Policy Gradient on cartpole

Farnaz Adib Yaghmaie

Linkoping University, Sweden
farnaz.adib.yaghmaie@liu.se

March 12, 2021
Policy Gradient on cartpole

A harbor

The cartpole

Photo credit: @http://rhm.rainbowco.com.cn/

Photo credit: @https://gym.openai.com/
- **States:** 1. position of the cart on the track, 2. angle of the pole with the vertical, 3. cart velocity, and 4. rate of change of the angle.

- **Actions:** +1, -1

- **Reward:**

\[ r_t = \begin{cases} 
1, & \text{if the pendulum is upright} \\
0, & \text{otherwise} 
\end{cases} \]
**Episode ends when:**

- The pole is more than 15 degrees from vertical or
- The cart moves more than 2.4 units from the center or
- The episode lasts for 200 steps.

**Solvability Criterion:** Getting average sum reward of 195.0 over 100 consecutive trials.
We build a deep network to represent the pdf $\pi_\theta = network(s)$

```python
network = keras.Sequential([keras.layers.Dense(30, input_dim=n_s, activation='relu'),
                           keras.layers.Dense(30, activation='relu'),
                           keras.layers.Dense(n_a, activation='softmax')])
```

and assign a cross entropy cost function for it

```python
network.compile(loss='categorical_crossentropy')
```
1. Collect data
   - Observe $s$ and sample $a \sim \pi_\theta(s)$
     
     $\text{softmax\_out} = \text{network}(\text{state})$
     
     $a = \text{np.random.choice}(n_a, p=\text{softmax\_out}.\text{numpy()}[0])$

     - Apply $a$ and observe $r$.
     - Add $s, a, r$ to the history.

2. Update the parameter $\theta$
   - We calculate the reward to go and standardize it.
   - We optimize the policy

     $\text{target\_actions} = \text{tf.keras.utils.to_categorical}(\text{np.array}(\text{actions}), n_a)$
     
     $\text{loss} = \text{self.network}\_.\text{train\_on\_batch}(\text{states}, \text{target\_actions},$
     
     $\text{sample\_weight=rewards\_to\_go})$
Try the following:

- Run Crash_course_on_RL/pg_on_cartpole_notebook.ipynb and verify to get the solution after \( \sim 1000 \) episodes.

- Change \( 0 \leq \gamma \leq 1 \) to see if you can solve the problem faster '

  'GAMMA': 0.9 in agent_par

- Make sure you understand the code!
How the reward looks like during learning

Figure: Total reward vs. no. of episodes
Email your questions to

farnaz.adib.yaghmaie@liu.se