Gradient Estimator Summary

Zhexin(Ellery) Lai

University of Toronto

June 5, 2018

Motivation

In general, we want to able to estimate the gradient of

$$\mathbb{E}_{p(b|\theta)}[f(b)]$$

w.r.t. parameter θ.
Note that

$$\partial_\theta \mathbb{E}_{p(b|\theta)}[f(b)] = \partial_\theta \int_{-\infty}^{\infty} f(b)p(b|\theta)db \quad (1)$$

$$= \int_{-\infty}^{\infty} f(b)\partial_\theta p(b|\theta)db \quad (2)$$

The problem here is that equation (2) doesn't have the form of expectation. So we can't use simple Monte Carlo to estimate the expectation

Score Function

One way of dealing with this is that using Score Function

$$\int_{-\infty}^{\infty} f(b)\partial_\theta p(b|\theta)db \tag{3}$$

$$= \int_{-\infty}^{\infty} f(b)p(b|\theta)\partial_\theta log(p(b|\theta))db \tag{4}$$

So equation (4) can be seen as expectation of $f(b)\partial_\theta log(p(b|\theta))$ over distribution $p(b|\theta)$.

Score Function

$$\partial L = \int_{-\infty}^{\infty} f(b)p(b|\theta)\partial_\theta log(p(b|\theta))db \tag{5}$$

$$= \mathbb{E}_{p(b|\theta)]}[f(b)\partial_\theta log(p(b|\theta))] \tag{6}$$

$$\approx \frac{1}{S}\sum_{s}^{S} f(b^{(s)})\partial_\theta log(p(b^{(s)}|\theta)) \tag{7}$$

Where $b^{(b)}$ is sample from $p(b|\theta)$ based on current parameter θ

- Pros of this method is that it doesn't require function f to be differentiable as we never differentiate f and b can be either discrete or continuous as long as its pdf is differentiable
- However, this method to be known has high variance and unable to update parameter θ efficiently

Reparameterization

Alternatively, since our original problem arises from that θ parametrizes the expectation. So instead of differentiate θ, we differentiate a different parameters on which θ depends.

- sample $\epsilon \sim N(0,1)$
- choose a smooth function h and write $b = h(\epsilon, \theta)$
-

$$\begin{aligned}
\partial_\theta L &= \partial_\theta \mathbf{E}_{p(b|\theta)}[f(b)] && (8)\\
&= \partial_\theta \mathbf{E}_{\epsilon \sim p(\epsilon)}[f(g(\theta,\epsilon)] && (9)\\
&= \mathbf{E}_{\epsilon \sim N(0,1)}[\partial_\theta f(g(\theta,\epsilon)] && (10)\\
&= \mathbf{E}_{\epsilon \sim N(0,1)}[\partial_\theta f(g(\theta,\epsilon))\partial_\theta g(\theta,\epsilon)] && (11)\\
&\approx \frac{1}{S}\sum_{s}^{S} \partial_\theta f(g(\theta,\epsilon^{(s)}))\partial_\theta g(\theta,\epsilon^{(s)}) && (12)
\end{aligned}$$

Reparameterization

- This reparameterization trick gives us lower variances
- However, if we look at the term inside the sum. We notice that we need function f to be differentiable w.r.t. ϕ; and we need a differentiable function g to represent b.
- But what if f is not differentiable ?
- What if b is discrete ?

So we see that the Pros of Score Function becomes Cons in reparameterization and vice verse.

Goal

Therefore, we need something more powerful.

- More flexibility of f: non differentiable
- More flexibility of random variable b:can be either discrete or continuous
- Low variance of gradient estimator
- Gradient Estimator is unbiased

Initial Idea

We introduce a control variate function C to alleviate the high variance of score function

$$\mathbb{E}_{p(b|\theta)}[(f(b) - C)\partial_\theta log(p(b|\theta))] \tag{13}$$

Note that if C is a constant, then (13) estimator is unbiased, but if C is function depends on b, it's biased (Shown next slides)
So we need a correction of bias

$$\mathbb{E}_{p(b|\theta)}[(f(b) - C(b))\partial_\theta log(p(b|\theta))] + \partial_\theta \mathbb{E}_{p(b|\theta)}[C(b)] \tag{14}$$

Justification

Before we propose the next gradient estimator; let's justify why equation (14) make sense

