BACKPROPAGATING THROUGH ANYTHING BY OPTIMIZING DIFFERENTIABLE CONTROL VARIATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Gradient-based optimization is the foundation of deep learning and reinforcement learning. Even when the mechanism being optimized is unknown or not differentiable, optimization using high-variance or biased gradient estimates is still often the best strategy. We introduce a general framework for learning low-variance, unbiased gradient estimators for black-box functions. These estimators can be jointly trained with model parameters or policies, and are applicable in both discrete and continuous settings. We give unbiased, adaptive analogs of state-of-the-art reinforcement learning methods such as deep deterministic policy gradients and advantage actor-critic. We also demonstrate this framework for training discrete latent-variable models.

1 INTRODUCTION

Gradient-based optimization has been key to most recent advances in machine learning and deep reinforcement learning. The back-propagation algorithm (Rumelhart & Hinton, 1986), also known as reverse-mode automatic differentiation (Speelpenning, 1980; Rall, 1981) computes exact gradients of deterministic, differentiable objective functions. The reparameterization trick (Kingma & Welling, 2014; Rezende et al., 2014) enables backpropagation to give unbiased, low-variance estimates of gradients of expectations on continuous random variables. This has allowed effective stochastic optimization of large probabilistic latent-variable models.

Unfortunately, backpropagation cannot be straightforwardly applied to problems involving discrete random variables, or when the function being optimized is a black box (Schulman et al., 2015). This is the case in most reinforcement learning settings, or when fitting probabilistic models with discrete latent variables. Much recent work has been devoted to constructing gradient estimators for these situations. In reinforcement learning, advantage actor-critic methods (Mnih et al., 2016) and deep deterministic policy gradients (Lillicrap et al., 2015) give low-variance but biased gradient estimates by jointly optimizing policy parameters together with baselines. In discrete latent-variable models, low-variance but biased gradient estimates can be given by continuous relaxations of discrete variables (Maddison et al., 2016; Jang et al., 2016).

A recent advance by Tucker et al. (2017) used low-variance but biased baselines given by continuous relaxations to construct unbiased, low-variance gradient estimates. Furthermore, Tucker et al. (2017) showed how to tune the free parameters of these relaxations to minimize their variance during training.

In this paper, we simplify and generalize the method of Tucker et al. (2017) to learn a free-form control variate parameterized by a neural network, giving lower-variance, unbiased gradient estimates. Importantly, our method is applicable even when no continuous relaxation is available, as in reinforcement learning. We derive improved variants of popular reinforcement learning methods with unbiased gradients and more stable training dynamics.

2 BACKGROUND: GRADIENT ESTIMATORS

For simplicity of exposition, consider a simple stochastic model that has a discrete latent variable b with probability $p_\theta(b)$ and loss function $f(b)$.

Note we assume without loss of generality that $f(b)$ is independent of the parameters θ of $p_\theta(b)$. Training the model involves minimizing the expected cost $\mathcal{L}(\theta) = \mathbb{E}_{p_\theta(b)}[f(b)]$. This can be achieved efficiently using gradient optimization, requiring exact or approximate computation of $g = \partial\mathcal{L}/\partial\theta$.

In many cases of interest, an analytic form of the gradient g is not known, and so an estimator $\hat{g}$ is required, such as the Monte Carlo estimator $\hat{g} \approx \sum_{i=1}^{n} \hat{g}(b_i)/n$ where $b_i \sim p_\theta(b)$ and n is the number of samples.

2.1 REINFORCE

The REINFORCE gradient estimator (Williams, 1992) expands g as $\mathbb{E}_{p_\theta(b)}[f(b)\frac{\partial}{\partial\theta}\log p_\theta(b)]$, and uses exactly a Monte Carlo estimation scheme. However, the derivative of the log-likelihood w.r.t. its parameters (known as the score function in statistics) has high variance under Monte Carlo estimation, and so variance-reduction techniques such as control variates or Rao-Blackwellization are usually applied to improve the speed of convergence to a good solution.

