Defeating the Invaders with Deep Reinforcement Learning
Christian L. Martinez-Nieves, chris151@Stanford.edu
Department of Computer Science, Stanford University

Motivation
- The purpose is to achieve human-like performance using RL on a game like Space Invaders.
- Two algorithms are implemented: Deep Q-Learning (DQL) and Deep Deterministic Policy Gradients (DDPG).
- After training both RL algorithms on Space Invaders, their performance is compared by testing them on 100 consecutive game episodes.
- Although both algorithms perform well, DDPG manages to get better results with significantly less training.

Data and Features
- The input data to the models are raw grayscale pixel values (game screen) provided by the OpenAI Gym Atari Emulator.
- The input data provided is pre-processed by converting it to grayscale, down sampling, and cropping it to have size 84x84x1.
- Pre-processed input has 7,056 features.
- Pre-processed data allows the model to extract useful information, while also reducing processing necessary for each input.

Models
- Deep-Q Learning Model: a variant of the Q-learning algorithm, which approximates the Q function using a deep neural network. It’s used in conjunction with the Experience Replay technique to tackle the issue of correlated data and changing data distributions [1] [2]. See figure 2 for network structure.
- DDPG Model: Improves on top of DPG and DQL strategies, however, DDPG approximates a stochastic policy directly using an independent function. It also maintains a parameterized actor function (specifies action current policy) and critic function that is learned using the Bellman equation as in Q-learning.

Results
- The DQ Agent was trained for 60 epochs, where each epoch consisted of 45000 parameter updates.
- The DDPG Agent was trained for 20 epochs, where each epoch consisted of 15000 parameter updates.
- Rewards were clipped during training between -1 and 1.

Rewards Achieved By Each RL Algorithm

<table>
<thead>
<tr>
<th>RL Algorithm</th>
<th>Avg. Reward Training</th>
<th>Avg. Reward Test</th>
<th>Top 5 Rewards Test time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDPG</td>
<td>13.7</td>
<td>255.05</td>
<td>650, 605, 590, 570, 485</td>
</tr>
<tr>
<td>DQL</td>
<td>11.9</td>
<td>196.85</td>
<td>900, 595, 575, 565, 525</td>
</tr>
<tr>
<td>Random</td>
<td></td>
<td>144.5</td>
<td>555, 385, 305, 275, 245</td>
</tr>
</tbody>
</table>

Discussion
- Although the DDPG agent was trained less than the DQ agent, it took much longer to train because it’s updating two models (actor/critic) at each train step.
- DDPG agent required less training than DQ agent to achieve greater overall test performance.
- DQ agent managed to get higher top score. This is likely due to fact that it trained more.

Future Work
- Test different neural network structures and see effect on performance.
- Incorporate batch normalization to both networks.

References