Stochastic Gradient Estimation With Finite Differences

Lars Buesing, Theophane Weber, Shakir Mohamed
DeepMind
{lbuesing, theophane, shakir}@google.com

Abstract

Choosing the parameters of a probability distribution in order to minimize an expected loss is the central problem in many machine learning applications, including inference in latent variable models or policy search in reinforcement learning. In most cases however, the exact gradient of the expected loss is not available as a closed-form expression. This has led to the development of flexible gradient estimators, with a focus on the conflicting objectives of being both general-purpose and low-variance. If the loss is non-differentiable, as is the case in reinforcement learning, or if the distribution is discrete, as for probabilistic models with discrete latent variables, we have to resort to score-function (SF) gradient estimators. Naive SF estimators have high variance and therefore require sophisticated variance reduction techniques, such as baseline models, to render them effective in practice. Here we show that under certain symmetry and parametric assumptions on the distribution, one can derive unbiased stochastic gradient estimators based on finite differences (FD) of the loss function. These estimators do not require learning baseline models and potentially have less variance. Furthermore, we highlight connections of the FD estimators to simultaneous perturbation sensitivity analysis (SPSA), as well as weak derivative and "straight-through" gradient estimators.

1 Introduction

Numerous problems in engineering, operations research, and machine learning naturally take the form of minimizing an expected loss:

$$\min_\theta \quad \mathbb{E}_{x\sim P_\theta}[H(x)]. \tag{1}$$

Here we allow parameters to stochastically influence the loss function H through the random variable x[1]. Often, one can neither compute exact expectations of H wrt. P_θ, nor its gradients $\partial_\theta \mathbb{E}[H]$; in this case one has to rely on simulation-based / stochastic optimization by evaluating the loss on samples of x [10]. Nevertheless, one can still apply approximate gradient-based optimization techniques using the so-called score-function (SF) estimator. [4]. SF estimators are central to policy-gradient methods in RL [11] and have also been applied to learning probabilistic models with discrete latent variables [5].

As the SF estimator applies to very general optimization problems, it is not surprising that in practice it is dogged by high variance, necessitating intricate variance reduction techniques. Here we show, that if we are willing to assume additional properties about how the parameters influence the distribution P_θ, we can make use of this additional information to arrive at a estimator which leverages more structure of the problem. More specifically, we assume that the density of P_θ is an *even* function wrt. x and the parameters we are optimizing are "location-scale" parameters. If, furthermore, it is possible to evaluate the loss function multiple times on the same input, we show that the standard SF estimator can be replaced by an unbiased estimator based on finite-differences (FD) of the *loss* function. This

[1]For ease of notation we suppress the deterministic dependence of $H(x,\theta)$ and simply write $H(x)$. The results extend to the general case.

Submitted to NIPS 2016 workshop "Advances in Approximate Bayesian Inference". Do not distribute.

estimator uses approximate first and second order derivative information and therefore potentially renders the optimization much more tractable. The requirement of multiple evaluations of H excludes some general problems (such as general RL problems), but includes (potentially challenging) special cases such as bandit problems.

2 Finite difference estimators

2.1 Score-function estimator

For optimizing the criterion 1 by gradient-descent methods, we can use an estimator g_θ that approximates the gradient of the cost in expectation $\partial_\theta \mathbb{E}[H] \approx \mathbb{E}[g_\theta]$. A very general, unbiased gradient estimator is the score-function (SF) estimator $g_{\theta,\mathrm{SF}}$:

$$g_{\theta,\mathrm{SF}}(x,\theta) \quad = \quad H(x)\partial_\theta \log f_\theta(x),$$

where f_θ is the probability density of P_θ.

