Q-learning on cartpole

Farnaz Adib Yaghmaie

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

April 6, 2021
A harbor crane

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

The cartpole

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 take the state and generate $Q$ for all actions

$$Q(s, a) = \text{network}(state)$$

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

and assign a mean squared error cost function for it

```python
self.network.compile(loss='mean_squared_error',
                     optimizer=keras.optimizers.Adam())
```

The policy $\pi$ is the index which the output of the network is maximized.

```python
policy = np.argmax(network(state))
```
1. Collect data
   - Observe $s$ and select $a$
     
     $a = \begin{cases} 
     \text{random action} & \text{if } r < \epsilon, \\
     \arg \max_a Q(s, a) & \text{Otherwise.}
     \end{cases}$

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

2. Update the parameter $\theta$.
   - Define $Q_{\text{target}}(r_t, s_{t+1}) = r_t + \gamma \arg \max_a Q(s_{t+1}, a)$
   - Minimize the mean squared error
     
     $loss = \text{self.network.train_on_batch(states, q_target)}$
Try the following:

- Run
  
  Crash_course_on_RL/q_on_cartpole_notebook.ipynb
  and verify to get the solution after $\sim 2885$ episodes.

- Set
  
  `'epsilon': 0.0` in `agent_par`
  and verify that the agent cannot solve the problem!

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

Figure: Total reward vs. no. of episodes
Replay Q learning

2885 episodes?? quite bad!

Replay Q can improve it:

- Build a memory and save data sequentially. When the memory is full, disregard the oldest data and add the new data

- Sample the memory instead of using the latest episode
Try the following:

- Run `replay_q_on_cartpole_notebook.ipynb` and verify to get the solution after \( \sim 475 \) episodes.

- 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