2.2 Concrete relaxation

Another approach to estimating gradients through discrete random variables is a continuous relaxation. Maddison et al. (2016) and Jang et al. (2016) developed a differentiable relaxation of the categorical distribution using the Gumbel-Max trick for sampling from a discrete distribution.

2.3 REBAR estimator

The REBAR gradient estimator develops a lower-variance gradient estimator that outperforms one based on a Concrete relaxation. REBAR relies on a carefully designed control variate, so we begin with a review of this theory.

2.3.1 Control Variates for Variance Reduction

For an unbiased estimator $f(b)$, a control variate is a function $\tilde{f}$ with a known or estimatable mean $\mathbb{E}[\tilde{f}]$. Since the mean of the function can be subtracted from the expectation, $f(b) - \eta(\tilde{f} - \mathbb{E}[\tilde{f}])$ remains an unbiased estimator. A control variate is scaled by a constant η. Note that we can write $\text{Var}(f(b) + \eta(\tilde{f} - \mathbb{E}[\tilde{f}]))$ as

$$\text{Var}(f(b) + \eta\tilde{f}) = \text{Var}(X) + \eta^2\text{Var}(\tilde{f}) + 2\eta\text{Cov}(f(b), \tilde{f}), \tag{1}$$

which, evaluating the first derivative w.r.t. η and solving for zero yields:

$$\eta = -\frac{Cov(f(b), \tilde{f})}{Var(\tilde{f})}. \tag{2}$$

The variance reduction effect of a control variate is induced by the high correlation of the control variate with the original estimator. The intuition behind this is as follows. When $f(b)$ and $\tilde{f}$ are positively correlated, then this means $\tilde(f)$ is large when $f(b)$ is large. So, if in some minibatch $\tilde{f}$ is greater than its known mean, then with high probability $f(b)$ is also greater than its true mean. That means this minibatch would contribute variance to the overall estimation algorithm, since the values we're estimating of $f(b)$ are with high probability greater than the true mean (the true gradient, in the case of REBAR). With positive correlation, $\text{Cov}(f(b), \tilde{f})) > 0$, and so η is negative. Then, the effect of the control variate in such a minibatch is to reduce the REINFORCE estimate by subtracting the quantity $\eta(\tilde{f} - \mathbb{E}[\tilde{f}]$.

2.3.2 Reducing gradient variance through control variates

The core of the REBAR estimator is a REINFORCE-style estimate of a non-differentiable reparameterization of the discrete latent variable as $b = H(z)$, where H is the hard-threshold function an

$$z := g(u, \theta) := \log\frac{\theta}{1-\theta} + \log\frac{u}{1-u}, u \sim \text{Unif}[0, 1] \tag{3}$$

While z is a reparameterization that renders the parameters of b learnable by gradient-based methods, H introduces a new discontinuity in the loss. Instead of relaxing the hard threshold as in Maddison et al. (2016), Tucker et al. (2017) uses a REINFORCE estimator for the gradient reparameterized with $H(z)$:

$$\frac{\partial}{\partial\theta}\mathbb{E}_{p(b)}[f(b)] = \frac{\partial}{\partial\theta}\mathbb{E}_{p(u)}[f(H(z))] = \mathbb{E}_{p(u)}[f(H(z))\frac{\partial}{\partial\theta}\log p(z)] \tag{4}$$

This allows the parameters θ to be learned using gradient information, but the loss is still non-differentiable due to the hard threshold. Since this uses a REINFORCE estimator, it also has high variance.