- for any fix c, $\mathbb{E}_{p(b|\theta)}\big[(f(b)-c)\partial_\theta log(p(b|\theta))\big]$ is unbiased

$$\mathbb{E}_{p(b|\theta)}\big[(f(b)-c)\partial_\theta log(p(b|\theta))\big] \tag{15}$$

$$= \int p(b|\theta)(f(b)-c)\partial_\theta log(p(b|\theta)) \tag{16}$$

$$= \int p(b|\theta)(f(b)-c)\frac{1}{p(b|\theta)}\partial_\theta(p(b|\theta)) \tag{17}$$

$$= \int (f(b)-c)\partial_\theta(p(b|\theta)) \tag{18}$$

$$= \int f(b)\partial_\theta p(b|\theta) - \int c\partial_\theta p(b|\theta) \tag{19}$$

$$= \int f(b)\partial_\theta p(b|\theta) = \partial_\theta \mathbb{E}_{p(b|\theta)}[f(b)] \tag{20}$$

Note that $\int c\partial_\theta p(b|\theta) = c\partial_\theta \int p(b|\theta) = c\partial_\theta 1 = 0$. The integral is over the range b

Justification

Notice from above (equation (19)), that if c is not constant but a function of random variable b, above results might not hold (eg. if $c(b) = b$, then second term in equation (19) is $\partial_\theta \mathbb{E}_{p(b|\theta)}(b)$). Equality in (20) doesn't hold.
Continues from equation (19) with $C(b)$

$$\int f(b)\partial_\theta p(b|\theta) - \int c(b)\partial_\theta p(b|\theta) \tag{21}$$
$$= \partial_\theta \mathbb{E}_{p(b|\theta)}[f(b)] - \partial_\theta \mathbb{E}_{p(b|\theta)}[c(b)] \tag{22}$$

This justify why we need a correction term (which is simply add one more term to cancel the negative bias). So, Equation (14) is an unbiased estimator

Initial Idea

- Notice, that when $C = f$; equation(14) becomes exactly as regular reparameterization.
- Hence, C acts as a balance of variances between score function and reparameterization (Vanriance can be as low as reparameterization trick)
- Also, if $C \neq f$, no need to differentiate through f. We no longer requires f to be differentiable

Objective

- We want C to be differentiable w.r.t. θ
- We want to approximate C such that it makes the equation (14) with lowest variances
- Since one of our goal is to deal with discrete b; the equation (14) requries us to find a good reparameterization function h

High Level Procedure

- Find a differentiable function h for random variable b
- update θ that optimization the equation (14)
- update the parameters of function C such that minimized the variance of equation (14)

First Approach (Concrete)

Let's solve each of our goals step by step. First, we aim at dealing discrete b by find a good smooth function for re-parameter. Let $b \sim p(b|\phi)$ be a Bernoulli random variable for simple illustration

- sample $u \sim Unif(0, 1)$
- deterministically define $z = h(u, \theta)$ given u (e.g. $z = log(\frac{\theta}{1-\theta}) + log(\frac{u}{1-u})$)
- Now; we reparameterize $b = H(z) \approx \sigma_\lambda(z) = (1 + exp(-\frac{z}{\lambda}))^{-1}$ where $H(z) = 1$ if $z \geq 0$ and $H(z) = 0$ otherwise
  Hence,

$$\begin{aligned} \partial_\theta \mathbb{E}_{p(b|\theta)}[f(b)] &\approx \partial_\theta \mathbb{E}_{p(z|\theta)}[f(\sigma_\lambda(z))] && (23) \\ &= \partial_\theta \mathbb{E}_{p(u)}[f(\sigma_\lambda(h(u, \theta)))] && (24) \\ &= \mathbb{E}_{p(u)}[\partial_\theta f(\sigma_\lambda(h(u, \theta)))] && (25) \end{aligned}$$

- Now, we can estimate equation (25) with reparameterization trick

Weakness of Concrete

- We see that Concrete Relax overcome one major issues: discrete random variable
- However, notice that in this approach, chosen hyperparameter λ is hard
- If λ is large, σ_λ is more horizontally flat(streched), so lower variance for different values of z but it less accurate as an approximator of Bernoulli b
- As $\lambda \to 0$, more squashed, it approximate Bernoulli more accurately, but the variance goes to very high

