Project Continuous Control

Amith Parameshwara

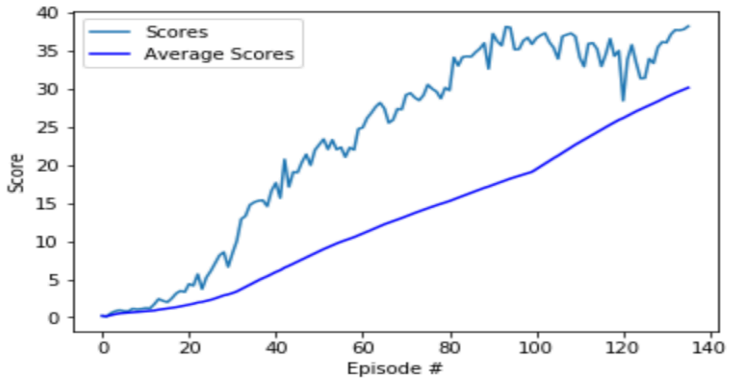
This project aims to create an agent that control a robotic arm. Goal is to let this double jointed robotic arm is maintain contact with the green sphere. Reward of +0.1 is awarded for each timestep that the robotic arm is on the green sphere. Agent needs to collect a reward of +30 for 100 consecutive episodes for the environment to be considered solved.

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.

Solution

Agent is created using the DDPG (Deep Deterministic Policy Gradient) suitable for stochastic and continuous action spaces. This agent could solve the environment in 36 episodes as below.



Algorithm and Hyperparameters

This agent uses 2-layer LSTM network for both actor and critic. LSTM is chosen with the intuition that ideal action of the agent depends on the previous action, state etc and LSTM is good to represent temporal sequences. Number of neurons is set to 256 to learn complex representation and relationship among dimensions of the state.

Learning Method: Both actor and critic are set to learn (i.e. network is updated) every 50 timesteps. Each learning step involves 5 epochs.

Learning Rate: Learning rate is chosen to 1e-3 for the actor and 3e-3 after multiple experiments. Further, learn rate scheduler is used wherein the learning rate decays by a factor of 0.1 at each epoch.

Optimizer: RMSPROP is used after trying ADAM initially. RMSPROP is found to be suitable for RNNs in many cases.

Future work

While this agent uses LSTM, ideally LSTM networks should be fed with data with sequence length of more than 1 so that temporal representations is properly utilised. This requires additional pre-processing (minor modifications to replay-buffer) of data to be fed to LSTM. This work is expected to make the LSTM more efficient and help agent solve the problem in fewer episodes.

Other opportunities for improvements could be from different algorithms such as Proximal Policy Optimization, Distributed Distributional Policy Gradients etc.