# Discrete Latent Variables

Art Sobolev

Research Scientist at Samsung AI Center

DeepBayes

1. Why Discreteness?
2. Problem
3. Relaxations
4. Variance Reduction
5. Conclusion

# Why Discreteness?

- Better interpretability
  - Easier to interpret discrete categories that continuous spectrum
- Manipulating control flow
  - Let the model make the discrete choice
- Inherent trait of the problem
  - Sometimes you need discrete predictions to have some properties

## Motivation



- Better interpretability
  - Easier to interpret discrete categories that continuous spectrum
- Manipulating control flow
  - Let the model make the discrete choice
- Inherent trait of the problem
  - Sometimes you need discrete predictions to have some properties

# Motivation

- Better interpretability
  - Easier to interpret discrete categories that continuous spectrum
- Manipulating control flow
  - Let the model make the discrete choice
- Inherent trait of the problem
  - Sometimes you need discrete predictions to have some properties

# Motivation

- Better interpretability
  - Easier to interpret discrete categories that continuous spectrum
- Manipulating control flow
  - Let the model make the discrete choice
- Inherent trait of the problem
  - Sometimes you need discrete predictions to have some properties

- Discrete Variational Autoencoder
  - Assume observations can be described by some binary (or categorical) code
  - We want to learn both encoder and decoder for such code and observations
- Hard Attention
  - An attention module generates binary mask on where to look at
  - The network classifies masked images
  - We want attention module to attend only important ares of an image
- GANs for text
  - Generator outputs discrete text
  - Discriminator takes discrete text as input and classifies how real it is
  - We want generator to output text that fools the discriminator

- Discrete Variational Autoencoder
  - Assume observations can be described by some binary (or categorical) code
  - We want to learn both encoder and decoder for such code and observations
- Hard Attention
  - An attention module generates binary mask on where to look at
  - The network classifies masked images
  - We want attention module to attend only important ares of an image
- GANs for text
  - Generator outputs discrete text
  - Discriminator takes discrete text as input and classifies how real it is
  - We want generator to output text that fools the discriminator

# Examples



- Discrete Variational Autoencoder
  - Assume observations can be described by some binary (or categorical) code
  - We want to learn both encoder and decoder for such code and observations
- Hard Attention
  - An attention module generates binary mask on where to look at
  - The network classifies masked images
  - We want attention module to attend only important ares of an image
- GANs for text
  - Generator outputs discrete text
  - Discriminator takes discrete text as input and classifies how real it is
  - We want generator to output text that fools the discriminator

- Discrete Variational Autoencoder
  - Assume observations can be described by some binary (or categorical) code
  - We want to learn both encoder and decoder for such code and observations
- Hard Attention
  - An attention module generates binary mask on where to look at
  - The network classifies masked images
  - We want attention module to attend only important ares of an image
- GANs for text
  - Generator outputs discrete text
  - Discriminator takes discrete text as input and classifies how real it is
  - We want generator to output text that fools the discriminator

# Problem

Typical problems boil down to optimizing the following objective

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(\mathbf{z})} f(\mathbf{z}) \to \max_\phi$$

- We consider models where objective is continuous w.r.t. $\phi$
  - Hence the gradient $\frac{\partial}{\partial \phi}\mathcal{L}(\phi)$ exists
- Expectation is intractable, resort to *Stochastic Optimization*
- Stochastic Optimization requires stochastic (unbiased) estimate $g(\mathbf{z}, \phi)$ of the true gradient:

$$\mathbb{E}_{q_\phi(\mathbf{z})} g(\mathbf{z}, \phi) = \frac{\partial}{\partial \phi}\mathcal{L}(\phi)$$

- No continuous reparametrization is possible for $z$
  - Because $z$ is taking finitely many different values

Typical problems boil down to optimizing the following objective

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(\mathbf{z})} f(\mathbf{z}) \to \max_\phi$$

- We consider models where objective is continuous w.r.t. $\phi$
  - Hence the gradient $\frac{\partial}{\partial \phi}\mathcal{L}(\phi)$ exists
- Expectation is intractable, resort to *Stochastic Optimization*
- Stochastic Optimization requires stochastic (unbiased) estimate $g(\mathbf{z}, \phi)$ of the true gradient:

$$\mathbb{E}_{q_\phi(\mathbf{z})} g(\mathbf{z}, \phi) = \frac{\partial}{\partial \phi}\mathcal{L}(\phi)$$

- No continuous reparametrization is possible for $z$
  - Because $z$ is taking finitely many different values

## Optimizational Point of View

Typical problems boil down to optimizing the following objective

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(\mathbf{z})} f(\mathbf{z}) \to \max_\phi$$

- We consider models where objective is continuous w.r.t. $\phi$
  - Hence the gradient $\frac{\partial}{\partial \phi}\mathcal{L}(\phi)$ exists
- Expectation is intractable, resort to *Stochastic Optimization*
- Stochastic Optimization requires stochastic (unbiased) estimate $g(\mathbf{z}, \phi)$ of the true gradient:

$$\mathbb{E}_{q_\phi(\mathbf{z})} g(\mathbf{z}, \phi) = \frac{\partial}{\partial \phi}\mathcal{L}(\phi)$$

- No continuous reparametrization is possible for $z$
  - Because $z$ is taking finitely many different values

## Optimizational Point of View



Typical problems boil down to optimizing the following objective

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(\mathbf{z})} f(\mathbf{z}) \to \max_\phi$$

- We consider models where objective is continuous w.r.t. $\phi$
  - Hence the gradient $\frac{\partial}{\partial \phi}\mathcal{L}(\phi)$ exists
- Expectation is intractable, resort to *Stochastic Optimization*
- Stochastic Optimization requires stochastic (unbiased) estimate $g(\mathbf{z}, \phi)$ of the true gradient:

$$\mathbb{E}_{q_\phi(\mathbf{z})} g(\mathbf{z}, \phi) = \frac{\partial}{\partial \phi}\mathcal{L}(\phi)$$

- No continuous reparametrization is possible for $z$
  - Because $z$ is taking finitely many different values

Typical problems boil down to optimizing the following objective

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(\mathbf{z})} f(\mathbf{z}) \to \max_\phi$$

- We consider models where objective is continuous w.r.t. $\phi$
  - Hence the gradient $\frac{\partial}{\partial \phi}\mathcal{L}(\phi)$ exists
- Expectation is intractable, resort to *Stochastic Optimization*
- Stochastic Optimization requires stochastic (unbiased) estimate $g(\mathbf{z}, \phi)$ of the true gradient:

$$\mathbb{E}_{q_\phi(\mathbf{z})} g(\mathbf{z}, \phi) = \frac{\partial}{\partial \phi}\mathcal{L}(\phi)$$

- No continuous reparametrization is possible for $z$
  - Because $z$ is taking finitely many different values

$$\frac{1}{M} \sum_{m=1}^{M} f(\mathbf{z}^{(m)}) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}^{(m)}), \qquad \mathbf{z}^{(m)} \sim q_\phi(\mathbf{z})$$

