Smoothed Action Value Functions for Learning Gaussian Policies (Supplementary Material)

Ofir Nachum [1] **Mohammad Norouzi** [1] **George Tucker** [1] **Dale Schuurmans** [1 2]

[1]Google Brain [2]Department of Computing Science, University of Alberta. Correspondence to: Ofir Nachum <ofirnachum@google.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

A. Proof of Theorem 1

We want to show that for any s, a,

$$\frac{\partial \tilde{Q}^\pi(s,a)}{\partial \Sigma(s)} = \frac{1}{2} \cdot \frac{\partial^2 \tilde{Q}^\pi(s,a)}{\partial a^2} \tag{1}$$

We note that similar identities for Gaussian integrals exist in the literature (Price, 1958; Rezende et al., 2014) and point the reader to these works for further information.

Proof. The specific identity we state may be derived using standard matrix calculus. We make use of the fact that

$$\frac{\partial}{\partial A}|A|^{-1/2} = -\frac{1}{2}|A|^{-3/2}\frac{\partial}{\partial A}|A| = -\frac{1}{2}|A|^{-1/2}A^{-1}, \tag{2}$$

and for symmetric A,

$$\frac{\partial}{\partial A}||v||^2_{A^{-1}} = -A^{-1}vv^TA^{-1}. \tag{3}$$

We omit s from $\Sigma(s)$ in the following equations for succinctness. The LHS of (1) is

$$\begin{aligned}
&\int_{\mathcal{A}} Q^\pi(s,\tilde{a})\frac{\partial}{\partial \Sigma}N(\tilde{a}|a,\Sigma)\mathrm{d}\tilde{a} \\
&= \int_{\mathcal{A}} Q^\pi(s,\tilde{a}) \exp\left\{-\frac{1}{2}||\tilde{a}-a||^2_{\Sigma^{-1}}\right\}\left(\frac{\partial}{\partial \Sigma}|2\pi\Sigma|^{-1/2} - \frac{1}{2}|2\pi\Sigma|^{-1/2}\frac{\partial}{\partial \Sigma}||\tilde{a}-a||^2_{\Sigma^{-1}}\right)\mathrm{d}\tilde{a} \\
&= \frac{1}{2}\int_{\mathcal{A}} Q^\pi(s,\tilde{a})N(\tilde{a}|a,\Sigma)\left(-\Sigma^{-1} + \Sigma^{-1}(\tilde{a}-a)(\tilde{a}-a)^T\Sigma^{-1}\right)\mathrm{d}\tilde{a}.
\end{aligned}$$

Meanwhile, towards tackling the RHS of (1) we note that

$$\frac{\partial \tilde{Q}^\pi(s,a)}{\partial a} = \int_{\mathcal{A}} Q^\pi(s,\tilde{a})N(\tilde{a}|a,\Sigma)\Sigma^{-1}(\tilde{a}-a)\mathrm{d}\tilde{a}\ . \tag{4}$$

Thus we have

$$\begin{aligned}
\frac{\partial^2 \tilde{Q}^\pi(s,a)}{\partial a^2} &= \int_{\mathcal{A}} Q^\pi(s,\tilde{a})\left(\Sigma^{-1}(\tilde{a}-a)\frac{\partial}{\partial a}N(\tilde{a}|a,\Sigma) + N(\tilde{a}|a,\Sigma)\frac{\partial}{\partial a}\Sigma^{-1}(\tilde{a}-a)\right)\mathrm{d}\tilde{a} \\
&= \int_{\mathcal{A}} Q^\pi(s,\tilde{a})N(\tilde{a}|a,\Sigma)(\Sigma^{-1}(\tilde{a}-a)(\tilde{a}-a)^T\Sigma^{-1} - \Sigma^{-1})\,\mathrm{d}\tilde{a}\ .
\end{aligned}$$

■

B. Compatible Function Approximation

We claim that a $\tilde{Q}^\pi_w$ is compatible with respect to μ_θ if

1. $\nabla_a \tilde{Q}^\pi_w(s,a)\big|_{a=\mu_\theta(s)} = \nabla_\theta \mu_\theta(s)^T w$,
2. $\nabla_w \int_{\mathcal{S}} \left(\nabla_a \tilde{Q}^\pi_w(s,a)\big|_{a=\mu_\theta(s)} - \nabla_a \tilde{Q}^\pi(s,a)\big|_{a=\mu_\theta(s)}\right)^2 \mathrm{d}\rho^\pi(s) = 0$ (*i.e.,* w minimizes the expected squared error of the gradients).

