Off-Policy Deep Reinforcement Learning without Exploration

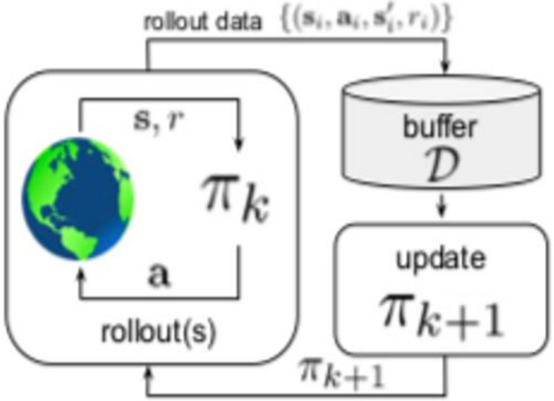
Presenter: Yian Wong

Off-Policy RL

- Evaluate and update one policy while following another
- Policies may not necessarily be similar

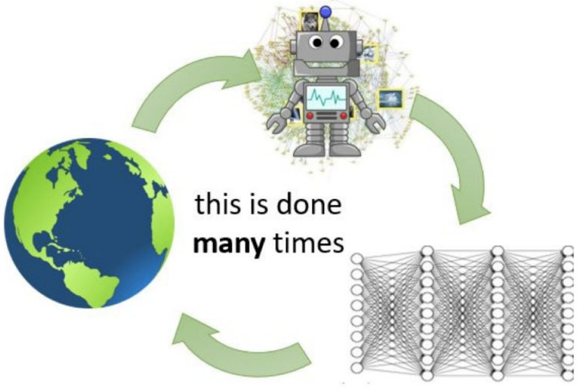
- **Q-Learning** or **DDPG** are classic examples of Off-Policy RL

- **Online RL**
  - Sample efficiency
  - Implementation
  - Stability

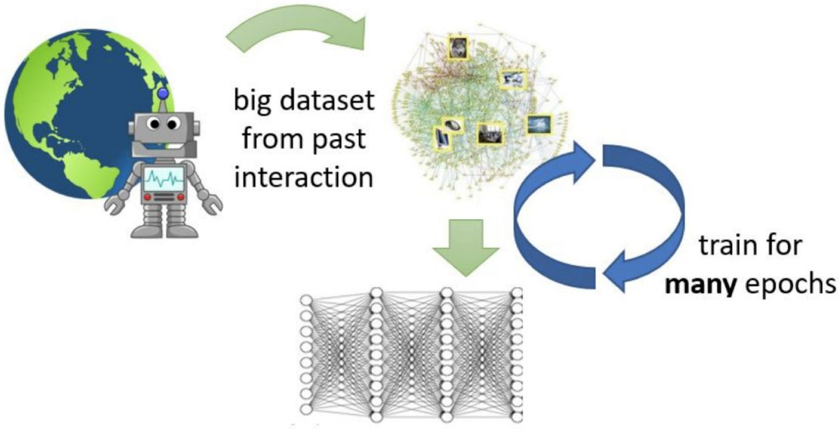


Batch RL

- "Growing batch RL"
  - Algorithm is learning from earlier trajectories that it collected
- In Batch RL, the data could be completely uncorrelated with the current policy
  - High **extrapolation error** between the dataset policy and the current policy



