The Rotten Truth of Deep RL
The Rotten Truth of Deep RL

Deep RL can successfully solve tasks, but…

- Poor reliability over repeated runs

[Henderson et al, 2017a,b] [Lewis et al, 2018]
The Rotten Truth of Deep RL

Deep RL can successfully solve tasks, but…

- Poor reliability over repeated runs
- High sensitivity to hyperparameters

[Henderson et al, 2017a,b] [Lewis et al, 2018]
The Rotten Truth of Deep RL

Deep RL can successfully solve tasks, but...

- Poor reliability over repeated runs
- High sensitivity to hyperparameters
- Lack of robustness to environmental artifacts

[Henderson et al, 2017a,b] [Lewis et al, 2018]
The Rotten Truth of Deep RL

Deep RL can successfully solve tasks, but...

- Poor reliability over repeated runs
- High sensitivity to hyperparameters
- Lack of robustness to environmental artifacts

Notably, benchmarks don’t reveal these issues

[Henderson et al, 2017a,b] [Lewis et al, 2018]
What’s going on?

[Ilyas Engstrom Santurkar Tsipras Janoos Rudolph M 2018]
Implementation Obscures Deep RL Algorithms

Source: GitHub issues
Implementation Obscures Deep RL Algorithms

Maximum Reward

“Orthogonal” NN initialization

Without Optimization  With Optimization
Implementation Obscures Deep RL Algorithms

Maximum Reward

- Without Optimization
- With Optimization

“Orthogonal” NN initialization
Implementation Obscures Deep RL Algorithms

Without Optimization  With Optimization

Maximum Reward

“Orthogonal” NN initialization
Implementation Obscures Deep RL Algorithms

- Reward Normalization
- LR Annealing
- Orthogonal init
- Value Clipping

---

Maximum Reward

- Without Optimization
- With Optimization

- Values: 0, 300, 600, 900, 1200
Back to First Principles
Back to First Principles

- Gradient Estimates
Back to First Principles

- Gradient Estimates
- Value Prediction
Back to First Principles

‣ Gradient Estimates

‣ Value Prediction

‣ Loss Landscape
Back to First Principles

- Gradient Estimates
- Value Prediction
- Loss Landscape
- Trust Region
Back to First Principles

- Gradient Estimates
- Value Prediction
- Loss Landscape
- Trust Region
Key assumption of policy gradient framework:

\[ \mathbb{E}_{X \sim P}[X] \approx \frac{1}{N} \sum_{x_i \sim P} x_i \]
Gradient Estimation

Key assumption of policy gradient framework:

\[
\mathbb{E}_{X \sim P}[X] \approx \frac{1}{N} \sum_{x_i \sim P} x_i
\]

How well does this work?
Gradient Estimation

\( \theta_t \) (current policy parameters)
Gradient Estimation

$\theta_t$ (current policy parameters)

$g_t^{(1)}$
Gradient Estimation

\[ g_t^{(1)} = \frac{1}{k} \sum_{i=1}^{k} \ldots \]

(k-sample gradient estimate)

\[ \theta_t \] (current policy parameters)
Gradient Estimation

$\theta_t$ (current policy parameters)

$g_t^{(1)}$

$g_t^{(2)}$
Gradient Estimation

\( g_t^{(1)} \)

\( g_t^{(2)} \)

\( g_t^{(3)} \)

\( \theta_t \) (current policy parameters)
Gradient Estimation

$\theta_t$ (current policy parameters)

$g_t^{(1)}$ $g_t^{(2)}$ $g_t^{(3)}$
Gradient Estimation

Gradient Variance (pairwise correlation)

$\theta_t$ (current policy parameters)
Gradient Estimation

\( \theta_t \) (current policy parameters)

\( g_t \)

\( g_t^{(1)} \)

\( g_t^{(2)} \)

\( g_t^{(3)} \)

\( g_t^{(*)} \)
Gradient Estimation

\[ g_t^{(*)} = \frac{1}{10^7} \sum_{i=1}^{10^7} \cdots \]

("true gradient")

\[ g_t = \frac{1}{10^7} \sum_{i=1}^{10^7} \cdots \]

\[ \theta_t \] (current policy parameters)
Gradient Estimation

\[ g_t^{(*)} = \frac{1}{10^7} \sum_{i=1}^{10^7} \cdots \]

("true gradient")

\[ g_t = \frac{1}{10^7} \sum_{i=1}^{10^7} \cdots \]

\( \theta_t \) (current policy parameters)
Gradient Estimation

\[ g_t^{(*)} = \frac{1}{10^7} \sum_{i=1}^{10^7} \ldots \]

(“true gradient”)

\[ g_t = \frac{1}{10^7} \sum_{i=1}^{10^7} \ldots \]

\( \theta_t \) (current policy parameters)

Gradient Concentration (\( g^* \) correlation)
Gradient Variance

- Black line: relevant sample regime
- Gradients are less concentrated than they could be
- Less correlated for “harder” tasks, later iterations
Gradient Concentration

- Black line: relevant sample regime
- Gradients are less concentrated than they could be
- Less correlated for “harder” tasks, later iterations
Gradient Estimation

- No good understanding of training dynamics
  - How does variance influence optimization?
  - Can we use insights from stochastic opt?
- Missing a link from reliability to sample size
Value Prediction
Value Prediction

Policy gradient is a sum weighted by returns
Value Prediction

Policy gradient is a sum weighted by returns

Concentration is hindered by high variance
Value Prediction

Policy gradient is a sum weighted by returns

Concentration is hindered by high variance

Observation: If we can estimate the value of a state, can significantly lower variance
Value Prediction

Variance reduction needs good value estimates

In Deep RL, values come from a neural network

To what degree do we actually reduce variance?
The agent’s value network helps in variance reduction, but not nearly as much as the true value function.
Value Prediction

- Might look small, but using a value network makes big difference
- How would using the true value affect training?
- Can we get better value estimates (info barrier)
More analysis (from the paper)

Similar conclusions from:

Optimization landscape is often noisy/misleading

Enforcement of “trust regions” has theoretical and practical caveats
Does AI translate from simulation to reality?

Simulation

This is Google's DeepMind AI teaching itself how to walk
Does AI translate from simulation to reality?

Simulation

Reality

This is Google's DeepMind AI teaching itself how to walk

Also: Are we even optimizing the right thing?
Takeaways
How do we proceed?

- Reconciling RL with our conceptual framework
  - How predictive are theoretical principles in practice?
  - What is the right way to model the RL setting?
- Rethinking primitives for modern settings
  - How do we deal with high dimensionality?
  - Delayed rewards?
- Better evaluation for RL systems
  - Benchmarks don’t capture reliability, safety, or robustness of RL agents