Improving Bootstrapped DQN

Roman Knyazhitskiy

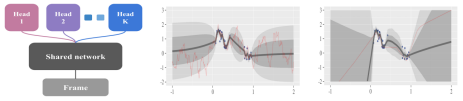
TLDR

- Exhaustive evaluation of epsilon-greedy vs bootstrapped vs bootstrapped with RP DQNs
- Implementing a VI method to sample DQNs from a stationary distribution of optimizers

Bootstrapped DQN

An ensemble method for epistemic exploration [1]:

1. Multiple network heads, same core network
2. Trained on distinct subsets of data



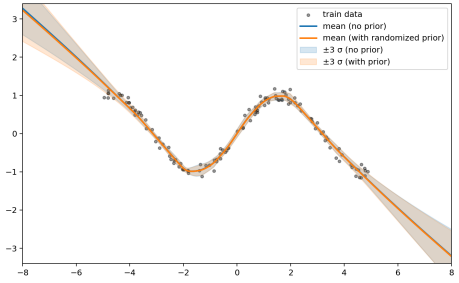
(a) Shared network architecture (b) Gaussian process posterior (c) Bootstrapped neural nets

The need for randomized priors

Problem: the networks are correlated

Solution: add a random function to DQN output, to improve diversity [2]

But isn't it just equivalent to different initial parameters? Is there a measurable effect?



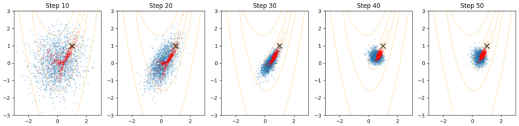
Stationary distribution of DQN weights

Optimizers converge to a stationary distribution [3], not a fixed solution, which is what makes bootstrapped DQN work.

The stationary distribution of SGD can be specified as:

$$p(\theta) \propto \exp(-\mathcal{L}(\theta))$$

Sampling directly allows to get "trained" DQN models, but without gradient descent!



SVI on stationary distribution

Two main ways to sample from a posterior (stationary distribution, in our case) is via either VI [4] or MCMC [5]

Training a VI model on a non-stationary target is hard. **What can be done?**

References

[1] Osband, I., Blundell, C., Pritzel, A., & Van Roy, B. (2016). Deep Exploration via Bootstrapped DQN

[2] Osband, I., Aslanides, J., & Cassirer, A. (2018). Randomized Prior Functions for Deep Reinforcement Learning

[3] Mandt, S., Hoffman, M. D., & Blei, D. M. (2017). Stochastic Gradient Descent as Approximate Bayesian Inference

[4] Welling, M., & Teh, Y. W. (2011). Bayesian Learning via Stochastic Gradient Langevin Dynamics

[5] Ishfaq, H., Lan, Q., Xu, P., Mahmood, A. R., Precup, D., Anandkumar, A., & Azizzadenesheli, K. (2023). Provable and Practical: Efficient Exploration in Reinforcement Learning via Langevin Monte Carlo