REBAR approach

Rebar Estimator is an alternative that deal with this bias and variance trade-off by combine concrete relax with equation (14)
Notice from equation (23)

$$\partial_\theta \mathbb{E}_{p(z|\theta)} f(\sigma_\lambda(z)) = \mathbb{E}_{p(z|\theta)}[f(\sigma_\lambda(z))\partial_\theta log(p(z|\theta))] \tag{26}$$

Also, notice that in equation (14)

$$\begin{aligned}
\mathbb{E}_{p(b|\theta)}[f(b)\partial_\theta log(p(b))] &= \partial_\theta \mathbb{E}_{p(b|\theta)}[f(b)] && (27)\\
&= \partial_\theta \mathbb{E}_{p(z|\theta)}[f(H(z))] && (28)\\
&= \mathbb{E}_{p(z|\theta)}[f(H(z))\partial_\theta log(p(z|\theta))] && (29)
\end{aligned}$$

REBAR Estimator

Recall from our original gradient estimator proposal

$$\mathbb{E}_{p(b|\theta)}[(f(b) * \partial_\theta log(p(b|\theta)) - C(b) * \partial_\theta log(p(b|\theta)))] + \partial_\theta \mathbb{E}_{p(b|\theta)}[C(b)]$$

- We use equation (29) to approximate $f(b)$ ($H(z)$ is discontinuous here but it's Ok as we don't reparameterize it; we solve it with score function; then alleviate its variance)
- Now, we need to design a $C(b)$ Since we require $C(b)$ to be differentiable even with discrete input b. We use the trick from Concrete Estimator. Basically, use equation(26) as our control variate.

REBAR Approach

Note that equation (26) can be written as

$$\mathbb{E}_{p(z|\theta)}[f(\sigma_\lambda(z))\partial_\theta log(p(z|\theta))] \tag{30}$$

$$= \sum_z p(z)[f(\sigma_\lambda(z))\partial_\theta log(p(z|\theta))] \tag{31}$$

$$= \sum_z \sum_b \left[p(b,z)f(\sigma_\lambda(z))\partial_\theta log(p(z|\theta))\right] \tag{32}$$

$$= \sum_z \sum_b \left[p(z|b)p(b)f(\sigma_\lambda(z))\partial_\theta log(p(z|\theta))\right] \tag{33}$$

$$= \sum_b p(b) \sum_z \left[p(z|b)f(\sigma_\lambda(z))\partial_\theta log(p(z|\theta))\right] \tag{34}$$

$$= \sum_b p(b) \sum_z \left[p(z|b)f(\sigma_\lambda(z))\partial_\theta \left[log(p(z|b,\theta)) + log(p(b|\theta))\right]\right] \tag{35}$$

REBAR Estimator

$$= \mathbb{E}_{p(b)}\left[\partial_\theta \mathbb{E}_{p(z|b)}\left[f(\sigma_\lambda(z))\right]\right] + \mathbb{E}_{p(b)}\left[\mathbb{E}_{p(z|b)}\left[f(\sigma_\lambda(z))\partial_\theta log(p(b))\right]\right] \tag{36}$$

equation(35) to equation(36) are derived from the definition of expectation and log trick
For the gradient inside the first term; we can estimate it with reparameterization

$$\mathbb{E}_{p(b)}\left[\partial_\theta \mathbb{E}_{p(z|b)}\left[f(\sigma_\lambda(z))\right]\right] = \mathbb{E}_{p(b)}\left[\mathbb{E}_{p(v)}\left[\partial_\theta f(\sigma_\lambda(\tilde{z}))\right]\right] \tag{37}$$

where $v \sim Unif(0,1), \tilde{z} = \tilde{h}(v, b, \theta)$;

while the second term can be estimated as follows

$$\mathbb{E}_{p(b)}\left[\mathbb{E}_{p(z|b)}\left[f(\sigma_\lambda(z))\partial_\theta log(p(b))\right]\right] \tag{38}$$

$$= \sum_b p(b) \sum_z p(z|b)\left[f(\sigma_\lambda(z))\partial_\theta log(p(b))\right] \tag{39}$$

$$= \sum_b p(b) \sum_z p(z|b)\left[f(\sigma_\lambda(z))\frac{1}{p(b)}\partial_\theta p(b)\right] \tag{40}$$

$$= \sum_b p(b)\frac{1}{p(b)} \sum_z p(z|b)\left[f(\sigma_\lambda(z))\partial_\theta p(b)\right] \tag{41}$$

$$= \sum_b \sum_z p(z|b)\left[f(\sigma_\lambda(z))\partial_\theta p(b)\right] \tag{42}$$

$$= \mathbb{E}_{p(b)}\left[\mathbb{E}_{p(z|b)}[f(\sigma_\lambda(z))\partial_\theta log(p(b|\theta))]\right] \tag{43}$$

$$= \mathbb{E}_{p(b)}\left[\mathbb{E}_{p(v)}[f(\sigma_\lambda(\tilde{z}))\partial_\theta log(p(b|\theta))]\right] \tag{44}$$