- Works for our case, discreteness does not get in the way
  - $f$ is not even required to be continuous
- Typically has large variance
- Requires sophisticated *Variance Reduction* methods
  - Just taking bigger $M$ won't help
  - Control Variates aka baselines, typically of the form

$$\frac{1}{M} \sum_{m=1}^{M} \left(f(\mathbf{z}^{(m)}) - b(\mathbf{z})\right) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}^{(m)}) + \frac{\partial}{\partial \phi} \mathbb{E}_{q_\phi(\mathbf{z})} b(\mathbf{z}),$$

for $b(\mathbf{z})$ s.t. the 2nd expectation is tractable and $\mathbf{z}^{(m)} \sim q_\phi(\mathbf{z})$

$$\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{z}^{(m)}) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}^{(m)}), \qquad \mathbf{z}^{(m)} \sim q_\phi(\mathbf{z})$$

- Works for our case, discreteness does not get in the way
  - $f$ is not even required to be continuous
- Typically has large variance
- Requires sophisticated *Variance Reduction* methods
  - Just taking bigger $M$ won't help
  - Control Variates aka baselines, typically of the form

$$\frac{1}{M}\sum_{m=1}^{M} \left(f(\mathbf{z}^{(m)}) - b(\mathbf{z})\right) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}^{(m)}) + \frac{\partial}{\partial \phi} \mathbb{E}_{q_\phi(\mathbf{z})} b(\mathbf{z}),$$

for $b(\mathbf{z})$ s.t. the 2nd expectation is tractable and $\mathbf{z}^{(m)} \sim q_\phi(\mathbf{z})$

$$\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{z}^{(m)}) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}^{(m)}), \qquad \mathbf{z}^{(m)} \sim q_\phi(\mathbf{z})$$

- Works for our case, discreteness does not get in the way
  - $f$ is not even required to be continuous
- Typically has large variance
- Requires sophisticated *Variance Reduction* methods
  - Just taking bigger $M$ won't help
  - Control Variates aka baselines, typically of the form

$$\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{z}^{(m)} - b(\mathbf{z})) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}^{(m)}) + \frac{\partial}{\partial \phi} \mathbb{E}_{q_\phi(\mathbf{z})} b(\mathbf{z}),$$

for $b(\mathbf{z})$ s.t. the 2nd expectation is tractable and $\mathbf{z}^{(m)} \sim q_\phi(\mathbf{z})$

# REINFORCE Estimator

$$\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{z}^{(m)})\frac{\partial}{\partial\phi}\log q_\phi(\mathbf{z}^{(m)}), \qquad \mathbf{z}^{(m)} \sim q_\phi(\mathbf{z})$$

- Works for our case, discreteness does not get in the way
  - $f$ is not even required to be continuous
- Typically has large variance
- Requires sophisticated *Variance Reduction* methods
  - Just taking bigger $M$ won't help
  - Control Variates aka baselines, typically of the form

$$\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{z}^{(m)} - b(\mathbf{z}))\frac{\partial}{\partial\phi}\log q_\phi(\mathbf{z}^{(m)}) + \frac{\partial}{\partial\phi}\mathbb{E}_{q_\phi(\mathbf{z})} b(\mathbf{z}),$$

for $b(\mathbf{z})$ s.t. the 2nd expectation is tractable and $\mathbf{z}^{(m)} \sim q_\phi(\mathbf{z})$

Consider 1-sample estimate

$$g^{\text{REINFORCE}}(\mathbf{z}, \phi) = \overbrace{f(\mathbf{z})}^{\text{scalar}} \overbrace{\frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z})}^{\text{vector}}, \qquad \mathbf{z} \sim q_\phi(\mathbf{z})$$

- Gradient estimate points in the direction of increasing probability of a given sample $\mathbf{z}$
  - Increases probability of $\mathbf{z}$ if it happened to be good
- The target function $f$ only enters as a scaling coefficient, and no gradient $\frac{\partial f}{\partial \mathbf{z}}$ is used
  - Has no idea where to move probability mass *systematically*
- **Random search in disguise!** [Rec18]

Consider 1-sample estimate

$$g^{\text{REINFORCE}}(\mathbf{z}, \phi) = \underbrace{f(\mathbf{z})}_{\text{scalar}} \overbrace{\frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z})}^{\text{vector}}, \qquad \mathbf{z} \sim q_\phi(\mathbf{z})$$

- Gradient estimate points in the direction of increasing probability of a given sample $\mathbf{z}$
  - Increases probability of $\mathbf{z}$ if it happened to be good
- The target function $f$ only enters as a scaling coefficient, and no gradient $\frac{\partial f}{\partial \mathbf{z}}$ is used
  - Has no idea where to move probability mass *systematically*
- **Random search in disguise!** [Rec18]

Consider 1-sample estimate

$$g^{\text{REINFORCE}}(\mathbf{z}, \phi) = \underbrace{f(\mathbf{z})}_{\text{scalar}} \underbrace{\frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z})}_{\text{vector}}, \qquad \mathbf{z} \sim q_\phi(\mathbf{z})$$

- Gradient estimate points in the direction of increasing probability of a given sample $\mathbf{z}$
  - Increases probability of $\mathbf{z}$ if it happened to be good
- The target function $f$ only enters as a scaling coefficient, and no gradient $\frac{\partial f}{\partial z}$ is used
  - Has no idea where to move probability mass *systematically*
- **Random search in disguise!** [Rec18]

Consider 1-sample estimate

$$g^{\text{REINFORCE}}(\mathbf{z}, \phi) = \underbrace{f(\mathbf{z})}^{\text{scalar}} \overbrace{\frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z})}^{\text{vector}}, \qquad \mathbf{z} \sim q_\phi(\mathbf{z})$$

- Gradient estimate points in the direction of increasing probability of a given sample $\mathbf{z}$
  - Increases probability of $\mathbf{z}$ if it happened to be good
- The target function $f$ only enters as a scaling coefficient, and no gradient $\frac{\partial f}{\partial \mathbf{z}}$ is used
  - Has no idea where to move probability mass *systematically*
- **Random search in disguise!** [Rec18]

Consider 1-sample estimate

$$g^{\text{REINFORCE}}(\mathbf{z}, \phi) = \underbrace{f(\mathbf{z})}^{\text{scalar}} \overbrace{\frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z})}^{\text{vector}}, \qquad \mathbf{z} \sim q_\phi(\mathbf{z})$$

- Gradient estimate points in the direction of increasing probability of a given sample $\mathbf{z}$
  - Increases probability of $\mathbf{z}$ if it happened to be good
- The target function $f$ only enters as a scaling coefficient, and no gradient $\frac{\partial f}{\partial \mathbf{z}}$ is used
  - Has no idea where to move probability mass *systematically*
- **Random search in disguise!** [Rec18]

# Relaxations

**Idea**: Relax the objective over discrete random samples into an objective oven continuous random samples during training and use the reparametrization trick:

$$\mathbb{E}_{q_\phi(\mathbf{z})} f(\mathbf{z}) \rightsquigarrow \mathbb{E}_{q_\phi(\mathbf{z})} f(\widetilde{\mathbf{z}}) \rightsquigarrow \mathbb{E}_{p(\gamma)} f(\widetilde{\mathbf{z}}(\gamma, \phi))$$

Keep the testing phase model unchanged

This requires $f$ to be able to work with relaxed values

- Limits the scope

**Idea**: Relax the objective over discrete random samples into an objective oven continuous random samples during training and use the reparametrization trick:

$$\mathbb{E}_{q_\phi(\mathbf{z})} f(\mathbf{z}) \rightsquigarrow \mathbb{E}_{q_\phi(\mathbf{z})} f(\widetilde{\mathbf{z}}) \rightsquigarrow \mathbb{E}_{p(\gamma)} f(\widetilde{\mathbf{z}}(\gamma, \phi))$$

