Policy Gradient Methods with Pong
Jeffrey Hu, Gregory Soh
Department of Computer Science, Stanford University

Goals
- Apply Deep Reinforcement Learning methods towards solving Pong
- Difficulty lies in understanding image, large state space, and delayed rewards

Methods
- Used AWS EC2 p2.xlarge cluster (NVIDIA Tesla K80 GPU) for processing
- OpenAI Gym with the stochastic environment Pong-v0
- Tried vanilla policy gradient with various models

Models
- Model 1: One 200 neuron hidden layer with ReLU activations
- Model 2: Convolutional layer that feeds into two fully connected layers
- Model 3: Changed discount rate from 0.99 to 0.95

Commonalities
- Adam optimizer with learning rate of 0.001
- Output of two softmax classes

Results
- Discount factor has no noticeable impact on convergence rate (also tried with discount factor of 0.9)
- Convolutional neural network expectedly leads to faster convergence

Graphs in progress:
- Training with 3 actions (up, down and stay), instead of only 2 actions (up and down)

<table>
<thead>
<tr>
<th># of trials until 0</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highest score</td>
<td>5056</td>
<td>4616</td>
<td>5129</td>
</tr>
<tr>
<td>Mean score (last 1k)</td>
<td>2.05</td>
<td>3.13</td>
<td>1.22</td>
</tr>
<tr>
<td>Mean score (overall)</td>
<td>-7.74</td>
<td>-4.52</td>
<td>-7.93</td>
</tr>
</tbody>
</table>

Future
- Implement using Q-learning and Deep Q-networks (DQN). Recent work has shown that DQN can reach human-level performance on a wide variety of Atari games.
- Implement using Asynchronous Advantage Actor-Critic (A3C). Research has shown that exploiting asynchronicity achieves better results than vanilla DQN and do so with fewer computational resources.

References