REBAR Estimator

Now, we can combined things together, plug equation (44),(37) into (36); then plug (36) into (14) (to replace $C(b)$); finally replace f with (28) Note that our reparameterization make b deterministically depends on u and v from Uniform distribution.

$$\mathbb{E}_{p(b)}\left[\mathbb{E}_{p(z|b)}(F)\right] = \sum_b \sum_z p(z|b)p(b)(F) \tag{45}$$

$$= \sum_b \sum_z p(z,b)(F) = \mathbb{E}_{p(u,v)}[F] \tag{46}$$

Our gradient estimator (originally equation (21))

$$\hat{g}_\theta = \mathbb{E}_{p(u,v)}\left[\Big(f(H(z)) - f(\sigma_\lambda(\tilde{z}))\Big)\partial_\theta log(p(b)) + \partial_\theta f(\sigma_\lambda(z)) - \partial_\theta f(\sigma_\lambda(\tilde{z}))\right] \tag{47}$$

- Instead of having λ to be hyperparameters, REBAR update λ with objection that reduce the variances

Varaince Control Objective Function

$$Var(\hat{g}_\theta) = \mathbb{E}[\hat{g}_\theta^2] - (\mathbb{E}[\hat{g}_\theta])^2] \tag{48}$$

Note that due to its unbiased property

$$\mathbb{E}[\hat{g}_\theta] = f(H(z)) \implies \partial_\lambda f(H(z)) = 0$$

Our gradient estimator for varaince is

$$\begin{aligned} \hat{g}_\lambda &= \partial_\lambda(\mathbb{E}[\hat{g}_\theta^2] - (\mathbb{E}[\hat{g}_\theta])^2) && (49) \\ &= \partial_\lambda \mathbb{E}[\hat{g}_\theta^2] && (50) \end{aligned}$$

Update rule for λ

$$\lambda \leftarrow \lambda - \alpha * \hat{g}_\lambda$$

REBAR Estimator

- We see how Concrete approach dealing with discrete random variable
- REBAR, based on Concrete, reduce the variances (adding control variate)
- However, notice that equation(47) requires f to be differentiable
- The only parameter for variance control is λ

Justification of reLAX

- Note that with correctio of bias term, the estimator is unbias and holds for any choice of C; and by Universal approximation theorem; we can represent C with a neural network C_ϕ where ϕ is the weight and bias of neural net.
- Instead of update the parameter λ in REBAR, we update the parameter ϕ (λ is incorporated into neural network as weights)
- Even if f is undifferentiable, we backpropagation on neural network w.r.t. its weight ϕ to control the variance. So more flexible of function f
- So in equation (47), we can replace f with C_ϕ to get our final version of graident estimator

Justification for reLAX

More specific,

$$\hat{g}_{\text{reLAX}} = \mathbb{E}_{p(u,v)}\left[\left[f(b) - [C_\phi(\tilde{z})]\right]\partial_\theta logp(b|\theta) - \partial_\theta[C_\phi(\tilde{z})]\right] + \partial_\theta[C_\phi(z)] \tag{51}$$

Algorithm

Let's summary all things together

Algorithm 1: reLAX

input : differentiable pdf $p(b|\theta)$, a function $f(\cdot)$, neural network C_ϕ, step size α_1, α_2,reparameterization distribution q, hard threshold function $H(\cdot)$, smooth function $h(u, \theta)$, $\tilde{h}(v, H(z), \theta)$

while *not converge* **do**

$u, v \sim q$; `// sample reparam random variable`

$z_i \leftarrow h(u, \theta)$; `// Input for Concrete Relax`

$\tilde{z}_i \leftarrow \tilde{h}(v, H(z_i), \theta)$; `// Input for conditional reparam on` $H(z)$

