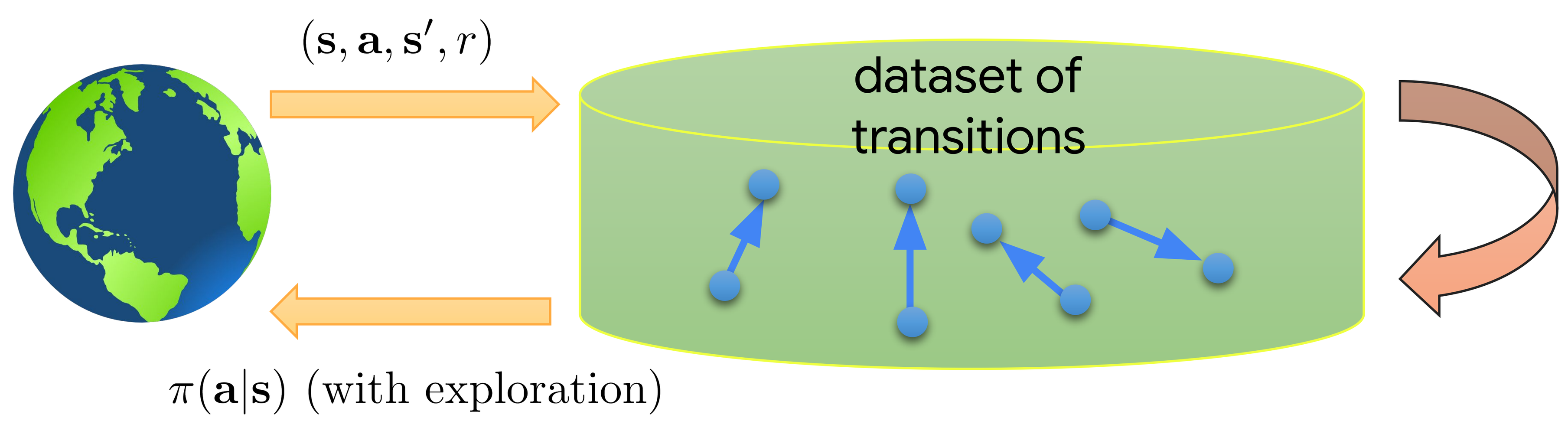
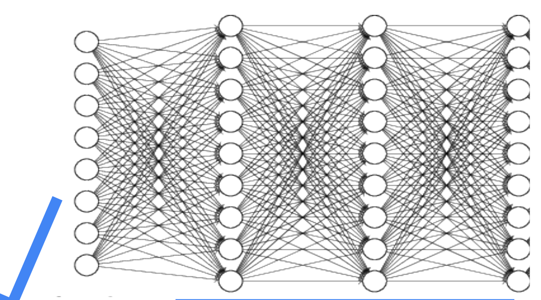
Implicit Under-Parameterization Inhibits Data Efficient Deep Reinforcement Learning

Aviral Kumar*, Rishabh Agarwal*, Dibya Ghosh and Sergey Levine

Modern Deep RL Algorithms





off-policy learning

Training deep neural net policies

Q-Learning

1. Train Q-functions by minimizing **TD Error**:

$$E_{(\mathbf{s},\mathbf{a})\sim\pi_\beta(\mathbf{s},\mathbf{a})}\left[(Q_\phi(\mathbf{s},\mathbf{a}) - (r(\mathbf{s},\mathbf{a}) + \gamma E[Q_\phi(\mathbf{s}',\mathbf{a}')]))^2\right]$$

2. Collect new data in the environment by rolling out the learned policy.

Typically solved "approximately" using gradient descent for a fixed number of steps

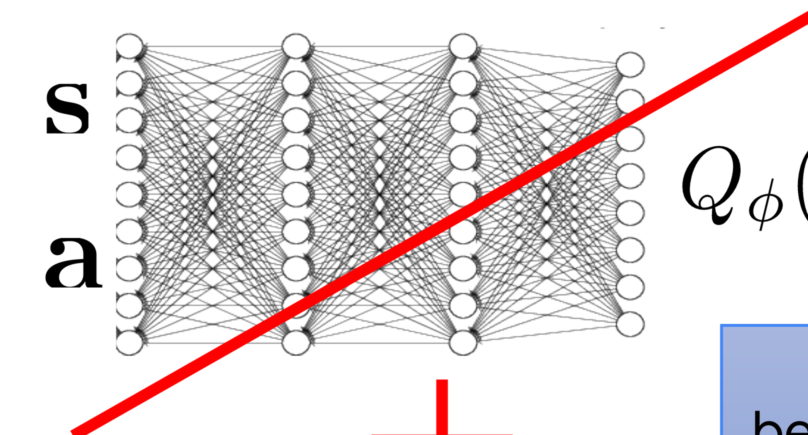
How does this approximate optimization procedure affect learning?

