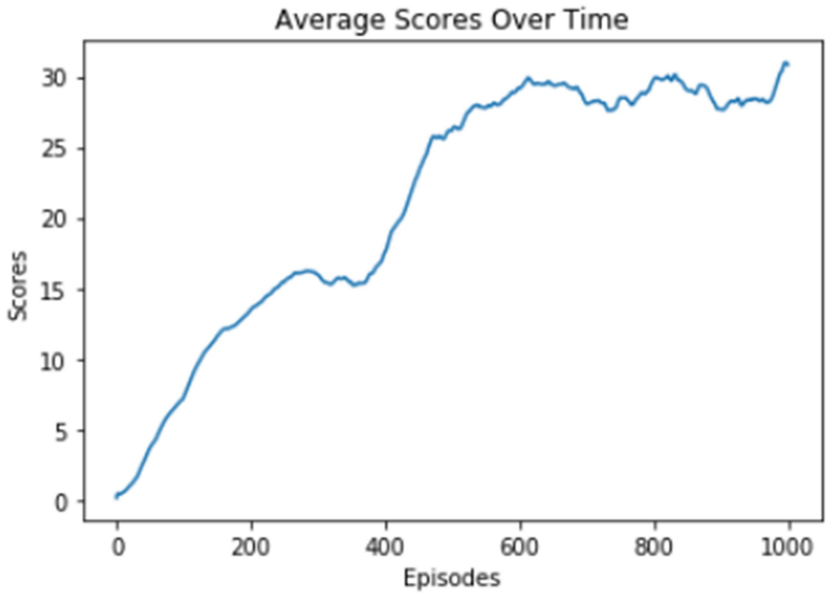
Report

```
Max Average Score: 30.98083430752438
```



This model includes an Actor Critic DDPG Asynchronous Model. There are multiple asynchronous Actors that act with each Environment Agent in the Reacher Unity space. The Agent uses a signal replay buffer to sample past experiences. Also, the Agent has a custom noise function that adds noise from a range of negative one to one and is activated by Epsilon Greedy. Finally, by looking at the graph above, the Agent was able to solve the environment, with an average over the average of agents of +30, in 620 episodes.

Learning Algorithm:

As said above the model includes an Actor Critic DDPG Asynchronous Model with multiple Actors (20) acting in the environment using a single replay buffer. The Agent gathers information from a full episode then, trains each of the Actors with a single Critic each for 50 rounds. Next, The Critic is updated by using a form of QLearning where it updates the parameters according to the future value (next states as input) of the Critic Model. Lastly, the Actor model is updated by Gradient Ascent from the mean of the Critic QValues.

Hyperparameters & Model Architectures:

The chosen hyper parameters are as follows: A buffer size of 5e5, a Batch Size of 256, Gamma at 0.99, Tau at 1e-3, learning rate for Actor 1e-4, and learning rate for Critic 1e-3. Also, a limit of 20 (using all agents in environment), an Epsilon start value of 1.0, and an Epsilon decay value of 0.986.

The Model Architectures is as follows: The Actor Local model along with its respective optimizers is duplicated for each of the 20 agents in the environment, while the Actor Target model has only one model. The Actor Model consists of a three layer neural network with 400 units for the first layer, 300 units for the second, and 4 units for the last which equals the number of actions. The activations for the layer are Relu, except for the last which is Tanh. Next, the Critic Network has a single Local and Target network. The Critic Network is also a three layer fully connected network with the first layer encoding the state. Then, the action is concatenated with the encoded state which is then feed into the second layer. There are 400 units in the first layer, 300 units in the second layer, and 1 unit in the last layer. The activations for the layers are Relu except for the last layer which is Linear.

Future Ideas for Improving the Agent's Performance:

To improve the networks performance I would first include “Importance Sampling.” This would help with training because the reward signal is sparse, and giving the Agent richer experiences from “Importance Sampling” should really benefit the model. Last, I would want to extend the buffer to also have current sampling along with random sampling. Additionally, adding current sampling would help with training time since current experiences are fresh and new, and would be prioritized a little more to increase the agents reward.