

Deep Generative Models

Lecture 7

Roman Isachenko

Moscow Institute of Physics and Technology

2022

Recap of previous lecture

Gaussian autoregressive flow (MAF)

$$\mathbf{x} = g(\mathbf{z}, \theta) \quad \Rightarrow \quad x_i = \sigma_i(\mathbf{x}_{1:i-1}) \cdot z_i + \mu_i(\mathbf{x}_{1:i-1}).$$

$$\mathbf{z} = f(\mathbf{x}, \theta) \quad \Rightarrow \quad z_i = (x_i - \mu_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\sigma_i(\mathbf{x}_{1:i-1})}.$$

Generation function $g(\mathbf{z}, \theta)$ is **sequential**. Inference function $f(\mathbf{x}, \theta)$ is **not sequential**.

Inverse autoregressive flow (IAF)

$$\mathbf{x} = g(\mathbf{z}, \theta) \quad \Rightarrow \quad x_i = \tilde{\sigma}_i(\mathbf{z}_{1:i-1}) \cdot z_i + \tilde{\mu}_i(\mathbf{z}_{1:i-1})$$

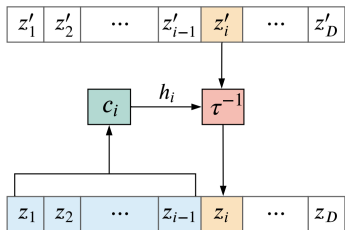
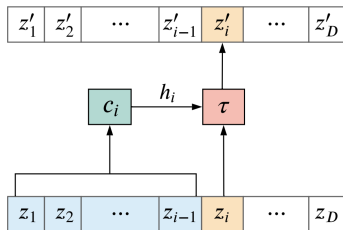
$$\mathbf{z} = f(\mathbf{x}, \theta) \quad \Rightarrow \quad z_i = (x_i - \tilde{\mu}_i(\mathbf{z}_{1:i-1})) \cdot \frac{1}{\tilde{\sigma}_i(\mathbf{z}_{1:i-1})}.$$

Papamakarios G., Pavlakou T., Murray I. *Masked Autoregressive Flow for Density Estimation*, 2017

Kingma D. P. et al. *Improving Variational Inference with Inverse Autoregressive Flow*, 2016

Recap of previous lecture

Autoregressive flows



RealNVP: Affine coupling law

$$\begin{cases} \mathbf{z}_{1:d} = \mathbf{x}_{1:d}; \\ \mathbf{z}_{d:m} = \tau(\mathbf{x}_{d:m}, c(\mathbf{x}_{1:d})); \end{cases} \Leftrightarrow \begin{cases} \mathbf{x}_{1:d} = \mathbf{z}_{1:d}; \\ \mathbf{x}_{d:m} = \tau^{-1}(\mathbf{z}_{d:m}, c(\mathbf{z}_{1:d})). \end{cases}$$

Dinh L., Krueger D., Bengio Y. *NICE: Non-linear Independent Components Estimation*, 2014

Dinh L., Sohl-Dickstein J., Bengio S. *Density estimation using Real NVP*, 2016

Outline

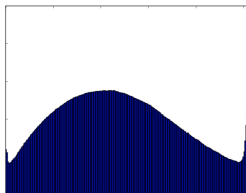
1. Data dequantization
2. ELBO surgery
3. VAE prior
4. VAE posterior

Outline

1. Data dequantization
2. ELBO surgery
3. VAE prior
4. VAE posterior

PixelCNN++

CIFAR-10 pixel values distribution



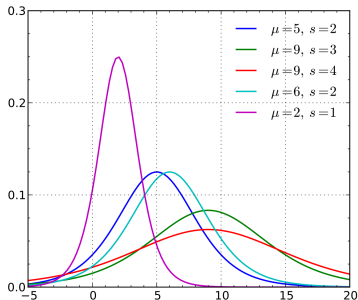
- ▶ Standard PixelCNN outputs softmax probabilities for values $\{0, 255\}$ (256 outputs feature maps).
- ▶ Categorical distribution do not know anything about numerical relationships (220 is close to 221 and far from 15).
- ▶ If pixel value is not presented in the training dataset, it won't be predicted.
- ▶ (Look at the edges of the distributions: they have higher probability mass).

Salimans T. et al. *PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications*, 2017

Mixture of logistic distributions

$$p(x|\mu, s) = \frac{\exp^{-(x-\mu)/s}}{s(1 + \exp^{-(x-\mu)/s})^2};$$

$$p(x|\boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\pi}) = \sum_{k=1}^K \pi_k p(x|\mu_k, s_k);$$



To adopt probability calculation to discrete values:

$$P_d(x|\boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\pi}) = P(x + 0.5|\boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\pi}) - P(x - 0.5|\boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\pi})$$

For the edge case of 0, replace $x - 0.5$ by $-\infty$, and for 255 replace $x + 0.5$ by $+\infty$.