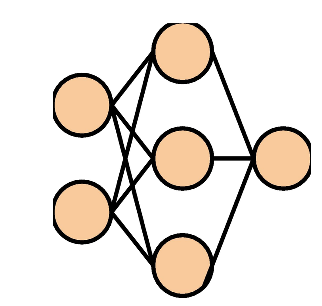
Implicit Under-Parameterization

TD Error (Regressing to itself)

$$E_{(\mathbf{s},\mathbf{a})\sim\pi_\beta(\mathbf{s},\mathbf{a})}\left[(Q_\phi(\mathbf{s},\mathbf{a}) - (r(\mathbf{s},\mathbf{a}) + \gamma E[Q_\phi(\mathbf{s}',\mathbf{a}')]))^2\right]$$

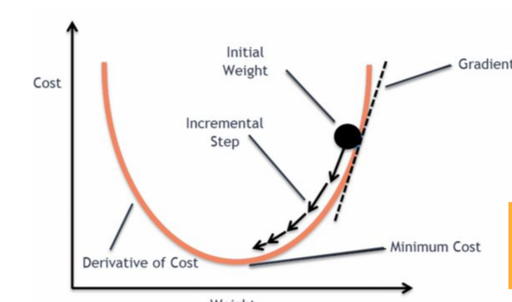
+





Big network **implicitly** behaves as a low-capacity, **under-parameterized** network

+



Gradient descent optimizer

Formalizing Implicit under-parameterization

$$Q_\phi(\mathbf{s},\mathbf{a}) = \mathbf{w}^T\Phi_\phi(\mathbf{s},\mathbf{a}) \qquad \Phi_\phi(\mathbf{s},\mathbf{a}) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times d}$$

Learned features

Effective rank

$$\text{srank}_\delta(\Phi) = \min\left\{k : \frac{\sum_{i=1}^k \sigma_i(\Phi)}{\sum_{i=1}^N \sigma_i(\Phi)} \geq 1 - \delta\right\}$$

Soft notion of rank of the matrix

Low rank => more aliasing => poor performance

Data-Efficient Deep Reinforcement Learning

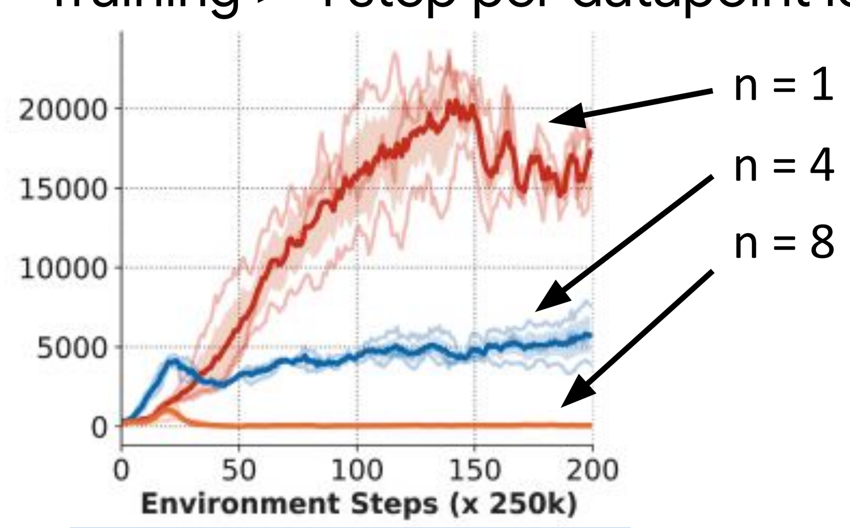
Data-Efficient Deep RL: Want to learn the most per unit amount of experience/data

How can we obtain data-efficiency?