Regular RL



Batch RL

Related Works in Deep RL

- Many SOTA RL algorithms are off-policy:
  - DDPG
  - DQN
  - IMPALA

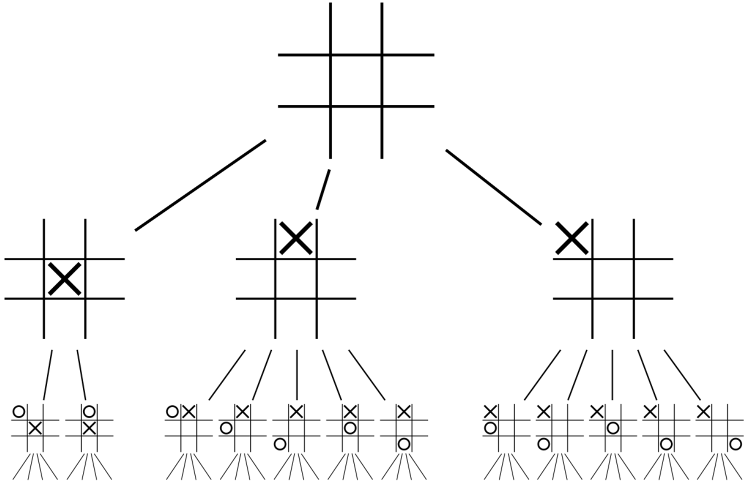
- Imitation Learning

- Catastrophically fail when exposed to the ‘Batch RL’ problem
  - High ‘extrapolation error’ between the current and behavioral policy

Algorithm

- Authors suggest high "extrapolation error" in existing approaches:
    - Visitation of state, action pairs that aren't similar to the ones found in the dataset
        - Poor Q estimates

- The algorithm restricts the target policy to be similar to the dataset behavioral policy



Batch Constrained Q-Learning (BCQ)

- Based on the DQN algorithm
- BCQ uses a generative model to generate highly plausible/similar actions to the dataset
  - Use a conditional VAE which encodes the state and generates actions
  - Perturb the selected actions of the VAE using
- This gives us the following for the policy:

$$\pi(s) = \underset{a_i+\xi_\phi(s,a_i,\Phi)}{\operatorname{argmax}} \; Q_\theta\Big(s, a_i + \xi_\phi(s, a_i, \Phi)\Big) \qquad \{a_i \sim G_\omega(s)\}_{i=1}^n$$

