Remember and Forget for Experience Replay

Guido Novati & Petros Koumoutsakos
Computational Science, ETH Zürich
Off-policy Reinforcement Learning

- Off-policy RL with Experience Replay typically alternates:

  ![Diagram]

  - **Learner**
    - Train $\pi^\mu(a | s)$ and/or $Q^\mu(s, a)$ with Replay Memory

  - **Agent**
    - Explore environment $a_t \sim \mu_t(a | s_t)$

**Behaviors:** $\mu_t(a | s)$

**Experiences:** $\{s_t, r_t, \mu_t, a_t\}$
Off-policy Reinforcement Learning

- Off-policy RL with Experience Replay typically alternates:
  - **Learner**: Train $\pi^w(a \mid s)$ and/or $Q^w(s, a)$ with Replay Memory
  - **Agent**: Explore environment $a_t \sim \mu_t(a \mid s_t)$

- Replay behaviors are typically associated with past policy iterations.
- Off-policy RL attempts to estimate on-policy quantities from off-policy data.

\[
E.g. \text{ maximize on-policy returns: } J(w) = \mathbb{E}_{t \sim \text{RM}} \left[ \frac{\pi^w(a_t \mid s_t)}{\mu_t(a_t \mid s_t)} Q^w(s_t, a_t) \right]
\]
1) Which learns a **parameterized policy**.

*E.g.* DDPG (Lillicrap *et al.* 2016) trains deterministic policy $m(s)$ and adds exploration noise:

$$
\pi^w(a \mid s) = m^w(s) + \mathcal{N}(0, \sigma^2)
$$

2) With **off-policy gradients estimated by ER**.

$$
g(w) = \mathbb{E}_{t \sim \text{RM}} \left[ \hat{g}(t, \bar{w}) \right]
$$

*E.g.* deterministic policy gradient (Silver *et al.* 2014):

$$
\hat{g}^{\text{DPG}}(t, \bar{w}) = \nabla_w m^w(s_t) \nabla_a Q^w(s_t, a) \bigg|_{a=m^w(s_t)}
$$
### RL algorithm

1) Which learns a **parameterized policy**. 

*E.g.* DDPG (Lillicrap et al. 2016) trains deterministic policy $m(s)$ and adds exploration noise: 

$$ \pi^w(a \mid s) = m^w(s) + \mathcal{N}(0, \sigma^2) $$

2) With **off-policy gradients estimated by ER**.

\[
g(w) = \mathbb{E}_{t \sim \text{RM}} \left[ \hat{g}(t, \bar{w}) \right]
\]

*E.g.* deterministic policy gradient (Silver et al. 2014):

\[
\hat{g}^{\text{DPG}}(t, \bar{w}) = \nabla_w m^w(s_t) \nabla_a Q^w(s_t, a) \bigg|_{a=m^w(s_t)}
\]

### ReF-ER

1) **Rejects samples** from gradient estimation if importance weight $\rho_t^w = \pi^w(a_t \mid s_t) / \mu_t(a_t \mid s_t)$ outside of a trust region.

2) **Penalizes policy towards training behaviors**.

\[
\hat{g}(t, \bar{w}) \leftarrow \begin{cases} 
\beta \hat{g}(t, \bar{w}) - (1 - \beta) \nabla D_{\mathcal{KL}} \left[ \mu_t \parallel \pi^w(\cdot \mid s_t) \right] & \text{if } \frac{1}{C} < \rho_t < C \\
-(1 - \beta) \nabla D_{\mathcal{KL}} \left[ \mu_t \parallel \pi^w(\cdot \mid s_t) \right] & \text{otherwise}
\end{cases}
\]

**Notes:**
- Trust region parameter $C$ can be annealed.
- Coefficient $\beta$ is iteratively updated to keep a fixed fraction of samples within the trust region.
Results

- We observe: effectively constrained $D_{KL}$, increased stability and performance.
- At the price of: sometimes slower progress at the beginning of training.

**DDPG on OpenAI gym MuJoCo tasks & flow control**

---

**Graphs**: Cumulative reward and $D_{KL}$ over time for different algorithms. The graphs illustrate the performance of DDPG with and without Ref-ER, as well as DDPG with experience replay and prioritized experience replay. The plots show that ReF-ER improves the stability and performance of DDPG, although at the cost of sometimes slower progress at the beginning of training.
Conclusion

GENERAL IMPLICATION:
Off-policy RL benefits from maintaining similarity between policy and training behaviors.

ReF-ER:  
• Easy to implement, modular improvement for off-policy RL.
• Aligns on-policy distribution (‘test set’) and replay experiences (‘training set’).
• Brings off-policy RL one step closer to supervised learning.

More info:
• poster : Pacific Ballroom # 50
• paper : https://arxiv.org/abs/1807.05827
• source code : https://github.com/cselab/smarties