Keep the testing phase model unchanged

This requires $f$ to be able to work with relaxed values

- Limits the scope

**Idea**: Relax the objective over discrete random samples into an objective oven continuous random samples during training and use the reparametrization trick:

$$\mathbb{E}_{q_\phi(\mathbf{z})} f(\mathbf{z}) \rightsquigarrow \mathbb{E}_{q_\phi(\mathbf{z})} f(\widetilde{\mathbf{z}}) \rightsquigarrow \mathbb{E}_{p(\gamma)} f(\widetilde{\mathbf{z}}(\gamma, \phi))$$

Keep the testing phase model unchanged

This requires $f$ to be able to work with relaxed values

- Limits the scope

**Idea**: Relax the objective over discrete random samples into an objective oven continuous random samples during training and use the reparametrization trick:

$$\mathbb{E}_{q_\phi(\mathbf{z})} f(\mathbf{z}) \rightsquigarrow \mathbb{E}_{q_\phi(\mathbf{z})} f(\widetilde{\mathbf{z}}) \rightsquigarrow \mathbb{E}_{p(\gamma)} f(\widetilde{\mathbf{z}}(\gamma, \phi))$$

Keep the testing phase model unchanged

This requires $f$ to be able to work with relaxed values

- Limits the scope

An old trick to sample Categorical random variables

$$z \sim \text{Categorical}(\pi_1, \dots, \pi_K),$$

Minimum of independent exponential distributions with carefully chosen probabilities has the same distribution:

$$z \stackrel{d}{=} \underset{k}{\text{argmin}} \frac{\xi_k}{\pi_k}, \qquad \xi_k \sim \text{Exp}(1)$$

Equivalently (applying $-\log$)

$$z \stackrel{d}{=} \underset{k}{\text{argmax}} \left[\log \pi_k - \log \xi_k\right], \qquad \xi_k \sim \text{Exp}(1)$$

Converts sampling $K$-ary discrete random variable into optimization by adding univariate noise $K$ times

An old trick to sample Categorical random variables

$$z \sim \text{Categorical}(\pi_1, \dots, \pi_K),$$

Minimum of independent exponential distributions with carefully chosen probabilities has the same distribution:

$$z \overset{d}{=} \underset{k}{\text{argmin}} \frac{\xi_k}{\pi_k}, \qquad \xi_k \sim \text{Exp}(1)$$

Equivalently (applying $-\log$)

$$z \overset{d}{=} \underset{k}{\text{argmax}} \left[\log \pi_k - \log \xi_k\right], \qquad \xi_k \sim \text{Exp}(1)$$

Converts sampling $K$-ary discrete random variable into optimization by adding univariate noise $K$ times

## Gumbel-Max Trick

An old trick to sample Categorical random variables

$$z \sim \text{Categorical}(\pi_1, \dots, \pi_K),$$

Minimum of independent exponential distributions with carefully chosen probabilities has the same distribution:

$$z \stackrel{d}{=} \underset{k}{\text{argmin}} \frac{\xi_k}{\pi_k}, \qquad \xi_k \sim \text{Exp}(1)$$

Equivalently (applying $-\log$)

$$z \stackrel{d}{=} \underset{k}{\text{argmax}} \left[\log \pi_k - \log \xi_k\right], \qquad \xi_k \sim \text{Exp}(1)$$

Converts sampling $K$-ary discrete random variable into optimization by adding univariate noise $K$ times

An old trick to sample Categorical random variables

$$z \sim \text{Categorical}(\pi_1, \dots, \pi_K),$$

Minimum of independent exponential distributions with carefully chosen probabilities has the same distribution:

$$z \stackrel{d}{=} \underset{k}{\text{argmin}} \frac{\xi_k}{\pi_k}, \qquad \xi_k \sim \text{Exp}(1)$$

Equivalently (applying $-\log$)

$$z \stackrel{d}{=} \underset{k}{\text{argmax}} \left[\log \pi_k - \log \xi_k\right], \qquad \xi_k \sim \text{Exp}(1)$$

Converts sampling $K$-ary discrete random variable into optimization by adding univariate noise $K$ times

# Gumbel-Softmax Trick [JGP17], [MMT17]

Approximate argmax with softmax (with temperature)

$$\text{softmax}_\tau(x)_j = \frac{\exp(x_j/\tau)}{\sum_{k=1}^K \exp(x_k/\tau)}$$

Temperature controls "sharpness" of the softmax:

- $\tau = 0$ recovers argmax $= \text{softmax}_0$
- $\tau = \infty$ leads to uniform distribution ignoring any disparities

We then replace discrete $z$ with their continuous relaxations $\widetilde{z}$

$$\widetilde{z}(\gamma, \pi) = \text{softmax}_\tau\left(\log \pi_1 + \gamma_1, \dots, \log \pi_K + \gamma_K\right)$$

Where $\gamma_k \sim \text{Gumbel}(0, 1)$ is a standard Gumbel random variable, and can be generated from uniform noise $u_k$ as

$$\gamma_k \stackrel{d}{=} -\log(-\log u_k), \qquad u_k \sim \text{Uniform}(0, 1)$$

## Gumbel-Softmax Trick [JGP17], [MMT17]



Approximate argmax with softmax (with temperature)

$$\mathrm{softmax}_\tau(x)_j = \frac{\exp(x_j/\tau)}{\sum_{k=1}^K \exp(x_k/\tau)}$$

Temperature controls "sharpness" of the softmax:

- $\tau = 0$ recovers argmax $= \mathrm{softmax}_0$
- $\tau = \infty$ leads to uniform distribution ignoring any disparities

We then replace discrete $z$ with their continuous relaxations $\widetilde{z}$

$$\widetilde{z}(\gamma, \pi) = \mathrm{softmax}_\tau \left(\log \pi_1 + \gamma_1, \dots, \log \pi_K + \gamma_K\right)$$

Where $\gamma_k \sim \mathrm{Gumbel}(0, 1)$ is a standard Gumbel random variable, and can be generated from uniform noise $u_k$ as

$$\gamma_k \overset{d}{=} -\log(-\log u_k), \qquad u_k \sim \mathrm{Uniform}(0, 1)$$

## Gumbel-Softmax Trick [JGP17], [MMT17]



Approximate argmax with softmax (with temperature)

$$\text{softmax}_\tau(x)_j = \frac{\exp(x_j/\tau)}{\sum_{k=1}^K \exp(x_k/\tau)}$$

Temperature controls "sharpness" of the softmax:

- $\tau = 0$ recovers argmax $= \text{softmax}_0$
- $\tau = \infty$ leads to uniform distribution ignoring any disparities

We then replace discrete $z$ with their continuous relaxations $\widetilde{z}$

$$\widetilde{z}(\gamma, \pi) = \text{softmax}_\tau\left(\log \pi_1 + \gamma_1, \dots, \log \pi_K + \gamma_K\right)$$

Where $\gamma_k \sim \text{Gumbel}(0, 1)$ is a standard Gumbel random variable, and can be generated from uniform noise $u_k$ as

$$\gamma_k \stackrel{d}{=} -\log(-\log u_k), \qquad u_k \sim \text{Uniform}(0, 1)$$