Hence, Tucker et al. (2017) develop a control variate. A natural continuous relaxation of the hard threshold function is the sigmoid function, leading Tucker et al. (2017) to choose a $H(z) \approx \sigma_\lambda(z) := \sigma(z/\lambda) = (1+\exp(-z/\lambda))^{-1}$. This relaxation leads to the following control variate summed with its expectation:

$$\frac{\partial}{\partial\theta}\mathbb{E}_{p(z)}[f(\sigma_\lambda(z))] = \mathbb{E}_{p(z)}[f(\sigma_\lambda(z))\frac{\partial}{\partial\theta}\log p(z)]. \tag{5}$$

Unfortunately, simply applying this control variate was found to be ineffective. The author's key insight was to derive a low-variance form of this control variate that takes advantage of a conditional marginalization of the reparameterized z given a particular choice of discrete b. This introduces a second reparameterization $\tilde{z}$ of $p(z|b)$, which depends on another sample $v \sim \text{Unif}[0,1]$. See Tucker et al. (2017) for details of the derivation.

The control variate has the following form:

$$f(\sigma_\lambda(\tilde{z}))\frac{\partial}{\partial\theta}\log p(H(z)) \tag{6}$$

and noting that

$$\mathbb{E}_{p(u,v)}[f(\sigma_\lambda(\tilde{z}))\frac{\partial}{\partial\theta}\log p(H(z))] = \mathbb{E}_{p(u,v)}[\frac{\partial}{\partial\theta}f(\sigma_\lambda(\tilde{z})) - \frac{\partial}{\partial\theta}f(\sigma_\lambda(z))] \tag{7}$$

gives us the REBAR gradient estimator:

$$\frac{\partial}{\partial\theta}\mathbb{E}_{p(b)}[f(b)] = \mathbb{E}_{p(u,v)}[f(\sigma_\lambda(\tilde{z}))\frac{\partial}{\partial\theta}\log p(H(z)) - \eta f(\sigma_\lambda(\tilde{z}))\frac{\partial}{\partial\theta}\log p(H(z)) + \eta\frac{\partial}{\partial\theta}f(\sigma_\lambda(z)) - \eta\frac{\partial}{\partial\theta}f(\sigma_\lambda(\tilde{z}))] \tag{8}$$

where η is trained to minimize the variance of the estimator.

The special form of $\tilde{z}$ yields a lower-variance gradient estimate because a number of the random variables are conditionally marginalized out of the estimator. Two features of this control variate make it particularly effective: its high correlation with the REINFORCE gradient, and a low-variance, reparameterized form of certain terms in the estimator.

3 RELAX: THE GENERALIZED REBAR ESTIMATOR

The REBAR estimator uses a control variate that evaluates the original loss function at relaxed inputs, reparameterized both unconditionally (denoted z, and conditionally, denoted $\tilde{z}$). The central result of this paper is that learning the function in the control variate leads to even better convergence properties. Specifically, we generalize the conditional marginalization and control variate of REBAR to the following form:

$$\mathbb{E}_{p(u,v)}[r(\tilde{z};\phi)\frac{\partial}{\partial\theta}\log p(H(z))] = \mathbb{E}_{p(u,v)}[\frac{\partial}{\partial\theta}r(\tilde{z};\phi) - \frac{\partial}{\partial\theta}r(z;\phi)], \tag{9}$$

where r is a neural network with parameters ϕ.

The generalized REBAR estimator replaces the loss function evaluations in the control variate with an adaptive r function which is trained via gradient decent to minimize the variance of the estimator.

As shown in Tucker et al. (2017), this can be easily computed. Denoting the RELAX estimator as $r(\phi)$ we obtain:

$$\frac{\partial}{\partial\phi}\text{Var}(r(\phi)) = \frac{\partial}{\partial\phi}\mathbb{E}[r(\phi)^2] + \frac{\partial}{\partial\phi}\mathbb{E}[r(\phi)]^2 = \frac{\partial}{\partial\phi}\mathbb{E}[r(\phi)^2] = \mathbb{E}[\frac{\partial}{\partial\phi}r(\phi)^2] \tag{10}$$

Where the second equality comes from the fact that for all ϕ, the RELAX estimator is unbiased and therefore $\frac{\partial}{\partial\phi}\mathbb{E}[r(\phi)]^2 = 0$.

