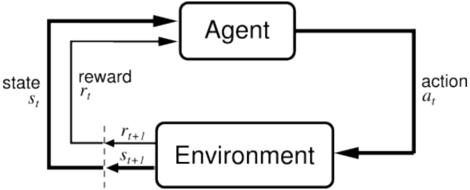
Improving Stability in Deep Reinforcement Learning with Weight Averaging

Evgenii Nikishin

10.09.2018

RL problem statement

Markov Decision Process (MDP):



- Environment states $s_t \in \mathcal{S}$
- Agent actions $a_t \in \mathcal{A}$
- Reward $r(s_t, a_t) \in \mathbb{R}$
- Agent policy $a_t \sim \pi(a_t|s_t)$
- State transitions $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$

RL problem statement

Interaction with an environment produces trajectory τ:

$$p_\pi(\tau) = p(s_0) \prod_{t=0}^{T} \pi(a_t|s_t) p(s_{t+1}|s_t, a_t)$$

Optimal policy maximizes the expected discounted return:

$$\pi^* = \arg\max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right] = \arg\max_\pi \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right]$$

$0 \leq \gamma \leq 1$ (0.99 is common)

Policy gradient methods

Parametrize policy $\pi(a|s) = \pi_\theta(a|s)$ within a family of differentiable functions and update w.r.t. its gradient:

$$\nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right]$$

Policy gradient methods

Parametrize policy $\pi(a|s) = \pi_\theta(a|s)$ within a family of differentiable functions and update w.r.t. its gradient:

$$\nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right] = \sum_\tau \nabla_\theta p_\theta(\tau) \sum_{t=0}^{T} \gamma^t r(s_t, a_t) =$$

Policy gradient methods

Parametrize policy $\pi(a|s) = \pi_\theta(a|s)$ within a family of differentiable functions and update w.r.t. its gradient:

$$\nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right] = \sum_\tau \nabla_\theta p_\theta(\tau) \sum_{t=0}^{T} \gamma^t r(s_t, a_t) =$$

$$\sum_\tau p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) \sum_{t=0}^{T} \gamma^t r(s_t, a_t) =$$

Policy gradient methods

Parametrize policy $\pi(a|s) = \pi_\theta(a|s)$ within a family of differentiable functions and update w.r.t. its gradient:

$$\nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right] = \sum_\tau \nabla_\theta p_\theta(\tau) \sum_{t=0}^{T} \gamma^t r(s_t, a_t) =$$

$$\sum_\tau p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) \sum_{t=0}^{T} \gamma^t r(s_t, a_t) =$$

$$\left\{ \log p_\theta(\tau) = \log p(s_0) + \sum_{t=0}^{T} (\log \pi_\theta(a_t|s_t) + \log p(s_{t+1}|s_t, a_t)) \right\} =$$

Policy gradient methods

Parametrize policy $\pi(a|s) = \pi_\theta(a|s)$ within a family of differentiable functions and update w.r.t. its gradient:

$$\nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right] = \sum_\tau \nabla_\theta p_\theta(\tau) \sum_{t=0}^{T} \gamma^t r(s_t, a_t) =$$

$$\sum_\tau p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) \sum_{t=0}^{T} \gamma^t r(s_t, a_t) =$$

$$\left\{ \log p_\theta(\tau) = \log p(s_0) + \sum_{t=0}^{T} \left(\log \pi_\theta(a_t|s_t) + \log p(s_{t+1}|s_t, a_t) \right) \right\} =$$

$$\mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\left(\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \right) \left(\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right) \right]$$

Policy gradient methods

$$\mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\left(\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \right) \left(\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right) \right]$$

Main problem: large variance of gradient estimates (much bigger than in supervised learning)

Policy gradient methods

$$\mathbb{E}_{\tau\sim p_\theta(\tau)}\left[\left(\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)\right)\left(\sum_{t=0}^{T}\gamma^t r(s_t,a_t)\right)\right]$$

Main problem: large variance of gradient estimates (much bigger than in supervised learning) Example:

$$r(s_t,a_t)=\begin{cases}-110 \text{ prob } 0.5\\ -100, \text{ prob } 0.5\end{cases}\qquad \text{Mean = -105, Var = 25}$$

$$\nabla_\theta \log \pi_\theta(a_t|s_t)=\begin{cases}-1, \text{ prob } 0.5\\ +1, \text{ prob } 0.5\end{cases}\qquad \text{Mean = 0, Var = 1}$$

$$\nabla_\theta \log \pi_\theta(a_t|s_t) r(s_t,a_t)=\begin{cases}-110, \text{ prob } 0.25\\ -100, \text{ prob } 0.25\\ +100, \text{ prob } 0.25\\ +110, \text{ prob } 0.25\end{cases}\qquad \text{Mean = 0, Var = 11050}$$

Dealing with noise

Possible ways to alleviate effect of large variance:

1. Introduce a baseline, that does not change the expectation, but can decrease variance (e.g. average reward)

Dealing with noise

Possible ways to alleviate effect of large variance:

1. Introduce a baseline, that does not change the expectation, but can decrease variance (e.g. average reward)
2. Introduce some bias in order to decrease the variance (Actor-Critic algorithm [Mnih et al., 2016])

Dealing with noise

Possible ways to alleviate effect of large variance:

1. Introduce a baseline, that does not change the expectation, but can decrease variance (e.g. average reward)
2. Introduce some bias in order to decrease the variance (Actor-Critic algorithm [Mnih et al., 2016])
3. Penalise for large policy deviations (Trust Region Policy Optimization [Schulman et al., 2015])

Dealing with noise

Possible ways to alleviate effect of large variance:

1. Introduce a baseline, that does not change the expectation, but can decrease variance (e.g. average reward)
2. Introduce some bias in order to decrease the variance (Actor-Critic algorithm [Mnih et al., 2016])
3. Penalise for large policy deviations (Trust Region Policy Optimization [Schulman et al., 2015])
4. ...more?