Now we can rewrite the expectation w.r.t. independent noise $\gamma_1, \dots, \gamma_K$

$$\mathcal{L}(\phi) = \mathbb{E}_\gamma f(\widetilde{\mathbf{z}}(\gamma, \phi)), \qquad \gamma_{dk} \sim \text{Gumbel}(0, 1)$$

Gradient estimate is obtained simply by exchanging $\frac{\partial}{\partial \phi}$ and $\mathbb{E}$:

$$g^{\text{Rep}}(\gamma, \phi) = \frac{\partial}{\partial \phi} f(\widetilde{z}(\gamma, \pi(\phi)))$$

Similar to stochastic discrete nodes replaced by their expectation (softmax), but has **noise injected into log-probabilities**

- Rooted in (Approximate) Bayesian Inference
- Noise helps exploration and regularizes
- Right kind of noise makes $\widetilde{z}$ similar to one-hot vectors
  - Reducing train-test mismatch

Now we can rewrite the expectation w.r.t. independent noise $\gamma_1, \dots, \gamma_K$

$$\mathcal{L}(\phi) = \mathbb{E}_\gamma f(\widetilde{\mathbf{z}}(\gamma, \phi)), \qquad \gamma_{dk} \sim \text{Gumbel}(0, 1)$$

Gradient estimate is obtained simply by exchanging $\frac{\partial}{\partial \phi}$ and $\mathbb{E}$:

$$g^{\text{Rep}}(\gamma, \phi) = \frac{\partial}{\partial \phi} f(\widetilde{z}(\gamma, \pi(\phi)))$$

Similar to stochastic discrete nodes replaced by their expectation (softmax), but has **noise injected into log-probabilities**

- Rooted in (Approximate) Bayesian Inference
- Noise helps exploration and regularizes
- Right kind of noise makes $\widetilde{z}$ similar to one-hot vectors
  - Reducing train-test mismatch

Now we can rewrite the expectation w.r.t. independent noise $\gamma_1, \dots, \gamma_K$

$$\mathcal{L}(\phi) = \mathbb{E}_\gamma f(\widetilde{\mathbf{z}}(\gamma, \phi)), \qquad \gamma_{dk} \sim \text{Gumbel}(0, 1)$$

Gradient estimate is obtained simply by exchanging $\frac{\partial}{\partial \phi}$ and $\mathbb{E}$:

$$g^{\text{Rep}}(\gamma, \phi) = \frac{\partial}{\partial \phi} f(\widetilde{z}(\gamma, \pi(\phi)))$$

Similar to stochastic discrete nodes replaced by their expectation (softmax), but has **noise injected into log-probabilities**

- Rooted in (Approximate) Bayesian Inference
- Noise helps exploration and regularizes
- Right kind of noise makes $\widetilde{z}$ similar to one-hot vectors
  - Reducing train-test mismatch

In a special case of $K = 2$ we can write a bit more sample-efficient scheme.

$$\widetilde{z} = \sigma_\tau\left(\log \tfrac{p}{1-p} + \upsilon\right), \qquad \upsilon \sim \text{Logistic}(0, 1)$$

where Logistic(0, 1) is the distribution of difference of 2 Gumbels:

$$\upsilon \overset{d}{=} \gamma_1 - \gamma_2$$

You can easily see this by yourself by simplifying the formula of the softmax over 2 classes.

In a special case of $K = 2$ we can write a bit more sample-efficient scheme.

$$\widetilde{z} = \sigma_\tau\left(\log \tfrac{p}{1-p} + \upsilon\right), \qquad \upsilon \sim \text{Logistic}(0, 1)$$

where Logistic(0, 1) is the distribution of difference of 2 Gumbels:

$$\upsilon \stackrel{d}{=} \gamma_1 - \gamma_2$$

You can easily see this by yourself by simplifying the formula of the softmax over 2 classes.

# Gumbel-Softmax Trick: Binary Case

In a special case of $K = 2$ we can write a bit more sample-efficient scheme.

$$\widetilde{z} = \sigma_\tau \left( \log \tfrac{p}{1-p} + \upsilon \right), \qquad \upsilon \sim \text{Logistic}(0, 1)$$

where Logistic(0, 1) is the distribution of difference of 2 Gumbels:

$$\upsilon \overset{d}{=} \gamma_1 - \gamma_2$$

You can easily see this by yourself by simplifying the formula of the softmax over 2 classes.

In a special case of $K = 2$ we can write a bit more sample-efficient scheme.

$$\widetilde{z} = \sigma_\tau\left(\log \tfrac{p}{1-p} + \upsilon\right), \qquad \upsilon \sim \text{Logistic}(0, 1)$$

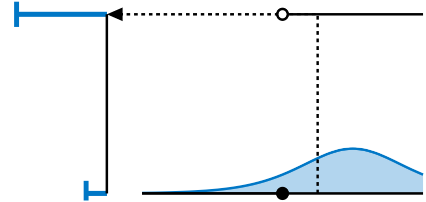
where Logistic(0, 1) is the distribution of difference of 2 Gumbels:

$$\upsilon \overset{d}{=} \gamma_1 - \gamma_2$$

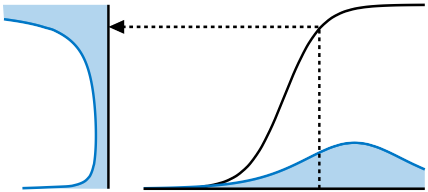
You can easily see this by yourself by simplifying the formula of the softmax over 2 classes.

# Gumbel-Softmax Trick: Temperature

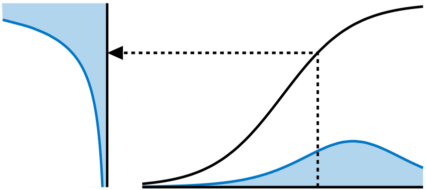




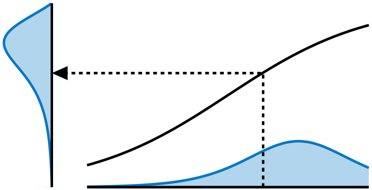
$\tau = 0$



$\tau = 1/2$



$\tau = 1$



$\tau = 2$

For $K$-ary categorical r.v. it has been shown that for $\tau \leqslant \frac{1}{K-1}$ there are no modes in the interior of the probability simplex

- All modes are in the vertices, which are one-hot vectors
  - (or edges, which is still good, since at least one component is close to zero)
- This makes relaxed samples more likely to be *contrastive*
  - Similar to actual discrete samples
  - Forces the model to adapt to the corresponding mode

How to choose the temperature?

- Small temperature leads to high variances, but resembles discrete case well
- Large temperatures have lower variance, but deviates away from the discrete case
- In practice grid search over a couple possible values

For $K$-ary categorical r.v. it has been shown that for $\tau \leqslant \frac{1}{K-1}$ there are no modes in the interior of the probability simplex

- All modes are in the vertices, which are one-hot vectors
  - (or edges, which is still good, since at least one component is close to zero)
- This makes relaxed samples more likely to be *contrastive*
  - Similar to actual discrete samples
  - Forces the model to adapt to the corresponding mode

How to choose the temperature?