2.2 Symmetric location-scale distributions

Assume that P_θ for $\theta = (\mu, \sigma) \in \mathbb{R}^d \times \mathbb{R}_+^d$ is a "location-scale" family of distributions with mean parameter μ and scale parameter σ and independent components:

$$f_\theta(x) \quad = \quad \prod_{i=1}^{d} \frac{1}{\sigma_i} f_i\left(\frac{x_i - \mu_i}{\sigma_i}\right).$$

Here, f_i are the densities which we require to be *even* functions:

$$f_i(-\epsilon_i) \quad = \quad f_i(\epsilon_i).$$

Under these assumptions we can express the gradient of the expected loss wrt. the location parameter μ in the following way:

$$\begin{aligned}
\partial_{\mu_i}\mathbb{E}[H] &= \mathbb{E}[H(x)\partial_{\mu_i}\log f_\theta(x)] \\
&= -\sigma_i^{-1}\mathbb{E}_{\epsilon\sim P_{(\mathbf{0},\mathbf{1})}}\left[H(\mu+\sigma\epsilon)s_i(\epsilon_i)\right] \\
&= -\frac{1}{2}\sigma_i^{-1}E_{\epsilon\sim P_{(\mathbf{0},\mathbf{1})}}\left[\left(H(\mu+\sigma\epsilon) - H(\mu-\sigma\epsilon)\right)s_i(\epsilon_i)\right],
\end{aligned}$$

where s_i is the score function:

$$s_i(\epsilon_i) \quad := \quad \frac{f_i'(\epsilon_i)}{f_i(\epsilon_i)}.$$

In the derivation we critically used the fact that the score function s_i is *odd* for an even density f_i, i.e. $s_i(-\epsilon_i) = -s_i(\epsilon_i)$. A similar reasoning can be applied for deriving the gradients wrt. the scale parameter σ, leading to the gradient estimators:

$$g_{\mu,\mathrm{FD}}(\epsilon,\theta) \quad = \quad -\frac{1}{2}\sigma^{-1}s(\epsilon)\left(H(\mu+\sigma\epsilon) - H(\mu-\sigma\epsilon)\right) \tag{2}$$

$$g_{\sigma,\mathrm{FD}}(\epsilon,\theta) \quad = \quad -\frac{1}{2}\sigma^{-1}\left(s(\epsilon)\epsilon + \mathbf{1}\right)\left(H(\mu+\sigma\epsilon) - 2H(\mu) + H(\mu-\sigma\epsilon)\right) \tag{3}$$

$$\epsilon \quad \sim \quad P_{(\mathbf{0},\mathbf{1})}, \tag{4}$$

where $\mathbf{0}, \mathbf{1}$ are vectors of 0s and 1s of appropriate size, $\epsilon := (\epsilon_1, \ldots, \epsilon_d)^\top$ (s is defined analogously), and σ^{-1} as well as $s(\epsilon)\epsilon$ are to be interpreted elementwise. We call the above quantities finite difference (FD) estimators, as they involve finite differences of first and second order of the cost H for mean and scale parameters respectively. It is straight-forward to show that they are unbiased. Furthermore, they are valid estimators even for non-differentiable cost functions H.

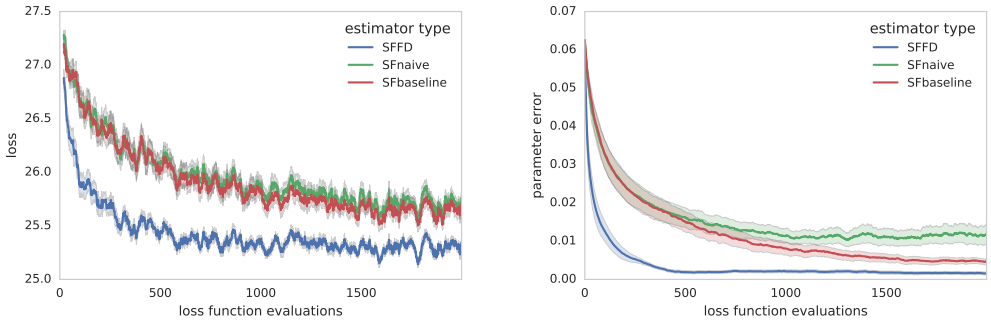

Figure 1: Finite difference gradient estimators (SFFD) outperform naive score function estimators with / without baseline (SFbaseline / SFnaive) for parameter estimation of Lotka-Volterra hyper-parameters from noisy observations, both in terms of approximate likelihood (negative loss) and parameter error (L2 norm $\|x_{\text{true}} - \bar{x}\|$ with true parameters x_{true} and posterior mean $\bar{x}$).

