Off-Policy Deep Reinforcement Learning without Exploration

Scott Fujimoto, David Meger, Doina Precup
Mila, McGill University
Surprise!

Agent orange and agent blue are trained with...

1. The **same off-policy algorithm** (DDPG).

2. The **same dataset**.
The Difference?

1. **Agent orange:** Interacted with the environment.
   - Standard RL loop.
   - Collect data, store data in buffer, train, repeat.

2. **Agent blue:** Never interacted with the environment.
   - Trained with data collected by agent orange concurrently.
1. Trained with the same off-policy algorithm.
2. Trained with the same dataset.
3. One interacts with the environment. One doesn’t.
Off-policy deep RL fails when truly off-policy.
Value Predictions

- **HalfCheetah-v1**: The graph shows the estimated value over time steps, increasing gradually with time.
- **Hopper-v1**: The value predictions fluctuate significantly over time steps, indicating a complex behavior.
- **Walker2d-v1**: The value predictions are less volatile, showing a smoother trend over time steps.
Extrapolation Error

\[ Q(s, a) \leftarrow r + \gamma Q(s', a') \]
Extrapolation Error

\[ Q(s, a) \leftarrow r + \gamma Q(s', a') \]
Extrapolation Error

\[ Q(s, a) \leftarrow r + \gamma Q(s', a') \]

1. \((s, a, r, s') \sim Dataset\)
2. \(a' \sim \pi(s')\)
Extrapolation Error

\[ Q(s, a) \leftarrow r + \gamma Q(s', a') \]

\((s', a') \notin Dataset\) \rightarrow Q(s', a') = \text{bad}

\rightarrow Q(s, a) = \text{bad}
Extrapolation Error

\[ Q(s, a) \leftarrow r + \gamma Q(s', a') \]

\((s', a') \notin Dataset \rightarrow Q(s', a') = \text{bad} \]

\rightarrow Q(s, a) = \text{bad} \]
Extrapolation Error

\[ Q(s, a) \leftarrow r + \gamma Q(s', a') \]

\[(s', a') \notin \text{Dataset} \rightarrow Q(s', a') = \text{bad} \]
\[ \rightarrow Q(s, a) = \text{bad} \]
Extrapolation Error

Attempting to evaluate $\pi$ without (sufficient) access to the $(s,a)$ pairs $\pi$ visits.
Batch-Constrained Reinforcement Learning

Only choose $\pi$ such that we have access to the $(s, a)$ pairs $\pi$ visits.
Batch-Constrained Reinforcement Learning

1. $a \sim \pi(s)$ such that $(s, a) \in Dataset$.
2. $a \sim \pi(s)$ such that $(s', \pi(s')) \in Dataset$.
3. $a \sim \pi(s)$ such that $Q(s, a)$ is maxed.
Batch-Constrained Deep Q-Learning (BCQ)

First imitate dataset via generative model:
\[ G(a|s) \approx P_{\text{Dataset}}(a|s). \]

\[ \pi(s) = \arg\max_{a_i} Q(s, a_i), \text{ where } a_i \sim G \]
(l.e. select the best action that is likely under the dataset)

(+ some additional deep RL magic)
Come say Hi @ Pacific Ballroom #38 (6:30 Tonight)

https://github.com/sfujim/BCQ

(Artist’s rendition of poster session)