- Small temperature leads to high variances, but resembles discrete case well
- Large temperatures have lower variance, but deviates away from the discrete case
- In practice grid search over a couple possible values

For $K$-ary categorical r.v. it has been shown that for $\tau \leqslant \frac{1}{K-1}$ there are no modes in the interior of the probability simplex

- All modes are in the vertices, which are one-hot vectors
  - (or edges, which is still good, since at least one component is close to zero)
- This makes relaxed samples more likely to be *contrastive*
  - Similar to actual discrete samples
  - Forces the model to adapt to the corresponding mode

How to choose the temperature?

- Small temperature leads to high variances, but resembles discrete case well
- Large temperatures have lower variance, but deviates away from the discrete case
- In practice grid search over a couple possible values

# Gumbel-Softmax Trick: Temperature

For $K$-ary categorical r.v. it has been shown that for $\tau \leqslant \frac{1}{K-1}$ there are no modes in the interior of the probability simplex

- All modes are in the vertices, which are one-hot vectors
  - (or edges, which is still good, since at least one component is close to zero)
- This makes relaxed samples more likely to be *contrastive*
  - Similar to actual discrete samples
  - Forces the model to adapt to the corresponding mode

How to choose the temperature?

- Small temperature leads to high variances, but resembles discrete case well
- Large temperatures have lower variance, but deviates away from the discrete case
- In practice grid search over a couple possible values

## Gumbel-Softmax Trick: Conclusions

DeepBayes

- Gumbel-Softmax relaxes discrete random variables into continuous, enabling the reparametrization trick
- Relaxations change the objective, yet no theory on how good the relaxation is
  - Relaxation introduces bias
- Temperature $\tau$ is a hyperparameter that needs to be tuned

There exist other relaxations, however they are

- Less mathematically elegant
- Do not seem to work better empirically
- Sometimes heuristic in nature

# Gumbel-Softmax Trick: Conclusions



- Gumbel-Softmax relaxes discrete random variables into continuous, enabling the reparametrization trick
- Relaxations change the objective, yet no theory on how good the relaxation is
  - Relaxation introduces bias
- Temperature $\tau$ is a hyperparameter that needs to be tuned

There exist other relaxations, however they are

- Less mathematically elegant
- Do not seem to work better empirically
- Sometimes heuristic in nature

## Gumbel-Softmax Trick: Conclusions

Deep|Bayes

- Gumbel-Softmax relaxes discrete random variables into continuous, enabling the reparametrization trick
- Relaxations change the objective, yet no theory on how good the relaxation is
  - Relaxation introduces bias
- Temperature $\tau$ is a hyperparameter that needs to be tuned

There exist other relaxations, however they are

- Less mathematically elegant
- Do not seem to work better empirically
- Sometimes heuristic in nature

## Gumbel-Softmax Trick: Conclusions

DeepBayes

- Gumbel-Softmax relaxes discrete random variables into continuous, enabling the reparametrization trick
- Relaxations change the objective, yet no theory on how good the relaxation is
  - Relaxation introduces bias
- Temperature $\tau$ is a hyperparameter that needs to be tuned

There exist other relaxations, however they are

- Less mathematically elegant
- Do not seem to work better empirically
- Sometimes heuristic in nature

## Gumbel-Softmax Trick: Conclusions

- Gumbel-Softmax relaxes discrete random variables into continuous, enabling the reparametrization trick
- Relaxations change the objective, yet no theory on how good the relaxation is
  - Relaxation introduces bias
- Temperature $\tau$ is a hyperparameter that needs to be tuned

There exist other relaxations, however they are

- Less mathematically elegant
- Do not seem to work better empirically
- Sometimes heuristic in nature

# Variance Reduction

Consider some $b(\mathbf{z})$ with **tractable** expectation $\mu = \mathbb{E}_{q(\mathbf{z})} b(\mathbf{z})$

$$\frac{1}{M} \sum_{m=1}^{M} (f(\mathbf{z}_m) - b(\mathbf{z}_m)) + \mu$$

Might be a better (low-variance) estimate if $f(\mathbf{z})$ and $b(\mathbf{z})$ have positive correlation.

- Unbiased estimator
- $b(\mathbf{z})$ is called *Control Variate*
- Especially convenient if $b(\mathbf{z})$ is zero-mean
- We can choose any $b(\mathbf{z})$ we want

Essentially, $b(\mathbf{z})$ extracts some tractable part of the $f(\mathbf{z})$ and estimates the rest using Monte Carlo.

# Control Variates

Consider some $b(\mathbf{z})$ with **tractable** expectation $\mu = \mathbb{E}_{q(\mathbf{z})} b(\mathbf{z})$

$$\frac{1}{M} \sum_{m=1}^{M} (f(\mathbf{z}_m) - b(\mathbf{z}_m)) + \mu$$

Might be a better (low-variance) estimate if $f(\mathbf{z})$ and $b(\mathbf{z})$ have positive correlation.

- Unbiased estimator
- $b(\mathbf{z})$ is called *Control Variate*
- Especially convenient if $b(\mathbf{z})$ is zero-mean
- We can choose any $b(\mathbf{z})$ we want

Essentially, $b(\mathbf{z})$ extracts some tractable part of the $f(\mathbf{z})$ and estimates the rest using Monte Carlo.

## Control Variates



Consider some $b(\mathbf{z})$ with **tractable** expectation $\mu = \mathbb{E}_{q(\mathbf{z})} b(\mathbf{z})$

$$\frac{1}{M} \sum_{m=1}^{M} (f(\mathbf{z}_m) - b(\mathbf{z}_m)) + \mu$$

Might be a better (low-variance) estimate if $f(\mathbf{z})$ and $b(\mathbf{z})$ have positive correlation.

- Unbiased estimator
- $b(\mathbf{z})$ is called *Control Variate*
- Especially convenient if $b(\mathbf{z})$ is zero-mean
- We can choose any $b(\mathbf{z})$ we want

Essentially, $b(\mathbf{z})$ extracts some tractable part of the $f(\mathbf{z})$ and estimates the rest using Monte Carlo.

$$g_b^{\text{REINFORCE}}(\mathbf{z}, \phi) = (f(\mathbf{z}) - b(\mathbf{z}))\frac{\partial}{\partial\phi}\log q_\phi(\mathbf{z}) + \frac{\partial}{\partial\phi}\mu(\phi)$$

- $b(\mathbf{z})$ is typically called *baseline*
- Essentially a control variate of the form $b(\mathbf{z})\frac{\partial}{\partial\phi}\log q_\phi(\mathbf{z})$
  - Other CVs are possible, but this is convenient as it approximates the function itself
- Unbiased estimate of the true gradient since $\mathbb{E}_{q_\phi(\mathbf{z})} b(\mathbf{z})\frac{\partial}{\partial\phi}\log q_\phi(\mathbf{z}) = \frac{\partial}{\partial\phi}\mu(\phi)$
- For right $b(\mathbf{z})$ might have much lower variance

The idea is the same: extract tractable part of $f(\mathbf{z})$, compute its gradient analytically, handle the rest using REINFORCE

# Control Variates for REINFORCE

$$g_b^{\text{REINFORCE}}(\mathbf{z}, \phi) = (f(\mathbf{z}) - b(\mathbf{z})) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) + \frac{\partial}{\partial \phi} \mu(\phi)$$