3 Implementation and experiments

Surrogate loss Software packages for automatic differentiation have proven extremely useful in modern machine learning for learning parameters of large models [2, 1]. In order to use FD estimators within automatic differentiation engines, we define a surrogate loss function $\tilde{L}$ [8]. Differentiation of $\tilde{L}$ wrt. θ yields the FD estimators defined above:

$$\tilde{L} = \mathbb{E}_{x\sim P_{\bar{\theta}}}[H(x,\phi)] + \mu^\top \mathbb{E}[g_{\mu,\text{FD}}(\epsilon,\bar{\mu})] + \sigma^\top \mathbb{E}[g_{\sigma,\text{FD}}(\epsilon,\bar{\sigma})].$$

Here, we set $\bar{\theta} := (\bar{\mu},\bar{\sigma}) := \theta$ during the forward pass (ie. function evaluation), but we do not back-propagate gradients wrt. $\bar{\theta}$ in the backward-pass. We also allow for the loss H to depend on additional parameters ϕ (which can also be functions of θ); if H is differentiable wrt. parameters ϕ, they can also be learned by gradient descent methods.

Lotka-Volterra ODEs Consider the Lotka-Volterra (LV) system of ODEs for modeling population sizes a, b of the prey / predator species:

$$\frac{da}{dt} = \alpha a - \beta ab, \qquad \frac{db}{dt} = \delta ab - \gamma b.$$

Assume we are given a set of N noisy observations $(o^n_{0:T}), n = 1,\ldots,N$ with $o^n_t := (a^n_t, b^n_t)$ consisting of trajectories of length T and initial conditions o^n_0. We wish to approximate the posterior distribution over the model parameters $\tilde{x} = (\alpha,\beta,\gamma,\delta)$ with a log-normal distribution q, ie. $\tilde{x} = \exp(x)$ and $q(x) = \mathcal{N}(x|\mu,\sigma)$. We do so by optimizing the variational lower bound on the marginal likelihood:

$$L(\mu,\sigma) \quad := \quad \mathbb{E}_{x\sim q}\left[\frac{1}{2}\sum_{n=1}^{N}\|o^n_{1:T} - \hat{o}(o^n_0,x)\|^2 + \log p(x) - \log q(x)\right],$$

where $\hat{o}(o^n_0,x)$ is the output of an LV simulator with initial conditions o^n_0 and parameters $\exp(x)$. Comparing the results of optimizing L using different gradient estimators shown in fig. 1, we can see that FD estimators outperform naive SF and SF with baseline estimators. Methods were compared based on the same number of function calls to the LV simulator.

4 Discussion

Antithetic sampling. The FD estimators can be interpreted as applying the standard variance reduction technique of antithetic sampling to the SF estimator [7]. In antithetic sampling (assuming an even density $f(\theta)$) one uses samples ϵ and their negation $-\epsilon$ to estimate expected values. This makes use of the fact that the expectation of an odd function under an even distribution is 0. The FD estimator $s(\epsilon)\,(H(\mu+\sigma\epsilon) - H(\mu-\sigma\epsilon))$ is exactly the even part of the SF estimator. We therefore expect the former to have lower variance compared to the latter if H is roughly an odd function around μ.

