Boulder

Generative Networks; Variational Auto-Encoders

YouTube Playlist

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu

Auto-Encoding Variational Bayes

YouTube Video

$X = \{x^i\}_{i=1}^N \rightarrow$ dataset (N i.i.d samples of some continuous or discrete random variable x)

$z \rightarrow$ unobserved (latent) continuous random variable

$z^i \rightarrow$ generated from $\underbrace{p_{\theta^*}(z)}_{\text{prior distribution}}$

$x^i \rightarrow$ generated from some conditional distribution $\underbrace{p_{\theta^*}(x|z)}_{\text{likelihood (generative model or probabilistic decoder)}}$

$p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz \rightarrow$ marginal likelihood (interactable)

$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} \rightarrow$ posterior (interactable)

Image denoising, inpainting, and super-resolution

$\underbrace{q_\phi(z|x)}_{\text{recognition model (probabilistic encoder)}} \rightarrow$ an approximation to the posterior distribution

$$\log p_\theta(x^1, \ldots, x^N) = \sum_{i=1}^N \log p_\theta(x^i)$$

$$\log p_\theta(x^i) = \underbrace{KL(q_\phi(z|x^i)||p_\theta(z|x^i))}_{\geq 0} + \underbrace{\mathcal{L}(\theta, \phi; x^i)}_{\mathbb{E}_{q_\phi(z|x^i)}[-\log q_\phi(z|x^i) + \log p_\theta(x^i, z)]}$$

reparameterization trick

$z \sim q_\phi(z|x)$

$z = \underbrace{g_\phi(\epsilon, x)}_{\text{differentiable transformation}}$ with $\overbrace{\epsilon \sim p(\epsilon)}^{\text{auxiliary noise variable}}$

$$\mathcal{L}(\theta, \phi; x^i) \approx \frac{1}{L}\sum_{\ell=1}^L \left[\log p_\theta(x^i, z^{i,\ell}) - \log q_\phi(z^{i,\ell}|x^i)\right]$$

$$z^{i,\ell} = g_\phi(\epsilon^\ell, x^i), \epsilon^\ell \sim p(\epsilon)$$

Variational Auto-Encoder

$p_\theta(z) = \mathcal{N}(z; 0, I)$

$\log q_\phi(z|x^i) = \log \mathcal{N}(z; \mu(x^i), \sigma^2(x^i)I)$

$g_\phi(\epsilon, x) = \mu(x) + \sigma(x)\epsilon$

$\epsilon \sim \mathcal{N}(\epsilon; 0, I)$

$\log p_\theta(x|z)$ depends on the type of data (e.g., Gaussian or Bernoulli)

Kingma, Diederik P., and Max Welling. "Auto-encoding variational Bayes." *arXiv preprint arXiv:1312.6114* (2013).

Stochastic Backpropagation and Approximate Inference in Deep Generative Models

Boulder

Deep Latent Gaussian Models (DLGMs)

$\boldsymbol{\xi}_l \sim \mathcal{N}(\boldsymbol{\xi}_l|\mathbf{0},\mathbf{I}), \quad l = 1,\ldots,L \rightarrow$ mutually independent Gaussian variables

$\mathbf{h}_L = \mathbf{G}_L\boldsymbol{\xi}_L,$

$\mathbf{h}_l = T_l(\mathbf{h}_{l+1}) + \mathbf{G}_l\boldsymbol{\xi}_l, \quad l = 1\ldots L-1$

$\mathbf{v} \sim \pi(\mathbf{v}|T_0(\mathbf{h}_1)),$

$\mathbf{G}_l \rightarrow$ matrices

$T_l \rightarrow$ MLPs (multi-layer perceptrons)

$\pi \rightarrow$ any appropriate distribution

$\boldsymbol{\theta}^g \rightarrow$ parameters of the MLPs and the matrices

$p(\boldsymbol{\theta}^g) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0},\kappa\mathbf{I}) \rightarrow$ Gaussian prior

