ABSTRACT

"Learning from Human Preferences": recently released OpenAI paper exploring learning a reward shaping function from sparse human feedback expressed as preferences over alternate state-action trajectories encoded as video clips and presented in a web interface

Objective: reproduce the results of the paper in a new setting - simple ball bouncing games based on the Unity3D ml-agents RL framework - and explore an enhancement where we train a reward shaping predictor for a subgoal, which is then reused as an additive reward shaping bias to train a new reward shaping predictor for the end goal.

MODELS

RL Algorithm - OpenAI Proximal Policy Optimization: variant of policy gradient, implemented as 2 x 64 node hidden layer MLP:

\[ L(\theta) = \mathbb{E}_r \sum_{s \in D} \log p(a_t | s; \theta) \]

Reward predictor - predicts latent reward \( r \) from probabilities \( P \) of user preference over alternate trajectories (o1, o2):

\[ \hat{P}[\sigma^1 > \sigma^2] = \frac{\exp \sum \hat{r}(o^1, a_t) + \exp \sum \hat{r}(o^2, a_t)}{\exp \sum \hat{r}(o^1, a_t) + \exp \sum \hat{r}(o^2, a_t)} \]

2 x 64 node MLP with softmax loss over binary preferences \( \mu_1 \) and \( \mu_2 \):

\[ \text{loss} = - \sum_{(s, a, \mu) \in D} \mu(1) \log \hat{P}[\sigma^1 > \sigma^2] + \mu(2) \log \hat{P}[\sigma^2 > \sigma^1] \]

Reward bias: a pre-learned reward function’s predictions can be used as additive bias when learning a new reward function:

\[ r_{\text{bias}}(o_t, a_t) = \hat{r}(o_t, a_t) \]

BASELINE

Bounce a ball to target with:

- RL = hard-coded reward
- SYNTH = preferences from hard-coded reward
- HUMAN = preferences

Similar performance to coded reward achieved with as few as 100 human comparison labels.

MAIN EXPERIMENT

Bounce ball between platforms

- RL = hard-coded reward: +0.1 if any platform hit, -1 if ball lost
- HUMAN1 = train right platform using human feedback to bounce ball towards left platform. Model learns correct behaviour for right platform, while the correct behaviour is preserved for the left platform. Success!
- HUMAN2 = reuse the first reward predictor with weights fixed as an additive shaping bias for the reward of a new predictor, which can be trained to drive the correct behaviour for left platform, while the correct behaviour is preserved for the right platform. Success!
- HUMAN2 (+ HUMAN1 bias) = training new reward function from human feedback, accumulated some true reward

RESULT ANALYSIS

<table>
<thead>
<tr>
<th>REWARD FUNCTION</th>
<th>MAX TRUE REWARD</th>
<th>TIMES TO MAX REWARD</th>
<th>TOTAL HUMAN DECISIVE COMPARISONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL</td>
<td>-1.383</td>
<td>31000</td>
<td>n/a</td>
</tr>
<tr>
<td>HUMAN1</td>
<td>-1.062</td>
<td>119000</td>
<td>72</td>
</tr>
<tr>
<td>HUMAN2 (+ HUMAN1 bias)</td>
<td>-1.929</td>
<td>226000</td>
<td>63</td>
</tr>
</tbody>
</table>

CONCLUSION

Using human feedback proves to be a practical approach to training RL systems in novel ways. The success of our relatively crude approach of additive reward shaping suggests the possibility of a future where combining fairly standardized learning models in unsophisticated, albeit creative ways, could yield meaningful results. Given more time, we’d have liked to study and approach reward shaping more rigorously, and apply this on more complex environments, e.g. OpenAI Roboschool.

References:

https://blog.openai.com/deep-reinforcement-learning-from-human-preferences/
https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Example-Environments.md
https://github.com/nottombrown/rl-teacher/
https://blog.openai.com/deep-reinforcement-learning-from-human-preferences/
Ng, A.Y., Harada, D., Russell, S.J.: Policy invariance under reward transformations.