Additionally, $\tilde{Q}^\pi_w$ is compatible with respect to Σ_ϕ if

1. $\nabla^2_a \tilde{Q}^\pi_w(s,a)\big|_{a=\mu_\theta(s)} = \nabla_\phi \Sigma_\phi(s)^T w$,

2. $\nabla_w \int_{\mathcal{S}} \left(\nabla_a^2 \tilde{Q}_w^\pi(s,a)\big|_{a=\mu_\theta(s)} - \nabla_a^2 \tilde{Q}^\pi(s,a)\big|_{a=\mu_\theta(s)} \right)^2 \mathrm{d}\rho^\pi(s) = 0$ (*i.e.,* w minimizes the expected squared error of the Hessians).

Proof. We shall show how the conditions stated for compatibility with respect to Σ_ϕ are sufficient. The reasoning for μ_θ follows via a similar argument. We also refer the reader to Silver et al. (2014) which includes a similar procedure for showing compatibility.

From the second condition for compatibility with respect to Σ_ϕ we have

$$\int_{\mathcal{S}} \left(\nabla_a^2 \tilde{Q}_w^\pi(s,a)\big|_{a=\mu_\theta(s)} - \nabla_a^2 \tilde{Q}^\pi(s,a)\big|_{a=\mu_\theta(s)} \right) \nabla_w \left(\nabla_a^2 \tilde{Q}_w^\pi(s,a)\big|_{a=\mu_\theta(s)} \right) \mathrm{d}\rho^\pi(s) \quad = \quad 0 \, .$$

We may combine this with the first condition to find

$$\int_{\mathcal{S}} \nabla_a^2 \tilde{Q}_w^\pi(s,a)\big|_{a=\mu_\theta(s)} \nabla_\phi \Sigma_\phi(s) \mathrm{d}\rho^\pi(s) \quad = \quad \int_{\mathcal{S}} \nabla_a^2 \tilde{Q}^\pi(s,a)\big|_{a=\mu_\theta(s)} \nabla_\phi \Sigma_\phi(s) \mathrm{d}\rho^\pi(s) \, ,$$

which is the desired property for compatibility. ■

C. Derivative Bellman Equations

The conditions for compatibility require training $\tilde{Q}_w^\pi$ to fit the true $\tilde{Q}^\pi$ with respect to derivatives. Howevever, in RL contexts, one often does not have access to the derivatives of the true $\tilde{Q}^\pi$. In this section, we elaborate on a method to train $\tilde{Q}_w^\pi$ to fit the derivatives of the true $\tilde{Q}^\pi$ without access to true derivative information.

Our method relies on a novel formulation: *derivative Bellman equations*. We begin with the standard $\tilde{Q}^\pi$ Bellman equation presented in the main paper:

$$\tilde{Q}^\pi(s,a) = \int_{\mathcal{A}} N(\tilde{a} \,|\, a, \Sigma(s))\, \mathbb{E}_{\tilde{r},\tilde{s}'} \left[\tilde{r} + \gamma \tilde{Q}^\pi(\tilde{s}', \mu(\tilde{s}')) \right] \mathrm{d}\tilde{a} \, . \tag{5}$$

One may take derivatives of both sides to yield the following identity for any k:

$$\frac{\partial^k \tilde{Q}^\pi(s,a)}{\partial a^k} = \int_{\mathcal{A}} \frac{\partial^k N(\tilde{a} \,|\, a, \Sigma(s))}{\partial a^k} \, \mathbb{E}_{\tilde{r},\tilde{s}'} \left[\tilde{r} + \gamma \tilde{Q}^\pi(\tilde{s}', \mu(\tilde{s}')) \right] \mathrm{d}\tilde{a} \, . \tag{6}$$

One may express the k-the derivative of a normal density for $k \le 2$ simply as

$$\frac{\partial^k N(\tilde{a} \,|\, a, \Sigma(s))}{\partial a^k} = N(\tilde{a} \,|\, a, \Sigma(s)) \Sigma(s)^{-k/2} \cdot H_k(\Sigma(s)^{-1/2}(\tilde{a} - a)), \tag{7}$$

where H_k is a polynomial. Therefore, we have the following derivative Bellman equations for any $k \le 2$:

$$\frac{\partial^k \tilde{Q}^\pi(s,a)}{\partial a^k} = \int_{\mathcal{A}} N(\tilde{a} \,|\, a, \Sigma(s)) \Sigma(s)^{-k/2} \cdot H_k(\Sigma(s)^{-1/2}(\tilde{a} - a))\, \mathbb{E}_{\tilde{r},\tilde{s}'} \left[\tilde{r} + \gamma \tilde{Q}^\pi(\tilde{s}', \mu(\tilde{s}')) \right] \mathrm{d}\tilde{a} \, . \tag{8}$$