Scalable Inference in DLGMs

$\mathbf{V} \rightarrow$ full dataset of size $N \times D$

$\mathbf{v}_n = [v_{n1},\ldots,v_{nD}]^\top \rightarrow$ observations

$$\mathcal{L}(\mathbf{V}) = -\log p(\mathbf{V}) = -\log \int p(\mathbf{V}|\boldsymbol{\xi},\boldsymbol{\theta}^g)p(\boldsymbol{\xi},\boldsymbol{\theta}^g)d\boldsymbol{\xi}$$

$$= -\log \int \frac{q(\boldsymbol{\xi})}{q(\boldsymbol{\xi})}p(\mathbf{V}|\boldsymbol{\xi},\boldsymbol{\theta}^g)p(\boldsymbol{\xi},\boldsymbol{\theta}^g)d\boldsymbol{\xi}$$

$$\leq \mathcal{F}(\mathbf{V}) = D_{KL}[q(\boldsymbol{\xi})\|p(\boldsymbol{\xi})] - \mathbb{E}_q\left[\log p(\mathbf{V}|\boldsymbol{\xi},\boldsymbol{\theta}^g)p(\boldsymbol{\theta}^g)\right]$$

$$q(\boldsymbol{\xi}|\mathbf{V},\boldsymbol{\theta}^r) = \prod_{n=1}^{N}\prod_{l=1}^{L}\mathcal{N}\left(\boldsymbol{\xi}_{n,l}|\boldsymbol{\mu}_l(\mathbf{v}_n),\mathbf{C}_l(\mathbf{v}_n)\right) \rightarrow \text{recognition model}$$

$$D_{KL}[\mathcal{N}(\boldsymbol{\mu},\mathbf{C})\|\mathcal{N}(\mathbf{0},\mathbf{I})] = \frac{1}{2}\left[\mathrm{Tr}(\mathbf{C}) - \log|\mathbf{C}| + \boldsymbol{\mu}^\top\boldsymbol{\mu} - D\right]$$

$$\mathcal{F}(\mathbf{V}) = -\sum_n \mathbb{E}_q\left[\log p(\mathbf{v}_n|\mathbf{h}(\boldsymbol{\xi}_n))\right] + \frac{1}{2\kappa}\|\boldsymbol{\theta}^g\|^2$$

$$+\frac{1}{2}\sum_{n,l}\left[\|\boldsymbol{\mu}_{n,l}\|^2 + \mathrm{Tr}(\mathbf{C}_{n,l}) - \log|\mathbf{C}_{n,l}| - 1\right]$$

$$\nabla_{\theta_j^g}\mathcal{F}(\mathbf{V}) = -\mathbb{E}_q\left[\nabla_{\theta_j^g}\log p(\mathbf{V}|\mathbf{h})\right] + \frac{1}{\kappa}\theta_j^g$$

$$\nabla_{\boldsymbol{\theta}^r}\mathcal{F}(\mathbf{v}) = ?$$

Stochastic Backpropagation

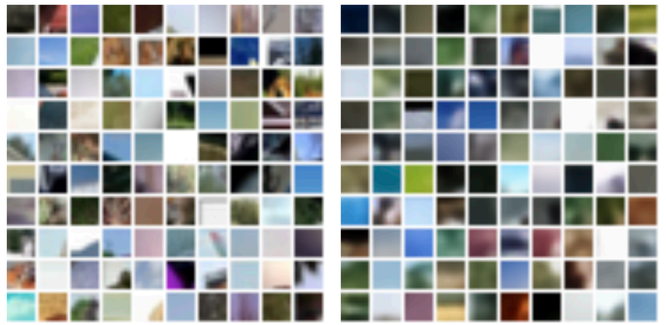
$\nabla_\theta \mathbb{E}_{q_\theta}\left[f(\boldsymbol{\xi})\right]$

$f(\boldsymbol{\xi}) = \log p(\mathbf{v}|\mathbf{h}(\boldsymbol{\xi}))$

$$\nabla_{\mathbf{R}}\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}\left[f(\boldsymbol{\xi})\right] = \nabla_{\mathbf{R}}\mathbb{E}_{\mathcal{N}(0,I)}\left[f(\boldsymbol{\mu}+\mathbf{R}\boldsymbol{\epsilon})\right] = \mathbb{E}_{\mathcal{N}(0,I)}\left[\boldsymbol{\epsilon}\mathbf{g}^\top\right]$$

