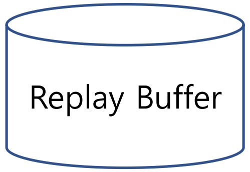
최적화 학습 알고리즘
코드 수정

2023/06/01

현재 알고리즘

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R



2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$ using target $y_i = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i') - \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i, \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$
   1. $J_t(\theta) = \frac{1}{N}\sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i|\boldsymbol{s}_i)}{\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \dots, K\ do\ \ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

현재 알고리즘

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R

/run.py#L418~457에 해당

2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$ using target $y_i = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i') - \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i, \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$
   1. $J_t(\theta) = \frac{1}{N}\sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i|\boldsymbol{s}_i)}{\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \ldots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/run.py#L418

현재 알고리즘

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log\pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$ using target $y_i = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i') - \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i, \log\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$
   1. $J_t(\theta) = \frac{1}{N}\sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i|\boldsymbol{s}_i)}{\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \dots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

ppoTF2.replayNew(): /ppoTF2.py#L765~805에 해당

run.py#L481→ppoTF2.py#L690 → ppoTF2.py#L765 순으로 호출.

https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/run.py#L481
https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/policy/ppoTF2.py#L690
https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/policy/ppoTF2.py#L765

현재 알고리즘

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$ using target $y_i = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i') - \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i, \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$
   1. $J_t(\theta) = \frac{1}{N} \sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i|\boldsymbol{s}_i)}{\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \ldots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

Target value 계산: /ppoTF2.py#L790 and #795

Training: /ppoTF2.py# 804

https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/policy/ppoTF2.py#L790
https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/policy/ppoTF2.py#L795
https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/policy/ppoTF2.py#L804

현재 알고리즘

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log\pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$ using target $y_i = r_i + \gamma\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = r_i + \gamma\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i') - \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i, \log\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$
   1. $J_t(\theta) = \frac{1}{N}\sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i|\boldsymbol{s}_i)}{\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)}\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \ldots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha\nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

Critic DNN Model: /ppoTF2.py#249

Critic Loss: /ppoTF2.py#282

https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/policy/ppoTF2.py#L249
https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/policy/ppoTF2.py#L282

현재 알고리즘

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$ using target $y_i = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i') - \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$

Advantage 계산: /ppoTF2.py#795

5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i, \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$
   1. $J_t(\theta) = \frac{1}{N} \sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i|\boldsymbol{s}_i)}{\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \dots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/policy/ppoTF2.py#L795

현재 알고리즘

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log\pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$ using target $y_i = r_i + \gamma\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = r_i + \gamma\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i') - \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i, \log\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$
   1. $J_t(\theta) = \frac{1}{N}\sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i|\boldsymbol{s}_i)}{\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)}\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \dots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha\nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

Training: /ppoTF2.py#803

https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/policy/ppoTF2.py#L803

현재 알고리즘

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$ using target $y_i = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = r_i + \gamma \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i') - \hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i, \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$
   1. $J_t(\theta) = \frac{1}{N} \sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i|\boldsymbol{s}_i)}{\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \dots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

Actor DNN Model: /ppoTF2.py#180

PPO Loss $J_t(\theta)$: /ppoTF2.py#217

https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/policy/ppoTF2.py#L180
https://github.com/etri-city-traffic-brain/traffic-signal-optimization/blob/8ac45baeedda25f78d5343223ac2edee7aa44ee8/atsc-rl/multiagent_tf2/policy/ppoTF2.py#L217

수정 알고리즘(Off-Policy)

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i)$ using target $y_i = r_i + \gamma \frac{1}{M} \sum_{\boldsymbol{a}_t'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i')$; $\boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \frac{1}{M} \sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t)$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i^t, \log \pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)\}$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
    1. $J_t(\theta) = \frac{1}{N} \sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i^t|\boldsymbol{s}_i)}{\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t), \sim\right)$ and $\theta^1 = \theta_t$
    2. $for\ k = 1, \ldots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
    3. $\theta_{t+1} = \theta^K$
6. Repeat.