- $b(\mathbf{z})$ is typically called *baseline*
- Essentially a control variate of the form $b(\mathbf{z}) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z})$
  - Other CVs are possible, but this is convenient as it approximates the function itself
- Unbiased estimate of the true gradient since $\mathbb{E}_{q_\phi(\mathbf{z})} b(\mathbf{z}) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) = \frac{\partial}{\partial \phi} \mu(\phi)$
- For right $b(\mathbf{z})$ might have much lower variance

The idea is the same: extract tractable part of $f(\mathbf{z})$, compute its gradient analytically, handle the rest using REINFORCE

$$g_b^{\text{REINFORCE}}(\mathbf{z}, \phi) = (f(\mathbf{z}) - b(\mathbf{z})) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) + \frac{\partial}{\partial \phi} \mu(\phi)$$

- $b(\mathbf{z})$ is typically called *baseline*
- Essentially a control variate of the form $b(\mathbf{z}) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z})$
  - Other CVs are possible, but this is convenient as it approximates the function itself
- Unbiased estimate of the true gradient since $\mathbb{E}_{q_\phi(\mathbf{z})} b(\mathbf{z}) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) = \frac{\partial}{\partial \phi} \mu(\phi)$
- For right $b(\mathbf{z})$ might have much lower variance

The idea is the same: extract tractable part of $f(\mathbf{z})$, compute its gradient analytically, handle the rest using REINFORCE

$$g_b^{\text{REINFORCE}}(\mathbf{z}, \phi) = (f(\mathbf{z}) - b(\mathbf{z}))\frac{\partial}{\partial \phi}\log q_\phi(\mathbf{z}) + \frac{\partial}{\partial \phi}\mu(\phi)$$

- $b(\mathbf{z})$ is typically called *baseline*
- Essentially a control variate of the form $b(\mathbf{z})\frac{\partial}{\partial \phi}\log q_\phi(\mathbf{z})$
  - Other CVs are possible, but this is convenient as it approximates the function itself
- Unbiased estimate of the true gradient since $\mathbb{E}_{q_\phi(\mathbf{z})} b(\mathbf{z})\frac{\partial}{\partial \phi}\log q_\phi(\mathbf{z}) = \frac{\partial}{\partial \phi}\mu(\phi)$
- For right $b(\mathbf{z})$ might have much lower variance

The idea is the same: extract tractable part of $f(\mathbf{z})$, compute its gradient analytically, handle the rest using REINFORCE

# Control Variates for REINFORCE

$$g_b^{\text{REINFORCE}}(\mathbf{z}, \phi) = (f(\mathbf{z}) - b(\mathbf{z}))\frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) + \frac{\partial}{\partial \phi}\mu(\phi)$$

- $b(\mathbf{z})$ is typically called *baseline*
- Essentially a control variate of the form $b(\mathbf{z})\frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z})$
  - Other CVs are possible, but this is convenient as it approximates the function itself
- Unbiased estimate of the true gradient since $\mathbb{E}_{q_\phi(\mathbf{z})} b(\mathbf{z})\frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) = \frac{\partial}{\partial \phi}\mu(\phi)$
- For right $b(\mathbf{z})$ might have much lower variance

The idea is the same: extract tractable part of $f(\mathbf{z})$, compute its gradient analytically, handle the rest using REINFORCE

# Control Variates for REINFORCE



$$g_b^{\text{REINFORCE}}(\mathbf{z}, \phi) = (f(\mathbf{z}) - b(\mathbf{z})) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) + \frac{\partial}{\partial \phi} \mu(\phi)$$

- $b(\mathbf{z})$ is typically called *baseline*
- Essentially a control variate of the form $b(\mathbf{z}) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z})$
  - Other CVs are possible, but this is convenient as it approximates the function itself
- Unbiased estimate of the true gradient since $\mathbb{E}_{q_\phi(\mathbf{z})} b(\mathbf{z}) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) = \frac{\partial}{\partial \phi} \mu(\phi)$
- For right $b(\mathbf{z})$ might have much lower variance

The idea is the same: extract tractable part of $f(\mathbf{z})$, compute its gradient analytically, handle the rest using REINFORCE

- **Constant baseline $b(\mathbf{z}) = c$**

$$(f(\mathbf{z}) - c)\frac{\partial}{\partial\phi}\log q_\phi(\mathbf{z}) + \overbrace{\frac{\partial}{\partial\phi}\mathbb{E}_{q_\phi(\mathbf{z})} c}^{=0}$$

Just centers the learning signal, optimal c is

$$c = \frac{\text{Cov}\left[f(\mathbf{z})\frac{\partial}{\partial\phi}\log q_\phi(\mathbf{z}), \frac{\partial}{\partial\phi}\log q_\phi(\mathbf{z})\right]}{\text{Var}\left[\frac{\partial}{\partial\phi}\log q_\phi(\mathbf{z})\right]}$$

- **NVIL** [MG14]: If some extra observation is available (like $x$ in VAE), we can consider some learnable $b(x)$ as a baseline
No analytic solution for $b(x)$, minimize expected MSE

$$\mathbb{E}_{p(x)}\mathbb{E}_{q_\phi(z|x)}(f(\mathbf{z}) - b(x))^2 \to \min_{b(x)}$$

- **Constant baseline $b(\mathbf{z}) = c$**

$$(f(\mathbf{z}) - c) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) + \overbrace{\frac{\partial}{\partial \phi} \mathbb{E}_{q_\phi(\mathbf{z})} c}^{=0}$$

  Just centers the learning signal, optimal $c$ is

$$c = \frac{\text{Cov}\left[ f(\mathbf{z}) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}), \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) \right]}{\text{Var}\left[ \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) \right]}$$

- **NVIL** [MG14]: If some extra observation is available (like $x$ in VAE), we can consider some learnable $b(x)$ as a baseline
  No analytic solution for $b(x)$, minimize expected MSE

$$\mathbb{E}_{p(x)} \mathbb{E}_{q_\phi(z|x)} (f(\mathbf{z}) - b(x))^2 \to \min_{b(x)}$$

- **Constant baseline** $b(\mathbf{z}) = c$

$$(f(\mathbf{z}) - c)\frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) + \overbrace{\frac{\partial}{\partial \phi}\mathbb{E}_{q_\phi(\mathbf{z})} c}^{=0}$$

  Just centers the learning signal, optimal $c$ is

$$c = \frac{\text{Cov}\left[f(\mathbf{z})\frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}), \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z})\right]}{\text{Var}\left[\frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z})\right]}$$

- **NVIL** [MG14]: If some extra observation is available (like $x$ in VAE), we can consider some learnable $b(x)$ as a baseline
  No analytic solution for $b(x)$, minimize expected MSE

$$\mathbb{E}_{p(x)}\mathbb{E}_{q_\phi(z|x)}(f(\mathbf{z}) - b(x))^2 \to \min_{b(x)}$$

Let $b(\mathbf{z})$ be first-order Taylor expansion of $f(\mathbf{z})$ at some point $\mathbf{z}_0$:

$$b(\mathbf{z}) = f(\mathbf{z}_0) + \frac{\partial f}{\partial \mathbf{z}}(\mathbf{z}_0)^T(\mathbf{z} - \mathbf{z}_0)$$