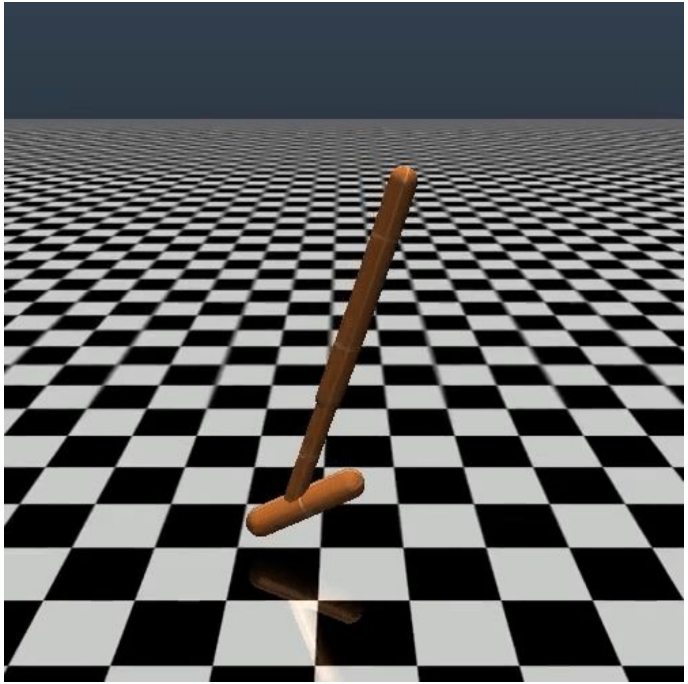
Q-value

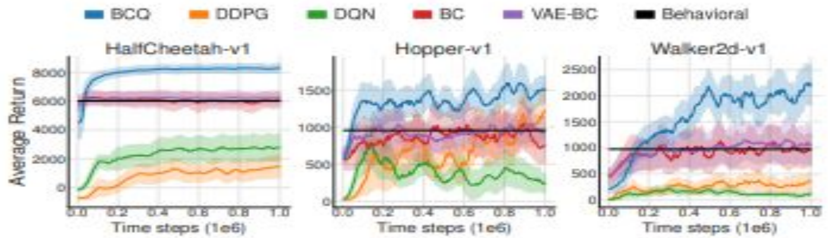
Random perturbation

Actions sampled from conditional-VAE

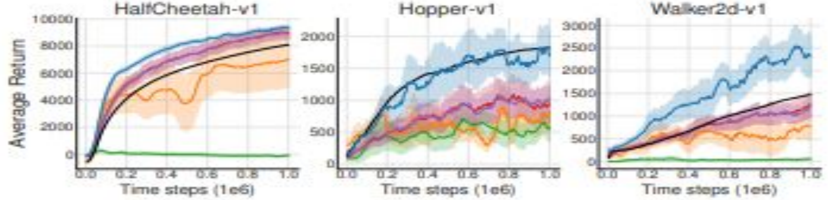
Experimental Setup

- Analyze results from OpenAI Gym MuJoCo's HalfCheetah, Hopper, and Walker2d environments
- Test on 4 kinds of Batch RL:
  - Final buffer
  - Concurrent
  - Imitation Learning
  - Imperfect demonstrations



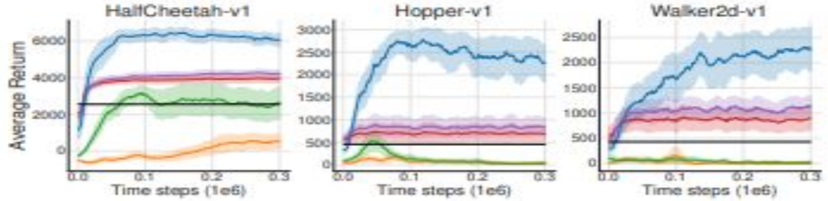


(a) Final buffer performance



(b) Concurrent performance

(c) Imitation performance



(d) Imperfect demonstrations performance

Discussion of Results

- BCQ outperforms on experiments that are outside of the conventional ‘growing batch-RL’ setup.

- Situations where the dataset can differ greatly from the current policy results in failure for DDPG and DQN

- BCQ can perform similarly to imitation learning algorithms as well as off-policy RL algorithms

- BCQ outperforms DDPG or DQN when learning from data generated by DDPG or DQN

Open Issues

- Bound by the performance of the behavioral policy of the dataset
- Doesn’t address the problem of data generated with bad policies (such as random actors)
    - Lack of exploration leads to just cloning the behavioral policy, without exceeding its performance
- Value based
    - Bad/random values learned for state-actions with poor visitation
    - Difficult to learn for

Future Work for Paper / Reading

- Model-based approaches
  - Using the dataset to **learn dynamics** of the MDP (ie transition function)
  - Capture **uncertainties** of learned model using probabilistic modeling
  - Maximize expected return using a model-free algorithm (DQN, PPO) in the learned dynamics system
- Inverse RL
  - Learn the reward function R(s, a) from the data. Pick actions that maximize the learned function.

Extended Readings

- “Extrapolating Beyond Suboptimal Demonstrations via Inverse Reinforcement Learning from Observations”
  - Use a ‘ranking’ of demonstrations to learn a reward functions
  - Achieve performance greater than the demonstrations
- “Scaling Data-driven Robotics with Reward Sketching and Batch Reinforcement Learning”
  - Use of ‘reward sketching’, which takes a subset of the dataset and uses human input to ‘sketch’ and idea of what the reward for those states are
  - Use BCQ with these sketched rewards to achieve better performnce

Summary

- Introduced the problem of Batch RL, learning a policy from a dataset of trajectories
- Prior work only focuses on ‘offline RL’, which learns from trajectories produced by earlier iterations of the model.
    - DDPG and DQN perform badly when training on data that is very different from the policy
- BCQ uses a VAE to produce actions similar to the dataset behavioral policy, constraining the agent
- BCQ outperforms DDPG, DQN at all baseline tasks, while performing better than BC in adversarial task for imitation learning.