수정 알고리즘(Off-Policy)

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R

이전 코드와 동일. /run.py#L418~457에 해당

2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i)$ using target $y_i = r_i + \gamma \frac{1}{M} \sum_{\boldsymbol{a}_t'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i')$; $\boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \frac{1}{M} \sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t)$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i^t, \log \pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)\}$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
   1. $J_t(\theta) = \frac{1}{N} \sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i^t|\boldsymbol{s}_i)}{\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \ldots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

수정 알고리즘(Off-Policy)

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R

이전 코드와 동일. /ppo.py#L1032~1041에 해당

3. Update $\hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i)$ using target $y_i = r_i + \gamma \frac{1}{M} \sum_{\boldsymbol{a}_t'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i')$; $\boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \frac{1}{M} \sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t)$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i^t, \log \pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)\}$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
   1. $J_t(\theta) = \frac{1}{N} \sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i^t|\boldsymbol{s}_i)}{\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \ldots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

수정 알고리즘(Off-Policy)

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\boxed{\hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i)}$ using target $y_i = r_i + \gamma \frac{1}{M} \sum_{\boldsymbol{a}_t'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i');\ \boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i')$ Critic DNN Model: /ppo.py#L328
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \frac{1}{M} \sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t);\ \boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i^t, \log \pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)\};\ \boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
   1. $J_t(\theta) = \frac{1}{N} \sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i^t|\boldsymbol{s}_i)}{\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \ldots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

수정 알고리즘(Off-Policy)

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i)$ using target $y_i = r_i + \gamma \boxed{\frac{1}{M}\sum_{\boldsymbol{a}_t'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i');\ \boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i')}$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \frac{1}{M}\sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t)$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i^t, \log \pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)\}$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
   1. $J_t(\theta) = \frac{1}{N}\sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i^t|\boldsymbol{s}_i)}{\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \ldots, K\ do\ \ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

Value for next state $\boldsymbol{s}_i'$

$$\boxed{\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i') \approx \frac{1}{M}\sum_{\boldsymbol{a}_i'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i')}$$

$$\uparrow$$

$$\boxed{\boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i')}$$

/ppo.py#L1060, #959

수정 알고리즘(Off-Policy)

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i)$ using $\boxed{\text{target } y_i = r_i + \gamma \frac{1}{M} \sum_{\boldsymbol{a}_t'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i');\ \boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i')}$ Compute target values. /ppo.py#1066
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \frac{1}{M} \sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t)\,;\ \boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i^t, \log \pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)\}\,;\ \boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
    1. $J_t(\theta) = \frac{1}{N} \sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i^t|\boldsymbol{s}_i)}{\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t), \sim\right)$ and $\theta^1 = \theta_t$
    2. $for\ k = 1, \dots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
    3. $\theta_{t+1} = \theta^K$
6. Repeat.

수정 알고리즘(Off-Policy)

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i)$ using target $y_i = r_i + \gamma \frac{1}{M} \sum_{\boldsymbol{a}_t'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i')$; $\boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i')$

Training: /ppo.py#1072

4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \frac{1}{M} \sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}\left(\boldsymbol{s}_i, \boldsymbol{a}_i^t\right)$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
5. Minibatch Learning on $\left\{\boldsymbol{s}_i, \boldsymbol{a}_i^t, \log \pi_{\theta_t}\left(\boldsymbol{a}_i^t|\boldsymbol{s}_i\right)\right\}$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
   1. $J_t(\theta) = \frac{1}{N} \sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i^t|\boldsymbol{s}_i)}{\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}\left(\boldsymbol{s}_i, \boldsymbol{a}_i^t\right), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \dots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t\left(\theta^k\right)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

수정 알고리즘(Off-Policy)

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log \pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i)$ using target $y_i = r_i + \gamma \frac{1}{M} \sum_{\boldsymbol{a}_t'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i')$; $\boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i')$
4. $\boxed{\text{Evaluate } \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \frac{1}{M} \sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t)\,;\ \boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)}$

/ppo.py#978

5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i^t, \log \pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)\}$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
   1. $J_t(\theta) = \frac{1}{N} \sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i^t|\boldsymbol{s}_i)}{\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \ldots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

$$\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \boxed{\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i)}$$

$$\hat{V}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i) \approx \frac{1}{M} \sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t) \leftarrow \boxed{\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)}$$

Sample actions according to the current policy.