We'll take $\mathbf{z}_0(\phi) = \mathbb{E}_{q_\phi(\mathbf{z})}\mathbf{z}$. This leads to

$$g^{\mu\text{-prop}}(\mathbf{z}, \phi) = (f(\mathbf{z}) - b(\mathbf{z}))\frac{\partial}{\partial \phi}\log q_\phi(\mathbf{z}) + \frac{\partial f}{\partial \mathbf{z}}(\mathbf{z}_0(\phi))\frac{\partial z_0(\phi)}{\partial \phi}$$

- Backpropagates through the mean, and then fine-tunes inaccuracies with REINFORCE
- One could use 2nd order Taylor expansion, but that is more computationally expensive

Let $b(\mathbf{z})$ be first-order Taylor expansion of $f(\mathbf{z})$ at some point $\mathbf{z}_0$:

$$b(\mathbf{z}) = f(\mathbf{z}_0) + \frac{\partial f}{\partial \mathbf{z}}(\mathbf{z}_0)^T(\mathbf{z} - \mathbf{z}_0)$$

We'll take $\mathbf{z}_0(\phi) = \mathbb{E}_{q_\phi(\mathbf{z})}\mathbf{z}$. This leads to

$$g^{\mu\text{-prop}}(\mathbf{z}, \phi) = (f(\mathbf{z}) - b(\mathbf{z}))\frac{\partial}{\partial \phi}\log q_\phi(\mathbf{z}) + \frac{\partial f}{\partial \mathbf{z}}(\mathbf{z}_0(\phi))\frac{\partial z_0(\phi)}{\partial \phi}$$

- Backpropagates through the mean, and then fine-tunes inaccuracies with REINFORCE
- One could use 2nd order Taylor expansion, but that is more computationally expensive

Let $b(\mathbf{z})$ be first-order Taylor expansion of $f(\mathbf{z})$ at some point $\mathbf{z}_0$:

$$b(\mathbf{z}) = f(\mathbf{z}_0) + \frac{\partial f}{\partial \mathbf{z}}(\mathbf{z}_0)^T (\mathbf{z} - \mathbf{z}_0)$$

We'll take $\mathbf{z}_0(\phi) = \mathbb{E}_{q_\phi(\mathbf{z})} \mathbf{z}$. This leads to

$$g^{\mu\text{-prop}}(\mathbf{z}, \phi) = (f(\mathbf{z}) - b(\mathbf{z})) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) + \frac{\partial f}{\partial \mathbf{z}}(\mathbf{z}_0(\phi)) \frac{\partial z_0(\phi)}{\partial \phi}$$

- Backpropagates through the mean, and then fine-tunes inaccuracies with REINFORCE
- One could use 2nd order Taylor expansion, but that is more computationally expensive

Let $b(\mathbf{z})$ be first-order Taylor expansion of $f(\mathbf{z})$ at some point $\mathbf{z}_0$:

$$b(\mathbf{z}) = f(\mathbf{z}_0) + \frac{\partial f}{\partial \mathbf{z}}(\mathbf{z}_0)^T(\mathbf{z} - \mathbf{z}_0)$$

We'll take $\mathbf{z}_0(\phi) = \mathbb{E}_{q_\phi(\mathbf{z})}\mathbf{z}$. This leads to

$$g^{\mu\text{-prop}}(\mathbf{z}, \phi) = (f(\mathbf{z}) - b(\mathbf{z}))\frac{\partial}{\partial \phi}\log q_\phi(\mathbf{z}) + \frac{\partial f}{\partial \mathbf{z}}(\mathbf{z}_0(\phi))\frac{\partial z_0(\phi)}{\partial \phi}$$

- Backpropagates through the mean, and then fine-tunes inaccuracies with REINFORCE
- One could use 2nd order Taylor expansion, but that is more computationally expensive

## REBAR [TMM$^+$17]

Deep|Bayes

Use Gumbel-relaxed $f$ as a baseline

$$g^{\text{REBAR}}(\mathbf{z}, \phi) = \left(f(\mathbf{z}) - \eta f(\widetilde{\mathbf{z}}_\phi|\mathbf{z})\right) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) + \eta \frac{\partial}{\partial \phi} \left(f(\widetilde{\mathbf{z}}_\phi) - f(\widetilde{\mathbf{z}}_\phi|\mathbf{z})\right)$$

- $\widetilde{z}|z$ is *conditional relaxation* – relaxed sample that has known argmax, but otherwise arbitrary
  - Can be shown to be efficiently reparametrizable
- Backpropagates through Gumbel-relaxed $f$, but has additional corrections for the introduced bias
- **The baseline's expectation is intractable, but reparametrizable**

## REBAR [TMM+17]



Use Gumbel-relaxed $f$ as a baseline

$$g^{\text{REBAR}}(\mathbf{z}, \phi) = \left(f(\mathbf{z}) - \eta f(\widetilde{\mathbf{z}}_\phi|\mathbf{z})\right) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) + \eta \frac{\partial}{\partial \phi} \left(f(\widetilde{\mathbf{z}}_\phi) - f(\widetilde{\mathbf{z}}_\phi|\mathbf{z})\right)$$

- $\widetilde{z}|z$ is *conditional relaxation* – relaxed sample that has known argmax, but otherwise arbitrary
  - Can be shown to be efficiently reparametrizable
- Backpropagates through Gumbel-relaxed $f$, but has additional corrections for the introduced bias
- **The baseline's expectation is intractable, but reparametrizable**

# REBAR [TMM+17]

Use Gumbel-relaxed $f$ as a baseline

$$g^{\text{REBAR}}(\mathbf{z}, \phi) = \left(f(\mathbf{z}) - \eta f(\widetilde{\mathbf{z}}_\phi | \mathbf{z})\right) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) + \eta \frac{\partial}{\partial \phi} \left(f(\widetilde{\mathbf{z}}_\phi) - f(\widetilde{\mathbf{z}}_\phi | \mathbf{z})\right)$$

- $\widetilde{z}|z$ is *conditional relaxation* – relaxed sample that has known argmax, but otherwise arbitrary
  - Can be shown to be efficiently reparametrizable
- Backpropagates through Gumbel-relaxed $f$, but has additional corrections for the introduced bias
- The baseline's expectation is intractable, but reparametrizable

Use Gumbel-relaxed $f$ as a baseline

$$g^{\text{REBAR}}(\mathbf{z}, \phi) = \left(f(\mathbf{z}) - \eta f(\widetilde{\mathbf{z}}_\phi | \mathbf{z})\right) \frac{\partial}{\partial \phi} \log q_\phi(\mathbf{z}) + \eta \frac{\partial}{\partial \phi} \left(f(\widetilde{\mathbf{z}}_\phi) - f(\widetilde{\mathbf{z}}_\phi | \mathbf{z})\right)$$

- $\widetilde{z}|z$ is *conditional relaxation* – relaxed sample that has known argmax, but otherwise arbitrary
  - Can be shown to be efficiently reparametrizable
- Backpropagates through Gumbel-relaxed $f$, but has additional corrections for the introduced bias
- **The baseline's expectation is intractable, but reparametrizable**

Typically we seek low-variance estimators. Why not minimize the variance w.r.t. a baseline in the first place?

$$\text{Var}[g^{\text{REBAR}}(\mathbf{z}, \phi)] = \mathbb{E}g^{\text{REBAR}}(\mathbf{z}, \phi)^2 - \left(\mathbb{E}g^{\text{REBAR}}(\mathbf{z}, \phi)\right)^2$$