In Tucker et al. (2017) the concrete distribution Maddison et al. (2016) is used in the control variate due to its similarity to the Bernoulli distribution and assumed correlation with the target function (**TODO EXPAND THIS).

NEXT: PROVE THAT CONCRETE IS NOT THE OPTIMAL RELAXATION TO USE (SHOULD BE EASY TO PROVE BUT WE NEED TO DO IT)

Relaxing Assumptions

identifying assumptions other estimators in the past made

demonstrate (proof?) that those assumptions are not optimal

and that the optimum is a function of f, theta

4 Scope and Limitations

One major limitation of the REBAR estimator and the concrete relaxation is that they requires the function being optimized, whose input is only defined at discrete inputs, to also accept continuous inputs, and to be differentiable w.r.t. those inputs. This makes REBAR and the concrete relaxation inapplicable for optimizing black-box functions, as in reinforcement learning settings where the environment is unknown.

Following Tucker et al. (2017), the following overview focuses on a single discrete Bernoulli random variable. However, the generalization to categorial variables is straightforward.

5 Optimizing continuous black-box functions

Deep deterministic policy gradients (Lillicrap et al., 2015)

Also: (Levine et al., 2016)

6 Experiments

We demonstrate the effectiveness of our estimator on a number of challenging optimization problems. Following Tucker et al. (2017) we begin with a simple toy example to illuminate the potential of our method and then continue to the more relevant problems of optimize binary VAE's and reinforcement learning (OR GRAPH STRUCTURE LEARNING).

NEXT: OPTIMAL RELAXATION FOR TOY PROBLEM NEXT: ELABORATE ON DEPENDENCE ON THETA

**TODO NEW EXPERIMENT: HALFWAY THROUGH A VAE TRAINING STOP, AND OPTIMIZE THE VARINACE OF r AND OF REBAR WRT TEMP AND ETA AND COMPARE GRADIENTS

6.1 Toy Experiment

We seek to minimize $\mathbb{E}_{b\sim p(b|\theta)}[(b-t)^2]$ as a function of the parameter θ where $p(b|\theta) = \texttt{Bern}(b|\theta)$. Tucker et al. (2017) set the target $t = .45$. Here, we focus on the more challenging case where $t = .499$. With this setting of the target, REBAR and competing methods suffer from high variance and are unable to discover the true solution of $\theta = 0$.

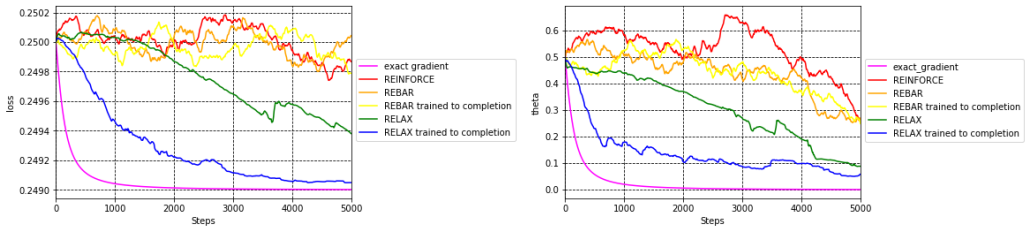

Figure 1: Estimator loss (left) and estimated θ values (right). Estimators with fixed (or no relaxation) struggle to converge to the optimal value

