I. Introduction

In Reinforcement learning, agents learn to respond to external stimulus in their environment and can modify this behavior to the same input by analyzing and learning the impact of the outcome, to maximize their reward, which serves as the success criteria.

In this project, we have taken a humanoid model provided in OpenAI Gym and trained it so that it learns to kick a ball into the soccer net, essentially scoring a goal. OpenAI Gym provides scripts and classes for the humanoid model to learn how to stand and walk on its own, and we took this further by training the humanoid to learn to interact with a disconnected object, a simulated soccer ball, and move it towards a goal.

II. Tools

1. Open AI Gym:
   - Gym is a platform developed by OpenAI, and has prebuilt environments which are used to develop and test the efficacy of the reinforcement learning algorithms
   - Every environment will return the same four variables after an action is taken- new state, reward, indicator of environment's state and debugging information

2. MuJoCo:
   - MuJoCo is a physics engine aimed to help researchers create accurate simulations while accounting for joint dynamics and contact forces, leaving the user to only bother about the RL algorithm.
   - Interactive 3d visualisation support is provided with openGL.

III. Algorithm

1. Proximal Policy Optimization:
   - A simplified version of Trust Region Policy Optimization, an iterative procedure that confines the step taken in the optimization procedure within a trust region, given by the KL divergence of the new policy and the old policy.
   - KL divergence term is added to the loss function of policy, and the normal gradient is calculated for the policy.

2. Policy and Value function:
   - The policy function takes into account the present environment, and returns an action based on the observation, and is modelled by a gaussian with a diagonal covariance matrix.
   - The value function will evaluate the policy function to either a positive reward or a negative reward.
   - Both the policy and value functions are both modelled with 3 layer neural networks and tanh serves as the activation function.

IV. Rewards

1. Move ball towards goal:
   - $d$ is the distance between ball & goal. $d$ has to shrink, so derivative is negative.
   - We multiply this rate by a negative weight to make reward positive.

2. Move humanoid towards ball:
   - Distance between the ball and the humanoid has to be decreased.
   - Multiplied with a negative weight to make reward positive.

3. Terminating condition:
   - Angles the center of the ball makes with the corners of the goal ($\Theta_1$ and $\Theta_2$) are calculated.
   - After the simulation, angle of the new position of the ball is calculated and if it's between $\Theta_1$ and $\Theta_2$, we continue, else we terminate.

V. Results

A measurable metric we developed is the average probability of the model making a goal. We tested 2 models: one where we did not reinforce any criteria to keep the model standing and one where we did.

No Standing Reinforcement
- The model would thrust its right leg forward and left leg backwards
- The right leg strikes the ball
- The ball rolls on a trajectory towards the goal and the humanoid model falls to the ground.

<table>
<thead>
<tr>
<th>Training Episodes</th>
<th>27,520</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Success Probability of 20 Episode Batch</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Standing Reinforcement
- The model stiffen its legs and spreads it arms outward.
- It moves its front right leg forward striking the ball.
- The ball moves on a slow trajectory towards the goal.
- The model pumps its arms as it begin to fall over.

<table>
<thead>
<tr>
<th>Training Episodes</th>
<th>97,540</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Success Probability of 20 Episode Batch</td>
<td>0.45</td>
</tr>
</tbody>
</table>

VI. Conclusions

The Proximal Policy Optimizer and our designed reward systems were very successful. In most cases, we were able to induce the desired behavior after enough training. With even more training, our average goal rate would have increased. That being said, training is computationally expensive and consumes vast amounts of time, days in fact. We stopped our trainings where they were because we didn’t have any more time to give them but we are still confident and happy with the results they bared.