- In general minimizing variance leads to increase in bias
- Our estimators are unbiased for any baseline
- Typically unbiased estimate of variance requires two samples
  - In our case expected gradient does not depend on baseline, and one sample is enough

Minimize expected $L_2$ norm of the gradient

$$\mathbb{E}g^{\text{REBAR}}(\mathbf{z}, \phi)^2 \to \min_{\tau, \eta}$$

One step further: why optimize over $\tau$ only? Can learn another neural network as a baseline instead! Works when we don't have can't tinker with $f$ (RL) [GCW+18]

Typically we seek low-variance estimators. Why not minimize the variance w.r.t. a baseline in the first place?

$$\text{Var}[g^{\text{REBAR}}(\mathbf{z}, \phi)] = \mathbb{E}g^{\text{REBAR}}(\mathbf{z}, \phi)^2 - \left(\mathbb{E}g^{\text{REBAR}}(\mathbf{z}, \phi)\right)^2$$

- In general minimizing variance leads to increase in bias
- Our estimators are unbiased for any baseline
- Typically unbiased estimate of variance requires two samples
  - In our case expected gradient does not depend on baseline, and one sample is enough

Minimize expected $L_2$ norm of the gradient

$$\mathbb{E}g^{\text{REBAR}}(\mathbf{z}, \phi)^2 \to \min_{\tau, \eta}$$

One step further: why optimize over $\tau$ only? Can learn another neural network as a baseline instead! Works when we don't have can't tinker with $f$ (RL) [GCW+18]

## Variance Minimization

Typically we seek low-variance estimators. Why not minimize the variance w.r.t. a baseline in the first place?

$$\text{Var}[g^{\text{REBAR}}(\mathbf{z}, \phi)] = \mathbb{E} g^{\text{REBAR}}(\mathbf{z}, \phi)^2 - \left(\mathbb{E} g^{\text{REBAR}}(\mathbf{z}, \phi)\right)^2$$

- In general minimizing variance leads to increase in bias
- Our estimators are unbiased for any baseline
- Typically unbiased estimate of variance requires two samples
  - In our case expected gradient does not depend on baseline, and one sample is enough

Minimize expected $L_2$ norm of the gradient

$$\mathbb{E} g^{\text{REBAR}}(\mathbf{z}, \phi)^2 \to \min_{\tau, \eta}$$

One step further: why optimize over $\tau$ only? Can learn another neural network as a baseline instead! Works when we don't have can't tinker with $f$ (RL) [GCW+18]

Typically we seek low-variance estimators. Why not minimize the variance w.r.t. a baseline in the first place?

$$\mathrm{Var}[g^{\mathrm{REBAR}}(\mathbf{z}, \phi)] = \mathbb{E} g^{\mathrm{REBAR}}(\mathbf{z}, \phi)^2 - \left(\mathbb{E} g^{\mathrm{REBAR}}(\mathbf{z}, \phi)\right)^2$$

- In general minimizing variance leads to increase in bias
- Our estimators are unbiased for any baseline
- Typically unbiased estimate of variance requires two samples
  - In our case expected gradient does not depend on baseline, and one sample is enough

Minimize expected $L_2$ norm of the gradient

$$\mathbb{E} g^{\mathrm{REBAR}}(\mathbf{z}, \phi)^2 \to \min_{\tau, \eta}$$

One step further: why optimize over $\tau$ only? Can learn another neural network as a baseline instead! Works when we don't have can't tinker with $f$ (RL) [GCW$^+$18]

# Variance Minimization



Typically we seek low-variance estimators. Why not minimize the variance w.r.t. a baseline in the first place?

$$\text{Var}[g^{\text{REBAR}}(\mathbf{z}, \phi)] = \mathbb{E} g^{\text{REBAR}}(\mathbf{z}, \phi)^2 - \left(\mathbb{E} g^{\text{REBAR}}(\mathbf{z}, \phi)\right)^2$$

- In general minimizing variance leads to increase in bias
- Our estimators are unbiased for any baseline
- Typically unbiased estimate of variance requires two samples
  - In our case expected gradient does not depend on baseline, and one sample is enough

Minimize expected $L_2$ norm of the gradient

$$\mathbb{E} g^{\text{REBAR}}(\mathbf{z}, \phi)^2 \to \min_{\tau, \eta}$$

One step further: why optimize over $\tau$ only? Can learn another neural network as a baseline instead! Works when we don't have can't tinker with $f$ (RL) [GCW+18]

# Variance Minimization

Typically we seek low-variance estimators. Why not minimize the variance w.r.t. a baseline in the first place?

$$\text{Var}[g^{\text{REBAR}}(\mathbf{z}, \phi)] = \mathbb{E}g^{\text{REBAR}}(\mathbf{z}, \phi)^2 - \left(\mathbb{E}g^{\text{REBAR}}(\mathbf{z}, \phi)\right)^2$$

- In general minimizing variance leads to increase in bias
- Our estimators are unbiased for any baseline
- Typically unbiased estimate of variance requires two samples
  - In our case expected gradient does not depend on baseline, and one sample is enough

Minimize expected $L_2$ norm of the gradient

$$\mathbb{E}g^{\text{REBAR}}(\mathbf{z}, \phi)^2 \to \min_{\tau,\eta}$$

One step further: why optimize over $\tau$ only? Can learn another neural network as a baseline instead! Works when we don't have can't tinker with $f$ (RL) [GCW+18]

# Conclusion

## Conclusion



Relaxation-based methods

- Straightforward to implement
- Work well in practice
- Have hyperparameters to tune
- Have biased gradients aka introduce train-test mismatch

Variance Reduction methods

- Cumbersome
- Not clear if their results are worth added complexity
- Always unbiased
- Allow you to tune baseline to minimize variance
- **Random search on steroids**

Still ongoing research topic, many other approaches not covered

## Conclusion



Relaxation-based methods

- Straightforward to implement
- Work well in practice
- Have hyperparameters to tune
- Have biased gradients aka introduce train-test mismatch

Variance Reduction methods

- Cumbersome
- Not clear if their results are worth added complexity
- Always unbiased
- Allow you to tune baseline to minimize variance
- **Random search on steroids**

Still ongoing research topic, many other approaches not covered

- Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud, *Backpropagation through the void: Optimizing control variates for black-box gradient estimation*, International Conference on Learning Representations, 2018.
- Shixiang Gu, Sergey Levine, Ilya Sutskever, and Andriy Mnih, *Muprop: Unbiased backpropagation for stochastic neural networks*, International Conference on Learning Representations, 2016.
- Eric Jang, Shixiang Gu, and Ben Poole, *Categorical reparameterization with gumbel-softmax*, International Conference on Learning Representations, 2017.
- Andriy Mnih and Karol Gregor, *Neural variational inference and learning in belief networks*, Proceedings of the 31st International Conference on Machine Learning (ICML), 2014.

# References II

Deep|Bayes


- Chris J. Maddison, Andriy Mnih, and Yee Whye Teh, *The concrete distribution: A continuous relaxation of discrete random variables*, International Conference on Learning Representations, 2017.
- Ben Recht, *arg min blog: The policy of truth*, `http://www.argmin.net/2018/02/20/reinforce/`, 2018.
- George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein, *Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models*, Advances in Neural Information Processing Systems 30 (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), Curran Associates, Inc., 2017, pp. 2627–2636.