$\mathbf{g} \rightarrow$ gradient of the function $f(\boldsymbol{\xi})$ $\qquad \mathbf{C} = \mathbf{R}\mathbf{R}^\top$

$$\nabla_{\boldsymbol{\theta}^r}\mathcal{F}(\mathbf{v}) = \nabla_{\boldsymbol{\mu}}\mathcal{F}(\mathbf{v})^\top\frac{\partial\boldsymbol{\mu}}{\partial\boldsymbol{\theta}^r} + \mathrm{Tr}\left(\nabla_{\mathbf{R}}\mathcal{F}(\mathbf{v})\frac{\partial\mathbf{R}}{\partial\boldsymbol{\theta}^r}\right)$$



(b) CIFAR

(c) Frey

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models." *International conference on machine learning*. PMLR, 2014.

Categorical Reparameterization with Gumbel-Softmax

Boulder

Score Function-based Gradient Estimators (SF)

REINFORCE or likelihood ratio estimator

$\nabla_\theta p_\theta(z) = p_\theta(z) \nabla_\theta \log p_\theta(z)$

$\nabla_\theta \mathbb{E}_z [f(z)] = \mathbb{E}_z [f(z) \nabla_\theta \log p_\theta(z)]$

does not require backpropagating through f or the sample z

suffers from high variance

$b(z) \rightarrow$ control variate

$\mu_b = \mathbb{E}_z [b(z) \nabla_\theta \log p_\theta(z)] \rightarrow$ analytical expectation

$\nabla_\theta \mathbb{E}_z [f(z)] = \mathbb{E}_z [(f(z) - b(z)) \nabla_\theta \log p_\theta(z)] + \mu_b$

NVIL, DARN, MuProp, VIMCO

Gumbel-Softmax distribution

$z \rightarrow$ categorical variable with class probabilities $\pi = (\pi_1, \ldots, \pi_k)$

k-dimensional one-hot vectors lying on the corners of the $(k-1)$-dimensional simplex Δ^{k-1}

$\mathbb{E}_p[z] = [\pi_1, \ldots, \pi_k] \rightarrow$ element-wise mean

Gumbel-Max Trick

a simple and efficient way to draw samples z from a categorical distribution with class probabilities π