SWA

Stochastic Weight Averaging [Izmailov et al., 2018]: average the weights collected during training with an SGD-like method.

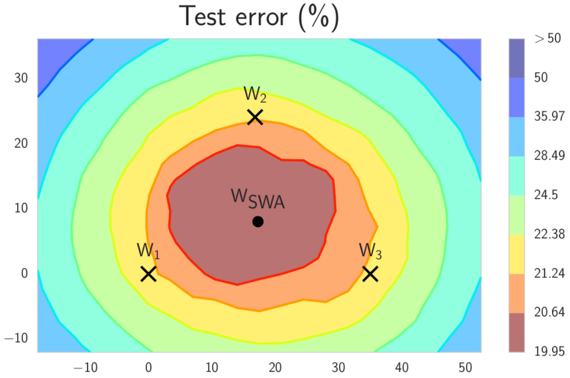
1. start after conventional pretraining
2. use constant or cyclical learning rate
   (to continue exploring regions of high-performing networks)
3. dynamically recalculate average:

$$w_{\text{SWA}} \leftarrow \frac{n_{\text{SWA}} \cdot w_{\text{SWA}} + w}{n_{\text{SWA}} + 1}$$

$$n_{\text{SWA}} \leftarrow n_{\text{SWA}} + 1$$

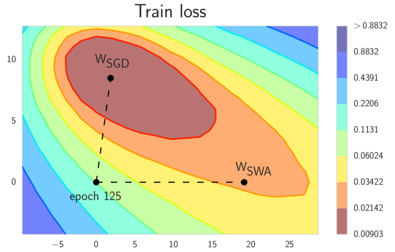
Motivation in SL

Mandt et al. [2017]: SGD with fixed learning rate samples from a Gaussian distribution centered at the minimum of the loss, i.e. SGD iterates stay at the boundary of a high-quality region:



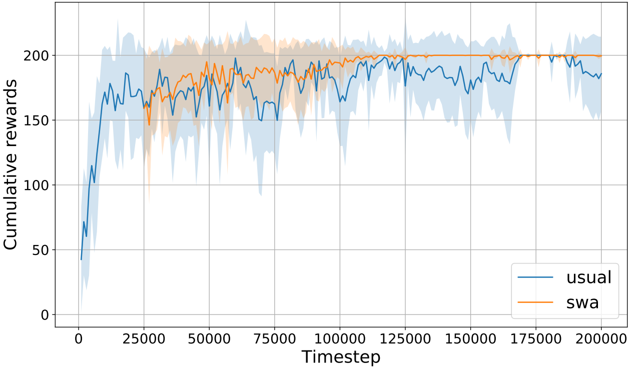
Motivation in SL

Due to shift between train and test loss surfaces, we are looking for wider optima. SWA leads to better generalization:



Motivation in RL

Deep RL methods are notoriously unstable.
A2C cumulative reward trajectory on CartPole environment with and without weight averaging:



SWA alleviates the effect of noise during training.

SWA for A2C and DDPG

We apply SWA to

- Advantage Actor-Critic [Mnih et al., 2016] for discrete action space environments (Atari games).
- Deep Deterministic Policy Gradient [Lillicrap et al., 2015] for continuous action space environments (MuJoCo).

After pretraining with conventional training, we apply SWA by collecting weights every c timesteps.

Atari experiments

Table: Average final cumulative reward with and without SWA.

ENV NAME	A2C	A2C + SWA
Breakout	522 ± 34	$\mathbf{703 \pm 60}$
Qbert	18777 ± 778	$\mathbf{21272 \pm 655}$
SpaceInvaders	7727 ± 1121	$\mathbf{21676 \pm 8897}$
Seaquest	1779 ± 4	$\mathbf{1795 \pm 4}$
CrazyClimber	$\mathbf{147030 \pm 10239}$	139752 ± 11618
BeamRider	9999 ± 402	$\mathbf{11321 \pm 1065}$

MuJoCo experiments

Table: Average final cumulative reward with and without SWA.

ENV NAME	DDPG	DDPG + SWA
Hopper	613 ± 683	$\mathbf{1615} \pm \mathbf{1143}$
Walker2d	1803 ± 96	$\mathbf{2457} \pm \mathbf{241}$
Half-Cheetah	3825 ± 1187	$\mathbf{4228} \pm \mathbf{1117}$
Ant	865 ± 899	$\mathbf{1051} \pm \mathbf{696}$

Summary

RL + SWA:

- Easy to implement
- Sample-efficient way to improve stability
- Alleviates problem of forgetting good policies

References

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *The Journal of Machine Learning Research*, 18(1):4873–4907, 2017.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.