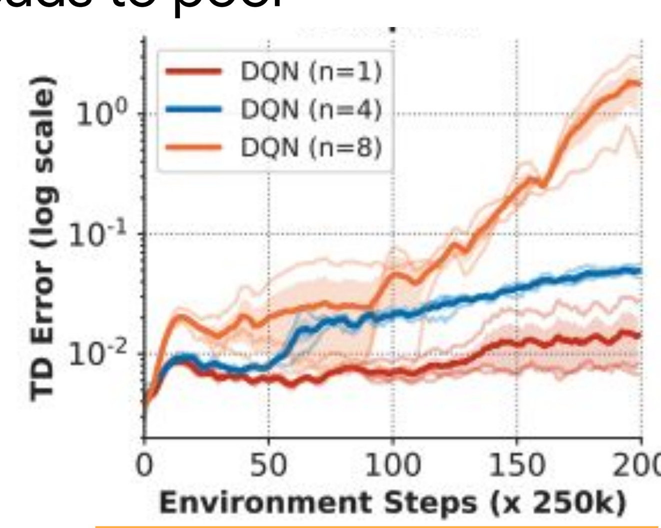
Typically 100-200 updates per datapoint

Supervised Learning:
Train more, control for statistical overfitting
Train error = 0, validation error = high

Reinforcement Learning:
Training >=1 step per datapoint leads to poor



OK, maybe I need to prevent overfitting?



But training error is high with larger n

Why do we see "underfitting" with more training?

Fu*, Kumar* et al. Diagnosing Bottlenecks in Deep Q-Learning Algorithms. ICML 2019.
Fedus*, Ramachandran*, Agarwal et al. Revisiting Fundamentals of Experience Replay. ICML 2020.

What Causes Implicit Under-Parameterization?

Gradient descent leads to low rank solutions

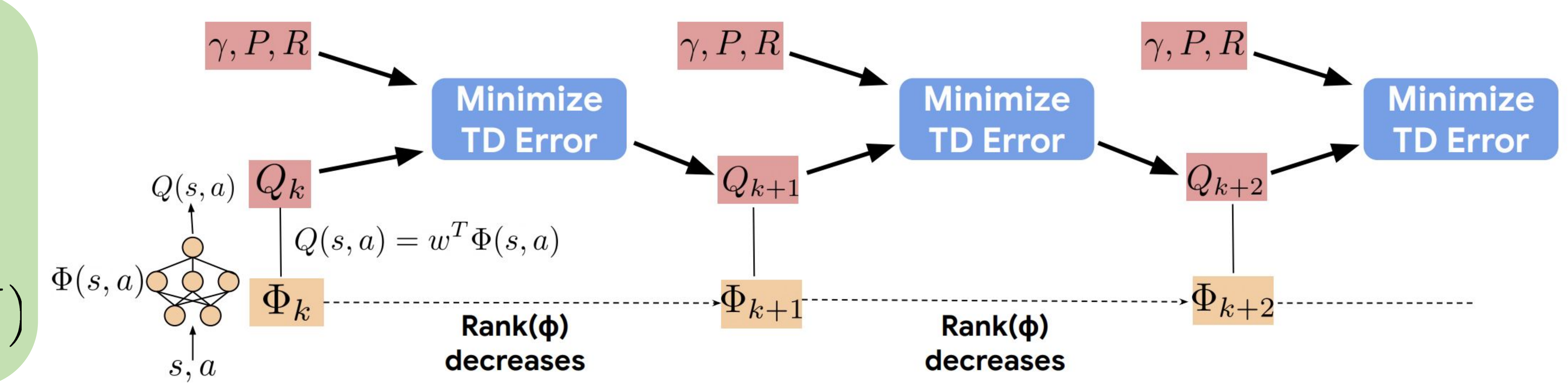
$$\min_X \;\; ||AX - y||_2^2$$

$$\min_X \; ||AX - y||_2^2 + \lambda f(X)$$

$f(X)$ makes large singular values larger.

$$\frac{\sigma_{\text{large}}}{\sigma_{\text{small}}} \uparrow \;\text{ over training}$$

Rank decrease effect due to supervised learning gets **compounded** due to bootstrapping



Bounded increase, that goes down with more training

$$\forall\, l \in \mathbb{N} \;\; and \;\; t \geq k_l, \;\; \frac{\sigma_i(\mathbf{M}_t)}{\sigma_j(\mathbf{M}_t)} < \frac{\sigma_i(\mathbf{M}_{k_l})}{\sigma_j(\mathbf{M}_{k_l})} + \mathcal{O}\left(\left(\frac{\sigma_i(\mathbf{S})}{\sigma_j(\mathbf{S})}\right)^{k_l}\right)$$

Singular value ratio at time t

Singular value ratio some time before t

..analysis with kernel regression and deep linear nets in the paper

agarwl.github.io/iup for details!