Perturbation Analysis. Evaluating the FD estimators requires three function calls to H with inputs $\mu + \sigma\epsilon$, μ and $\mu - \sigma\epsilon$; hence they are roughly three times as expensive as SF estimators. However, in contrast to the latter, they do not require a learnable baseline model, which is essential to reduce variance of regular SF estimators in practice. The above FD estimators exhibit an interesting parallel to SPSA [9]: a naive finite difference estimator for the d-dimensional gradient would require $2d$ evaluations of the form $H(\mu \pm \Delta_i e_i)$, where e_i is the i-th basis vector. SPSA is based on a simultaneous perturbation of all coordinates, i.e. $H(\mu + \Delta) - H(\mu - \Delta)$; empirically this leads to a d-fold speedup over the naive FD implementation. The FD gradient estimators 2 and 3 are also based on simultaneous perturbations and we therefore expect them to inherit the speedup of SPSA compared to independent perturbations. In contrast to SPSA, the statistics of the perturbations do not have to be chosen, but are given by the current estimate of the scale parameters σ.

Straight-through estimators. Let us assume that H is differentiable and that values of $\sigma_i\epsilon_i$ are typically small. By definition of the gradient, we have

$$H(\mu + \sigma\epsilon) - H(\mu - \sigma\epsilon) \approx 2 \sum_i \sigma_i \epsilon_i \partial_{\mu_i} H(\mu),$$

which turns the FD estimator into

$$\mathbb{E}[g_{\mu_i,\mathrm{FD}}] = -\partial_{\mu_i} H(\mu)\, \mathbb{E}[\epsilon_i \frac{f_i'(\epsilon_i)}{f_i(\epsilon_i)}] = \partial_{\mu_i} H(\mu),$$

which is the straight through estimator (where $E[\epsilon_i \frac{f_i'(\epsilon_i)}{f_i(\epsilon_i)}] = -1$ by integration by parts) introduced by [3].

Weak derivative estimators. The proposed FD estimators as intimately connected to the concept of weak derivatives (WD) of distributions [6]. Briefly, a 3-tuple $(c(\theta), \partial f^+, \partial f^-)$ consisting of a θ-dependent constant c and two probability distribution $\partial f^\pm$ is called a weak derivative of P_θ if:

$$\partial_\theta\, \mathbb{E}_{x\sim P_\theta}[H(x)] = c(\theta)\left(\mathbb{E}_{x\sim \partial f^+}[H(x)] - \mathbb{E}_{x\sim \partial f^-}[H(x)]\right).$$

For the one-dimensional case, making use of the assumptions that $f_\theta(x)$ is an even function of x, it is straight-forward to show that:

$$g_{\mu,\mathrm{WD}} \propto H(\mu + \Delta) - H(\mu - \Delta); \quad \Delta \sim \partial f^+.$$

The support of ∂f^+ is $\mathbb{R}^+$ and $\partial f^+(x) \propto s(x) f_0(x)$. So, in 1-d for a location parameter, we can interpret the FD estimator 2 approximating the WD estimator by importance sampling with a proposal density f_0 and the resulting importance weight $s(x)$. For the case of $d > 1$ the WD estimator is computationally expensive as it requires $2d$ evaluations of $H(\mu \pm \Delta_i e_i + \Delta_{-i})$, where $\Delta_i \sim \partial f^+$ and $(\Delta_{-i})_j \sim f$ for $j \neq i$ and $(\Delta_{-i})_i = 0$.

References

[1] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow. org*, 1, 2015.

[2] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.

[3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[4] Michael C Fu. Gradient estimation. *Handbooks in operations research and management science*, 13:575–616, 2006.

[5] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

[6] G Ch Pflug. Sampling derivatives of probabilities. *Computing*, 42(4):315–328, 1989.

[7] Brian D Ripley. *Stochastic simulation*, volume 316. John Wiley & Sons, 2009.

[8] John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pages 3528–3536, 2015.

[9] James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992.

[10] James C Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.

[11] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.