One may train a parameterized $\tilde{Q}_w^\pi$ to satisfy these consistencies in a manner similar to that described in Section 4.2. Specifically, suppose one has access to a tuple $(s, \tilde{a}, \tilde{r}, \tilde{s}')$ sampled from a replay buffer with knowledge of the sampling probability $q(\tilde{a} \,|\, s)$ (possibly unnormalized) with full support. Then we draw a *phantom* action $a \sim N(\tilde{a}, \Sigma(s))$ and optimize $\tilde{Q}_w^\pi(s,a)$ by minimizing a weighted derivative Bellman error

$$\frac{1}{q(\tilde{a}|s)} \left(\frac{\partial^k \tilde{Q}_w^\pi(s,a)}{\partial a^k} - \Sigma(s)^{-k/2} \cdot H_k(\Sigma(s)^{-1/2}(a - \tilde{a}))(\tilde{r} + \gamma \tilde{Q}_w^\pi(\tilde{s}', \mu(\tilde{s}'))) \right)^2 , \tag{9}$$

for $k = 0, 1, 2$. As in the main text, it is possible to argue that when using target networks, this training procedure reaches an optimum when $\tilde{Q}_w^\pi(s,a)$ satisfies the recursion in the derivative Bellman equations (8) for $k = 0, 1, 2$.

Hyperparameter	Range	Sampling
actor learning rate	[1e-6,1e-3]	log
critic learning rate	[1e-6,1e-3]	log
reward scale	[0.01,0.3]	log
OU damping	[1e-4,1e-3]	log
OU stddev	[1e-3,1.0]	log
λ	[1e-6, 4e-2]	log
discount factor	0.995	fixed
target network lag	0.01	fixed
batch size	128	fixed
clipping on gradients of Q	4.0	fixed
num gradient updates per observation	1	fixed
Huber loss clipping	1.0	fixed

Table 1. Random hyperparameter search procedure. We also include the hyperparameters which we kept fixed.

D. Implementation Details

We utilize feed forward networks for both policy and Q-value approximator. For $\mu_\theta(s)$ we use two hidden layers of dimensions $(400, 300)$ and relu activation functions. For $\tilde{Q}^\pi_w(s, a)$ and $Q^\pi_w(s, a)$ we first embed the state into a 400 dimensional vector using a fully-connected layer and $\tanh$ non-linearity. We then concatenate the embedded state with a and pass the result through a 1-hidden layer neural network of dimension 300 with $\tanh$ activations. We use a diagonal $\Sigma_\phi(s) = e^\phi$ for Smoothie, with ϕ initialized to -1.

To find optimal hyperparameters we perform a 100-trial random search over the hyperparameters specified in Table 1. The OU exploration parameters only apply to DDPG. The λ coefficient on KL-penalty only applies to Smoothie with a KL-penalty.

D.1. Fast Computation of Gradients and Hessians

The Smoothie algorithm relies on the computation of the gradients $\frac{\partial \tilde{Q}^\pi_w(s,a)}{\partial a}$ and Hessians $\frac{\partial^2 \tilde{Q}^\pi_w(s,a)}{\partial a^2}$. In general, these quantities may be computed through multiple backward passes of a computation graph. However, for faster training, in our implementation we take advantage of a more efficient computation. We make use of the following identities:

$$\frac{\partial}{\partial x} f(g(x)) = f'(g(x)) \frac{\partial}{\partial x} g(x), \tag{10}$$

$$\frac{\partial^2}{\partial x^2} f(g(x)) = \left(\frac{\partial}{\partial x} g(x)\right)^T f''(g(x)) \frac{\partial}{\partial x} g(x) + f'(g(x)) \frac{\partial^2}{\partial x^2} g(x). \tag{11}$$

Thus, during the forward computation of our critic network $\tilde{Q}^\pi_w$, we not only maintain the tensor output O_L of layer L, but also the tensor G_L corresponding to the gradients of O_L with respect to input actions and the tensor H_L corresponding to the Hessians of O_L with respect to input actions. At each layer we may compute $O_{L+1}, G_{L+1}, H_{L+1}$ given O_L, G_L, H_L. Moreover, since we utilize feed-forward fully-connected layers, the computation of $O_{L+1}, G_{L+1}, H_{L+1}$ may be computed using fast tensor products.

References

Price, R. A useful theorem for nonlinear devices having gaussian inputs. *IRE Transactions on Information Theory*, 4(2): 69–72, 1958.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *ICML*, 2014.