수정 알고리즘(Off-Policy)

1. Take action $\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log\pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i)$ using target $y_i = r_i + \gamma \frac{1}{M}\sum_{\boldsymbol{a}_t'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i')$; $\boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot|\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \frac{1}{M}\sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t)$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot|\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i^t, \log\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)\}$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot|\boldsymbol{s}_i)$
   1. $J_t(\theta) = \frac{1}{N}\sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i^t|\boldsymbol{s}_i)}{\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \ldots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha\nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

Actor DNN Model: /ppoTF2.py#172

PPO Loss $J_t(\theta)$: /ppoTF2.py#233

기존과 동일

수정 알고리즘(Off-Policy)

Actually, we do not need this one.

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \boxed{\log \pi(\boldsymbol{a}|\boldsymbol{s})})$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log \pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i)$ using target $y_i = r_i + \gamma \frac{1}{M} \sum_{\boldsymbol{a}_t'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i')$; $\boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \frac{1}{M} \sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t)$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$

/ppo.py#1077 and #995

5. Minibatch Learning on $\boxed{\{\boldsymbol{s}_i, \boldsymbol{a}_i^t, \log \pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)\}\ ;\ \boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)}$
   1. $J_t(\theta) = \frac{1}{N} \sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i^t|\boldsymbol{s}_i)}{\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t), \sim\right)$ and $\theta^1 = \theta_t$
   2. $for\ k = 1, \ldots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha \nabla_{\theta^k} J_t(\theta^k)$
   3. $\theta_{t+1} = \theta^K$
6. Repeat.

Augment a batch by sampling actions according to the current policy.

수정 알고리즘(Off-Policy)

1. Take action $\boldsymbol{a} \sim \pi(\cdot\,|\boldsymbol{s})$, get $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', \log\pi(\boldsymbol{a}|\boldsymbol{s}))$ and store in R
2. Sample a batch $\{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_i', \log\pi(\boldsymbol{a}_i|\boldsymbol{s}_i)\}$ from buffer R
3. Update $\hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i)$ using target $y_i = r_i + \gamma \frac{1}{M}\sum_{\boldsymbol{a}_t'} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i', \boldsymbol{a}_i')$; $\boldsymbol{a}_i' \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i')$
4. Evaluate $\hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) = \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i) - \frac{1}{M}\sum_{\boldsymbol{a}_i^t} \hat{Q}_\phi^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t)$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
5. Minibatch Learning on $\{\boldsymbol{s}_i, \boldsymbol{a}_i^t, \log\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)\}$; $\boldsymbol{a}_i^t \sim \pi_{\theta_t}(\cdot\,|\boldsymbol{s}_i)$
    1. $J_t(\theta) = \frac{1}{N}\sum_i \min\left(\frac{\pi_\theta(\boldsymbol{a}_i^t|\boldsymbol{s}_i)}{\pi_{\theta_t}(\boldsymbol{a}_i^t|\boldsymbol{s}_i)} \hat{A}^{\pi_{\theta_t}}(\boldsymbol{s}_i, \boldsymbol{a}_i^t), \sim\right)$ and $\theta^1 = \theta_t$
    2. $for\ k = 1, \ldots, K\ do\ \theta^{k+1} \leftarrow \theta^k + \alpha\nabla_{\theta^k} J_t(\theta^k)$
    3. $\theta_{t+1} = \theta^K$
6. Repeat.

Training: /ppo.py#1089

Q & A