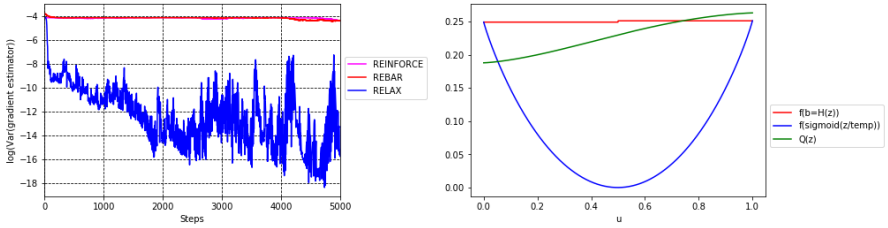

Figure 2: Left: Estimator variance. Right: Learned relaxation function for fixed θ

The fixed Concrete relaxation of REBAR is unable to produce a gradient who's signal outweighs the sample noise and is therefore unable to solve this problem noticeably faster than REINFORCE. Figure 6.1 plots the learned relaxations for a fixed value of θ. It can be seen that RELAX learns a relaxation who's derivative points in the direction of decreased loss for all values of reparameterization noise u whereas REBAR's fixed relaxation only does for values of $u > t$. (NEED TO VERIFY THIS IS RIGHT)

[pictures, variances graphs]

[head to head comparisons]

6.2 Discrete Variational Autoencoder

As in (Tucker et al., 2017), we benchmark the RELAX estimator on the task of training a variational autoencoder **?** where all random variables are Bernoulli taking values in $\{-1, 1\}$. As in Tucker et al. (2017), we compare training the variational lower-bound across the MNIST **?** and Omniglot **?** datasets. As in Tucker et al. (2017) we test models with 1 and 2 of Bernoulli random variables with linear mappings between them and a model with 1 layer of Bernoulli random variables with non-linear mappings between layers.

We found that due to the complicated structure of the loss function, the RELAX estimator performed worse than REBAR. Instead we add a learned relaxation to REBAR's control variate which we denote relaxed-REBAR. Our estimator takes the form of **??** with

$$\bar{r}(z) = r(z) + f(\sigma_\lambda(z))$$

where $r(z)$ is a learned neural network and $f(\sigma_\lambda(z))$ is the Concrete relaxation of REBAR with temperature parameter λ. In all experiments, adding the learned $r(z)$ reduced the variance of the gradients and improved the final results.

In (Tucker et al., 2017), a separate REBAR estimator was used to estimate the gradients of each model parameter (each weight matrix and bias vector). To apply our estimator to this formulation, we would need to learn a separate relaxation for each model parameter. To get around this, we instead place one gradient estimator on each activation that parameterizes a layer of Bernoulli variables. We then back-propagate this gradient estimate to produce a gradient estimate for each model parameter. To provide a fair comparison, we re-implemented REBAR in this way (denoted REBAR-ours in table 6.2). We believe this explains the large difference in performance between our implementation and

MNIST gen.	REBAR (Tucker et al., 2017)	REBAR-ours	relaxed-REBAR
Linear 1 layer	-111.6	-111.66	**-111.22**
Linear 2 Layer	-98.8	-98.23	**-98.04**
Nonlinear	-101.1	-83.02	**-79.49**
Omniglot gen.			
Linear 1 layer	-116.83	-116.75	**-116.62**
Linear 2 Layer	-108.99	-108.74	**-108.59**
Nonlinear	-108.72	-62.28	**-58.55**

Figure 3: Training variational lower bound after training.

that of (Tucker et al., 2017) for the nonlinear models since there are 3 layers of parameters that all share the same gradient estimator. In the linear models, each layer has its own gradient estimator making our implementation closer to that of (Tucker et al., 2017).

[VAE, results, variances]

6.3 ATARI

Mnih et al. (2015)

[Do we use ADAM (Kingma & Ba, 2015) for optimization?]

7 RELATED WORK

Miller et al. (2017) further reduce the variance of reparameterization gradients in an orthogonal way.

As gradient estimators become more complex, checking their unbiasedness numerically becomes difficult. The automatic theorem-proving-based unbiasedness checker developed by Selsam et al. (2017) may become relevant to this line of research.

