Challenges in Deep Reinforcement Learning

Sergey Levine
UC Berkeley
Real-World Experiments

**Not** accounting for uncertainty
(higher-speed collisions)
• Discuss some recent work in deep reinforcement learning

• Present a few major challenges

• Show some of our recent work toward tackling these challenges
Some recent work on deep RL

- Deep Q-Networks
  Mnih et al. 2013

- Guided policy search
  Levine et al. 2013

- RL on raw visual input
  Lange et al. 2009

- Trust region policy optimization
  Schulman et al. 2015

- Deep deterministic policy gradients
  Lillicrap et al. 2015

- End-to-end visuomotor policies
  Levine*, Finn* et al. 2015

- AlphaGo
  Silver et al. 2016

- Supersizing self-supervision
  Pinto & Gupta 2016

Keywords: stability, efficiency, scale
Challenges in Deep Reinforcement Learning

1. Stability
2. Efficiency
3. Scale
Challenges in Deep Reinforcement Learning

1. Stability

2. Efficiency

3. Scale
Deep RL with Policy Gradients

- Unbiased but high-variance gradient
- Stable
- Requires many samples
- Example: TRPO [Schulman et al. ‘15]

\[
\nabla_{\theta} J(\theta) = E_{\pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(u_t|x_t) \hat{Q}(x_t, u_t)]
\]

\[
\hat{Q}(x_t, u_t) \approx \sum_{t'=t}^{\infty} \gamma^{t-t'} r(x_t, u_t)
\]
Deep RL with Off-Policy Q-Function Critic

- Low-variance **but biased** gradient
- Much more efficient (because off-policy)
- Much less stable (because biased)
- Example: DDPG [Lillicrap et al. ‘16]

\[
Q_w \leftarrow \min_{w} E \left[ (Q_w(x_t, u_t) - (r(x_t, u_t) + \gamma Q_w(x_{t+1}, \pi_{\theta}(x_t)))^2 \right]
\]

\[
\nabla_{\theta} J(\theta) = E[\nabla_{u_t} Q_w(x_t, \pi_{\theta}(x_t)) \nabla_{\theta} \pi_{\theta}(x_t)]
\]
Improving Efficiency & Stability with Q-Prop

Policy gradient:
\[ \nabla_\theta J(\theta) = E_{\pi_\theta} \left[ \nabla_\theta \log \pi_\theta(u_t | x_t) \hat{Q}(x_t, u_t) \right] \]

Q-function critic:
\[ \nabla_\theta J(\theta) = E \left[ \nabla_{u_t} Q_w(x_t, \mu_\theta(x_t)) \nabla_\theta \mu_\theta(x_t) \right] \]

Q-Prop:
\[ \nabla_\theta J(\theta) = E_{\pi_\theta(x_t)} \left[ \nabla_{u_t} \nabla_{x_t} Q(x_t, \mu_\theta(x_t))(\nabla_{u_t} \mu_\theta(x_t))(x_t) \right] + \]
\[ E_{\pi_\theta(x_t, u_t)} \left[ \nabla_\theta \log \pi_\theta(u_t | x_t) (\hat{Q}(x_t, u_t) - \tilde{Q}(x_t, \mu_\theta(x_t))) (u_t - \mu_\theta(x_t)) \right] \]

\[ \tilde{Q}(x_t, u_t) = \nabla_{u_t} Q(x_t, \mu_\theta(x_t))(u_t - \mu_\theta(x_t)) \]

- Unbiased gradient, stable
- Efficient (uses off-policy samples)
- Critic comes from off-policy data
- Gradient comes from on-policy data
- Automatic variance-based adjustment
Comparisons

- Works with smaller batches than TRPO
- More efficient than TRPO
- More stable than DDPG with respect to hyperparameters
  - Likely responsible for the better performance on harder task
Challenges in Deep Reinforcement Learning

1. Stability

2. Efficiency

3. Scale
Parameter Space vs Policy Space

why policy space?
• local optima/easier optimization landscapes
• can be easier to update in policy space vs parameter space
Mirror Descent Guided Policy Search (MDGPS)

\[
\min_{\bm{x}} f(\bm{x}) \quad \text{s.t.} \quad \bm{x} \in \mathcal{X}
\]

\[
\bm{x}^{k+\frac{1}{2}} \leftarrow \min_{\bm{x}} \hat{f}(\bm{x}) \quad \text{s.t.} \quad D(\bm{x}, \bm{x}^k) \leq \epsilon
\]

\[
\bm{x}^{k+1} \leftarrow \min_{\bm{x}} D(\bm{x}, \bm{x}^{k+\frac{1}{2}}) \quad \text{s.t.} \quad \bm{x} \in \mathcal{X}
\]
Mirror Descent Guided Policy Search (MDGPs)

\[
\min_{x} J(x) \quad \text{s.t.} \quad \pi_{\theta} \in \Pi_{\theta}
\]

\[
x^{k+\frac{1}{2}} \leftarrow \min_{x} \hat{J}(x) \quad \text{s.t.} \quad D(\nu(x) || \pi^{k+\frac{1}{2}}) \leq \epsilon
\]

\[
x^{k+1} \leftarrow \min_{\theta} D(\nu(x^{k+\frac{1}{2}} || \pi^{k+\frac{1}{2}})) \quad x \in \mathcal{X}
\]

