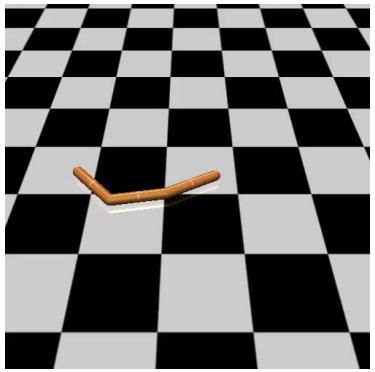
Deep Neural Networks for Policy Representation in Continuous Action domains

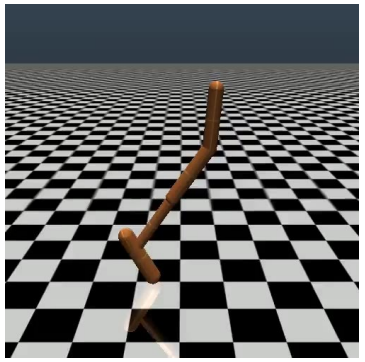
Sanjay Thakur (260722338)

Sumana Basu (260727568)

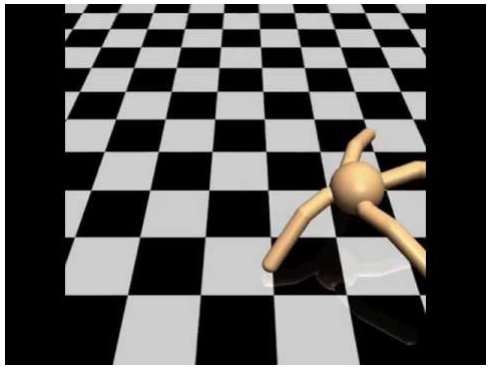
Environments:
(MuJOCO’s locomotion tasks):



Swimmer



Hopper



Ant

Why are these tasks(locomotion) challenging:

Both observation space and action space is continuous, and hence discretization is not useful.

All spaces have high degrees of freedom.

There exists a local-minima where it can decide either not to move forward at all, or fall over slowly.

Implementation

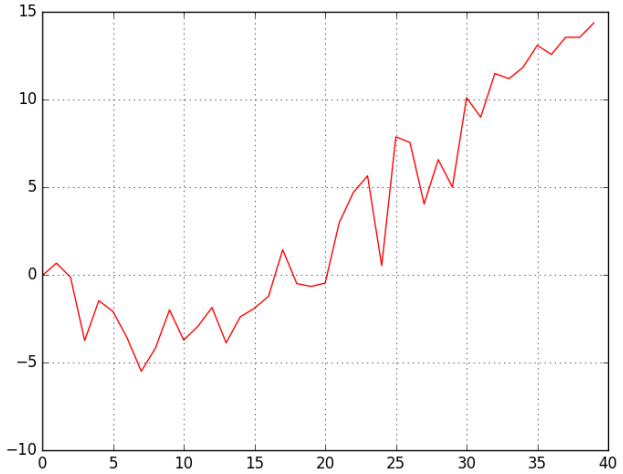
- Algorithm used: REINFORCE with baseline
- Neural Network Architecture for both Actor and Critic
  - Input – Observation from the environment
  - Output – Magnitude of the action to be taken in each dimension

Neural Network Implementation:
Used Tensorflow

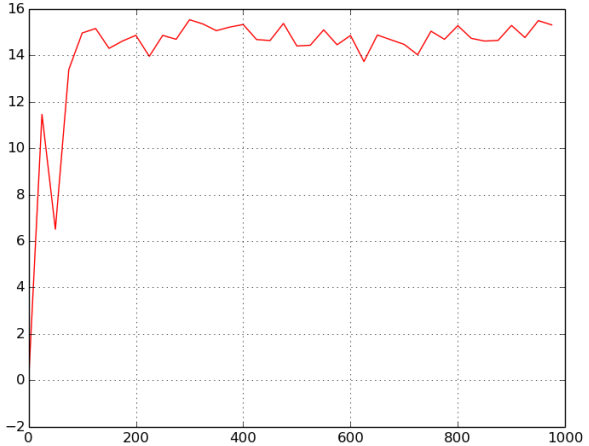
- Neural Network Architecture for both Actor and Critic
  - Input Layer (Degrees of freedom in the observation space)
  - Hidden layer 1 (100 nodes)
  - Hidden layer 2 (50 nodes)
  - Hidden layer 3 (25 nodes)
  - Final layer (Degrees of freedom in the action space)
  - Activation : ReLU
-

Results

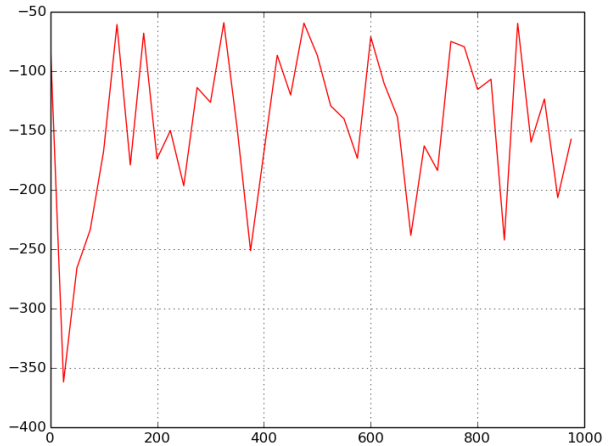
Rewards over episodes



REINFORCE on swimmer



REINFORCE on hopper



REINFORCE on Ant

Relevance

- Using Deep Learning for feature mapping frees us from environment specific feature engineering
- Most real world tasks are of high degrees of freedom
- Policy gradient techniques are better than Q-learning algorithms due to end to end structure

Scope for improvement:

Could use some smarter strategy that would mix exploration to policy improvements intrinsically.

Try using some non-gradient methods