Offline Q-Learning on Diverse Multi-Task Data Both Scales and Generalizes

Aviral Kumar, Rishabh Agarwal, Young Geng, George Tucker*, Sergey Levine*
Offline Reinforcement Learning

Offline RL

big dataset from past interactions

train for many epochs

deploy learned policy in new scenarios
The Generalization Promise of Offline RL

Training on large, pre-collected datasets to attain broad generalization

Large datasets

Expressive function approximators

Broadly generalizing policies
This Talk

What does it take to make offline RL scale and generalize?
Where Are We At?

Disclaimer: This **definitely** misses some works (sorry!) but reflects the general trend.

**Supervised learning**

- InstDisc
- SimCLR
- SimCLR (2x)
- SimCLR (4x) (red star)
- CPCv2-L
- CPCv2-L (blue star)
- PIRL
- PIRL-ens
- MoCo
- MoCo (2x)
- MoCo (4x)
- CMC
- BigBiGAN
- Rotation

**Reinforcement Learning**

- Gym / D4RL (3/4-layer feed-forward)  
  ~1M
- Mu Zero Reanalyze (16 res. blocks)  
  ~25M
- IMPALA architecture  
  ~30M
- AlphaGo Zero (20 res. blocks)  
  ~32M
- AlphaStar (~55M)  
  ~55M

Generally, much smaller models compared to supervised learning

Picture taken from the SimCLR paper
Large-Scale Study: Single Policy to Play Atari Games

Train a single policy on 40 Atari games

Evaluate on training games

Fine-tune to new games

Why Atari? Why is this problem challenging?

First large-scale test-bed to evaluate generalization and scaling
Requires large networks; offline Q-learning never worked
2 billion transitions, 40 games, sub-optimal data

Three key ingredients in our recipe.....
**Ingredient 1: An offline RL Algorithm**

Conservative Q-Learning (CQL)

\[
\min_{\theta} \alpha \left( \mathbb{E}_{s \sim D} \left[ \log \left( \sum_{a'} \exp(Q_\theta(s, a')) \right) \right] - \mathbb{E}_{s, a \sim D} [Q_\theta(s, a)] \right) + TDError(\theta; D)
\]

Encourages the Q-function to not over-estimate

Ingredient 2: Large Networks

- Not the typical IMPALA architecture!
- Multi-headed Q-function
- Keep track of spatial information in the image!
Ingredient 3: Methods to Effectively Use Capacity

DR3 normalization: Normalize features to have norm = 1

\[ \text{Normalize}(\phi(s, a)) = \frac{\phi(s, a)}{||\phi(s, a)||_2} \]

C51 (Q-value = categorical distribution)

Can train via cross-entropy loss

K., Agarwal, Ma, Courville, Tucker, Levine. **DR3: Value-Based Deep RL Requires Explicit Regularization.** ICLR 2022
Summary: “Scaled Q-Learning”

An offline RL algorithm

Conservative Q-learning (CQL)

A large neural network

A way to effectively use capacity

---

**DR3 normalization:** Normalize features to have norm = 1

\[
\text{Normalize}(\phi(s, a)) = \frac{\phi(s, a)}{\|\phi(s, a)\|_2}
\]
Performance of Scaled Q-Learning

**We find:** ~80M ResNet + sub-optimal data => better than online RL or decision transformers
Scaling Trends for Scaled Q-Learning

Scaled Q-learning does scale favorably, while other naïve choices (IMPALA) do not
Limited offline data for a new game + pre-trained model on the training games

82% improvement
Generalization: Online Fine-Tuning to New Modes

Scaled Q-Learning learns representations useful under changes to the environment.

- Freeway (m1d0)
- Hero (m1d0)
- Breakout (m12d0)
DR3 Enables Effective Use of Capacity

<table>
<thead>
<tr>
<th></th>
<th>Scaled QL (ResNet 34)</th>
<th>Scaled QL (ResNet 50)</th>
<th>Scaled QL (ResNet 101)</th>
</tr>
</thead>
<tbody>
<tr>
<td>without feature normalization</td>
<td>50.9%</td>
<td>73.9%</td>
<td>80.4%</td>
</tr>
<tr>
<td>with feature normalization</td>
<td>78.0% (+28.9%)</td>
<td>83.5% (+9.6%)</td>
<td>98.0% (+17.6%)</td>
</tr>
</tbody>
</table>

DR3 improves consistently along the way

Enables the use of higher capacity more effectively
Summary and Takeaways

- We present a simple way to scale Q-learning to large datasets + large models
- Models pre-trained via offline Q-learning learn **generalizing** representations
- Effectively leveraging capacity of large networks seem critical!

- **Preliminary Code:**
  [https://tinyurl.com/scaled-ql-code](https://tinyurl.com/scaled-ql-code)

Thank You!