$$z = \texttt{one\_hot}\left(\arg\max_i [g_i + \log \pi_i]\right)$$

$g_1, \ldots, g_k \rightarrow$ i.i.d samples drawn from Gumbel$(0, 1)$

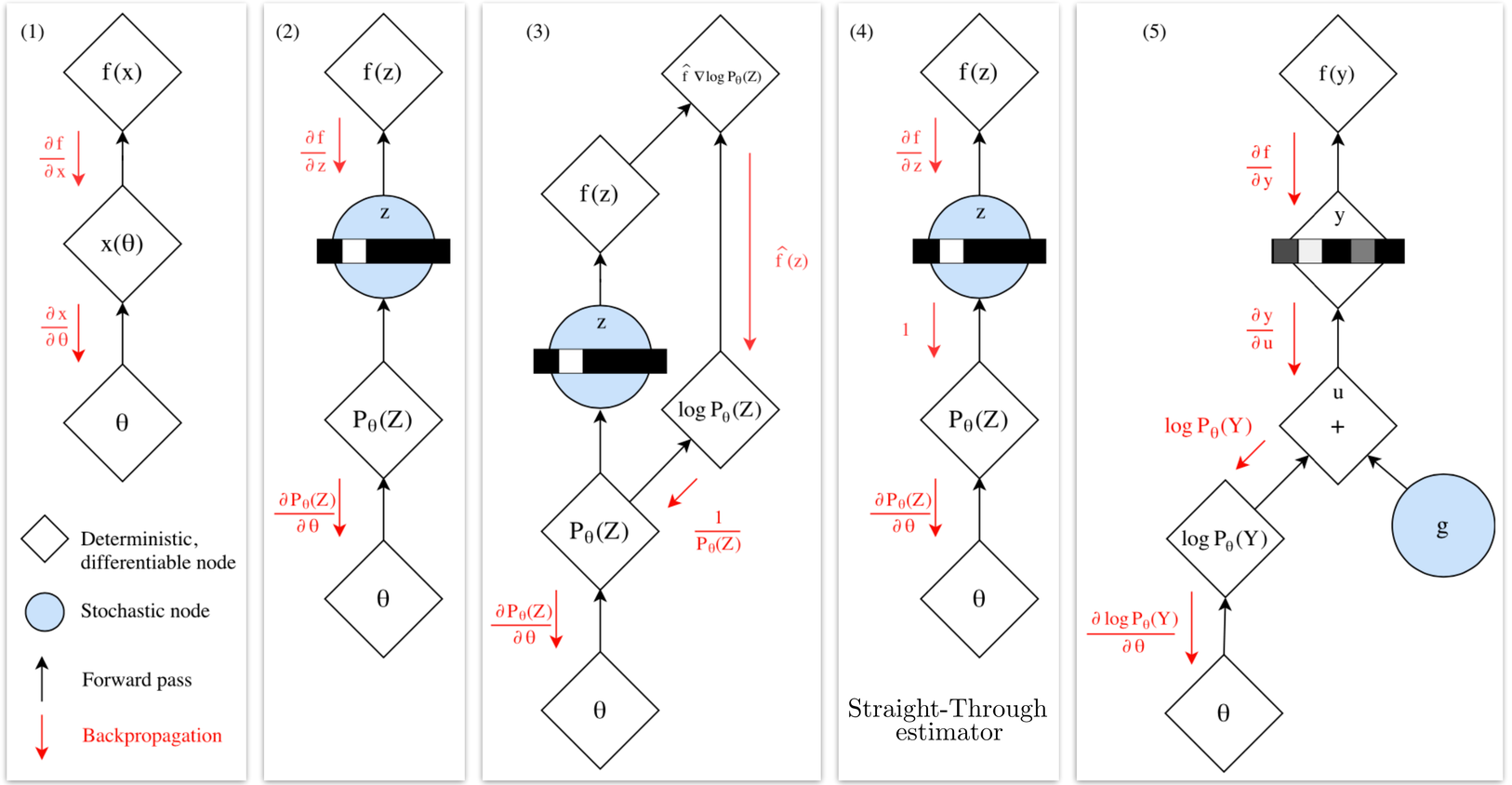
inverse transform sampling

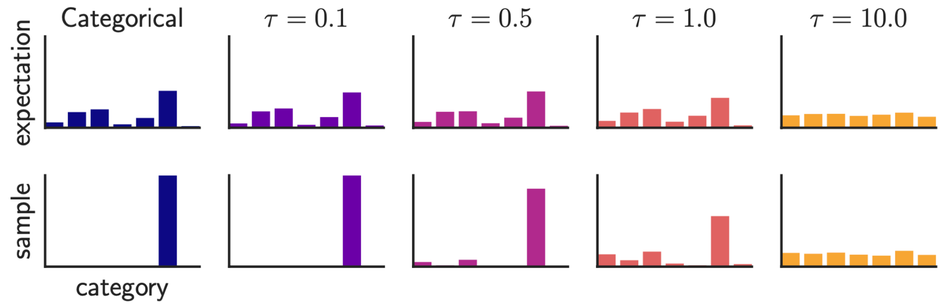
$u \sim \text{Uniform}(0, 1) \implies g = -\log(-\log(u))$

softmax: continuous, differentiable approximation to argmax

$$y_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{k} \exp((\log(\pi_j) + g_j)/\tau)} \quad \text{for } i = 1, \ldots, k.$$

$$p_{\pi,\tau}(y_1, \ldots, y_k) = \Gamma(k) \tau^{k-1} \left(\sum_{i=1}^{k} \pi_i / y_i^\tau\right)^{-k} \prod_{i=1}^{k} (\pi_i / y_i^{\tau+1})$$





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

Boulder

Questions?

YouTube Playlist