"projection": supervised learning

local policy optimization:

- trajectory-centric model-based RL [Montgomery ‘16]
- path integral policy iteration [Chebotar ‘16]
MDGPS with Random Initial States and Local Models

\[
\begin{align*}
\min_{\pi} & \ J(\pi) \text{ s.t. } \pi \in \Pi_{\theta} \\
\pi_{k+\frac{1}{2}} & \leftarrow \min_{\pi} \hat{J}(\pi) \text{ s.t. } D_{\text{KL}}(\pi \parallel \pi_{k}^{\theta}) \leq \epsilon \\
\pi_{\theta}^{k+1} & \leftarrow \min_{\theta} D_{\text{KL}}(\pi_{\theta} \parallel \pi_{k}^{\frac{1}{2}})
\end{align*}
\]

1. Fit \( N \) Gaussian trajectory distributions \( p_i(\tau) \)

2. For each distribution fit \( p_i(x_{t+1}|x_t, u_t) \) as time-varying linear-Gaussian.

3. Update time-varying linear-Gaussian \( \pi_i(u_t|x_t) \) using LQR with KL constraint.

Use supervised learning to train neural net \( \pi_{\theta}(u_t|x_t) \) to mimic all \( N \) “local policies” \( \pi_i(u_t|x_t) \)
Learning 2D reaching (simple benchmark task):
• TRPO (best known value): 3000 trials
• DDPG, NAF (best known value): 2000 trials
• Q-Prop: 2000 trials
• MDGPS: 500 trials
2. For each distribution fit \( p_i(x_{t+1}|x_t, u_t) \) as time-varying linear-Gaussian.

3. Update time-varying linear-Gaussian \( \pi_i(u_t|x_t) \) using LQR with KL constraint.

Update time-varying linear-Gaussian \( \pi_i(u_t|x_t) \) using PI\(^2\) algorithm:

\[
E[u_t] = \sum_i w_{it} u_{it}
\]

\[
w_{it} \propto \exp \left(-\beta \sum_{t'=t}^{T} c(x_{t'}, u_{t'})\right)
\]

+ much better handling of non-smooth problems (e.g. discontinuities)

- requires more samples, works best with demo initialization
Challenges in Deep Reinforcement Learning

1. Stability

2. Efficiency

3. Scale
ingredients for success in learning:

- supervised learning:
  - ✔ computation
  - ✔ algorithms
  - ✔ data

- reinforcement learning:
  - ✔ computation
  - ~ algorithms
  - ? data

L., Pastor, Krizhevsky, Quillen ‘16
Policy Learning with Multiple Robots

\[ \min_{\pi} J(\pi) \text{ s.t. } \pi \in \Pi_\theta \]

\[ \pi^{k+\frac{1}{2}} \leftarrow \min_{\pi} \hat{J}(\pi) \quad \pi^{k+1}_\theta \leftarrow \min_{\theta} D_{KL}(\pi_\theta \| \pi^{k+\frac{1}{2}}) \]
Policy Learning with Multiple Robots: Deep RL with NAF

\[
Q(x, u | \theta^Q) = A(x, u | \theta^A) + V(x | \theta^V) \\
A(x, u | \theta^A) = -\frac{1}{2}(u - \mu(x | \theta^\mu))^T P(x | \theta^P)(u - \mu(x | \theta^\mu))
\]
Future Outlook & Future Challenges

- Stability remains a huge challenge
  - Can’t do hyperparameter sweeps in the real world...
  - Likely missing a few more pieces of theory

- High efficiency is important, but what about diversity?
  - Efficiency seems at odds with generalization
  - Massively off-policy learning
  - Semi-supervised learning

- (not addressed in this talk) What about the reward function? Highly nonobvious how to set in the real world
Acknowledgements

Harley Montgomery

Anurag Ajay

Chelsea Finn

Shane Gu

Ethan Holly

Tim Lillicrap

Ali Yahya

Adrian Li

Mrinal Kalakrishnan

Yevgen Chebotar