NVIL, VIMCO, things like that

Generalized Reparameterization Gradients REBAR and the generalization in this paper uses a mixture of score function and reparameterization gradients. A recent paper by Ruiz et al. (2016) unifies these two gradient estimators as the generalized reparameterization gradient (GRG). This framework can help disentangle the various components of generalized REBAR.

REBAR innovation as further decomposition the correction term into secondary reparameterization components note this is a recursive application of the principles of GRG observe that the GRG suggests this recursive application to components of an estimator propose that other estimators could be similarly recursively decomposed?

[TODO: cite certigrad arxiv 1706.08605]

8 LIMITATIONS

9 CONCLUSIONS AND FUTURE WORK

Other possible applications:

GANs (Goodfellow et al., 2014) that generate text or other discrete objects.

Learning to parse (Kusner et al., 2017)

REFERENCES

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural informa-*

tion processing systems, pp. 2672–2680, 2014.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations*, 2014.

Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. *arXiv preprint arXiv:1703.01925*, 2017.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

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.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Andrew C Miller, Nicholas J Foti, Alexander D'Amour, and Ryan P Adams. Reducing reparameterization gradient variance. *arXiv preprint arXiv:1705.07880*, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 02 2015. URL http://dx.doi.org/10.1038/nature14236.

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*, pp. 1928–1937, 2016.

Louis B Rall. Automatic differentiation: Techniques and applications. 1981.

Danilo J Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1278–1286, 2014.

Francisco R Ruiz, Michalis Titsias RC AUEB, and David Blei. The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems*, pp. 460–468, 2016.

David E Rumelhart and Geoffrey E Hinton. Learning representations by back-propagating errors. *Nature*, 323:9, 1986.

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

Daniel Selsam, Percy Liang, and David L Dill. Developing bug-free machine learning systems with formal mathematics. *International Conference on Machine Learning*, 2017.

Bert Speelpenning. *Compiling Fast Partial Derivatives of Functions Given by Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, 1980.

George Tucker, Andriy Mnih, Chris J Maddison, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *arXiv preprint arXiv:1703.07370*, 2017.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

10 APPENDIX A: CONTROL VARIATES

Generalizing the reparameterization trick

Write sample from distribution $s(\epsilon)$ as $\epsilon = \mathcal{T}^{-1}(\mathbf{z}; \nu)$ for some invertible transform $\mathcal{T}$ with variational parameters ν. write out transformed density example: normal with standard normal s example: inverse CDF of Gaussian with uniform s write out expected gradient under transformation show decomposition of expected gradient into reparameterization and correction terms

11 APPENDIX B: CATEGORICAL VARIABLES

Let $G_{1:k} = -\log -\log(U_{i:k})$ be samples from the Gumbel distribution, and learnable parameters $(\alpha_1, \ldots, \alpha_k)$ be interpreted as some unnormalized parameterization of the discrete distribution under consideration. Then, consider the following sampling procedure: for each k, find the k that maximizes $\log \alpha_k - G_k$, and then set $D_k = 1$ and $D_{i \neq k} = 0$. The Gumbel-Max trick states that sampling from the discrete distribution is equivalent to taking this argmax, that is, $p(D_k = 1) = \alpha_k / \sum_{i=1}^{n} \alpha_i$.

Since taking an argmax is still a discontinuous operation, Maddison et al. (2016) and Jang et al. (2016) proposed further relaxing the argmax operator through the softmax function with an additional temperature parameter λ:

$$x_k = \frac{\exp\{(\log \alpha_k + G_k)/\lambda\}}{\sum_{i=1}^{n} \exp\{(\log \alpha_i + G_i)/\lambda\}} \tag{11}$$

This relaxation allows values within the simplex, but in the low temperature limit, it becomes exactly the discrete argmax. One limitation of the concrete distribution is that it is a biased estimator except in limiting temperature. In other words, a small amount of bias is present for a non-zero temperature.