$\hat{g}_\theta \leftarrow$ equation (50)

$\hat{g}_\phi \leftarrow \partial_\phi Var(\hat{g}_\theta)$

$\theta \leftarrow \theta - \alpha_1 * \hat{g}_\theta$; `// update the distribution parameter`

$\phi \leftarrow \phi - \alpha_2 * \hat{g}_\phi$; `// update control variate neural network weight`

end

Application

In Stochastic Variational Inference, we are optimizing our objective function (minimized the negative ELBO)

$$L = -\mathbb{E}_{q(z|x,\theta)}[log(p(x|z) + log(p(z)) - log(q(z|x,\theta)))]$$

by performing gradient descent to update the variational parameters

$$\theta^t \leftarrow \theta^{(t-1)} - \alpha * \partial_\theta L$$

But compute $\partial_\theta L$ can be hard as most of case, it unlikely for it to be computed analytically.
Therefore, we can use the gradient estimator reLAX proposed to estimate $\partial_\theta L$

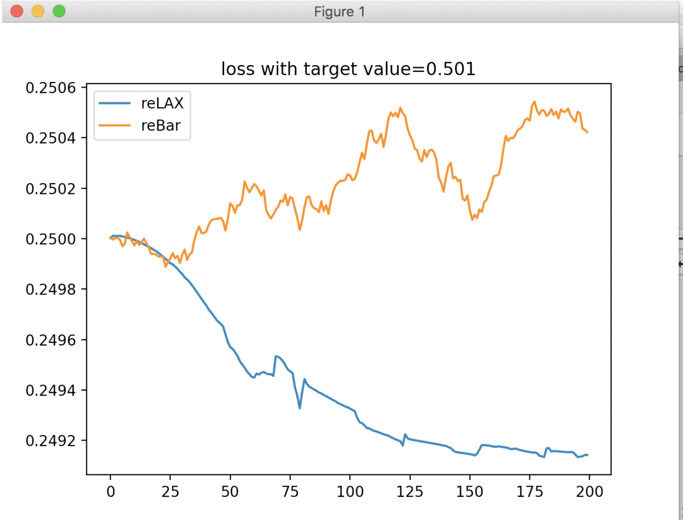
Experiment

- Consider $b \sim Bernoulli(\theta)$.
- objective to minimize $\mathbb{E}_b[(0.501 - b)^2]$
- Challenges: (1) Discrete Random Variable. (2) Very small loss value even for completely wrong move (we could have very high variance for each update) (3) We make the task even more challenge, each iteration of gradient estimator, we only sample one point

Let's compare reLAX estimator vs reBAR estimator

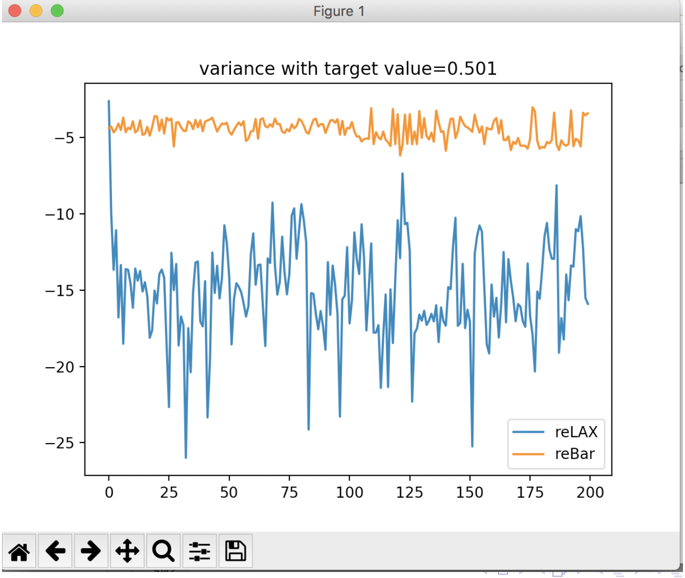
Experiment

We see that under challenge case, reBar fails for the task over 10000 iterations.
While reLAX successfully complete the task



Experiment

For the log variance, reLAX is always below reBar



Bibliography

- George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. REBAR: Low-variance, unbiased gradient estimates for discrete latent variable models. In Advances in Neural Information Processing Systems, pages 26242633, 2017.
- Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. arXiv:1711.00123, 2017.