Dequantization

- ▶ Images are discrete data, pixels lie in the $\{0, 255\}$ integer domain (the model is $P(\mathbf{x}|\theta) = \text{Categorical}(\pi(\theta))$).
- ▶ Flow is a continuous model (it works with continuous data \mathbf{x}).

By fitting a continuous density model to discrete data, one can produce a degenerate solution with all probability mass on discrete values.

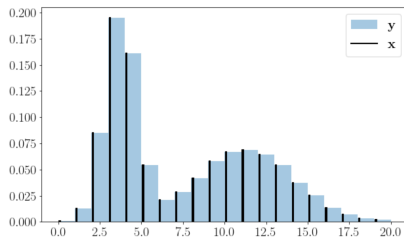
How to convert a discrete data distribution to a continuous one?

Uniform dequantization

$$\mathbf{x} \sim \text{Categorical}(\pi)$$

$$\mathbf{u} \sim U[0, 1]$$

$$\mathbf{y} = \mathbf{x} + \mathbf{u} \sim \text{Continuous}$$



Uniform dequantization

Theorem

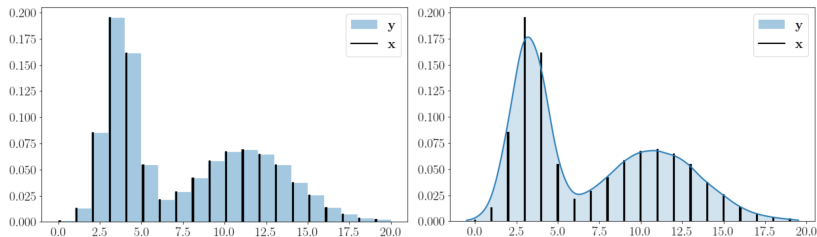
Fitting continuous model $p(\mathbf{y}|\boldsymbol{\theta})$ on uniformly dequantized data $\mathbf{y} = \mathbf{x} + \mathbf{u}$, $\mathbf{u} \sim U[0, 1]$ is equivalent to maximization of a lower bound on log-likelihood for a discrete model:

$$P(\mathbf{x}|\boldsymbol{\theta}) = \int_{U[0,1]} p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u}$$

Proof

$$\begin{aligned}\mathbb{E}_{\pi} \log p(\mathbf{y}|\boldsymbol{\theta}) &= \int \pi(\mathbf{y}) \log p(\mathbf{y}|\boldsymbol{\theta}) d\mathbf{y} = \\ &= \sum \pi(\mathbf{x}) \int_{U[0,1]} \log p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u} \leq \\ &\leq \sum \pi(\mathbf{x}) \log \int_{U[0,1]} p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u} = \\ &= \sum \pi(\mathbf{x}) \log P(\mathbf{x}|\boldsymbol{\theta}) = \mathbb{E}_{\pi} \log P(\mathbf{x}|\boldsymbol{\theta}).\end{aligned}$$

Variational dequantization



- ▶ $p(\mathbf{y}|\boldsymbol{\theta})$ assign uniform density to unit hypercubes $\mathbf{x} + U[0, 1]$ (left fig).
- ▶ Neural network density models are smooth function approximators (right fig).
- ▶ Smooth dequantization is more natural.

How to perform the smooth dequantization?

Variational dequantization

Introduce variational dequantization noise distribution $q(\mathbf{u}|\mathbf{x})$ and treat it as an approximate posterior.

Variational lower bound

$$\begin{aligned}\log P(\mathbf{x}|\boldsymbol{\theta}) &= \left[\log \int q(\mathbf{u}|\mathbf{x}) \frac{p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta})}{q(\mathbf{u}|\mathbf{x})} d\mathbf{u} \right] \geq \\ &\geq \int q(\mathbf{u}|\mathbf{x}) \log \frac{p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta})}{q(\mathbf{u}|\mathbf{x})} d\mathbf{u} = \mathcal{L}(q, \boldsymbol{\theta}).\end{aligned}$$

Uniform dequantization bound

$$\log P(\mathbf{x}|\boldsymbol{\theta}) = \log \int_{U[0,1]} p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u} \geq \int_{U[0,1]} \log p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u}.$$

Uniform dequantization is a special case of variational dequantization ($q(\mathbf{u}|\mathbf{x}) = U[0, 1]$).

Variational lower bound

$$\mathcal{L}(q, \theta) = \int q(\mathbf{u}|\mathbf{x}) \log \frac{p(\mathbf{x} + \mathbf{u}|\theta)}{q(\mathbf{u}|\mathbf{x})} d\mathbf{u}.$$

Let $\mathbf{u} = g(\epsilon, \mathbf{x}, \lambda)$ is a flow model with base distribution $\epsilon \sim p(\epsilon) = \mathcal{N}(0, \mathbf{I})$:

$$q(\mathbf{u}|\mathbf{x}) = p(g^{-1}(\mathbf{u}, \mathbf{x}, \lambda)) \cdot \left| \det \frac{\partial g^{-1}(\mathbf{u}, \mathbf{x}, \lambda)}{\partial \mathbf{u}} \right|.$$

Flow-based variational dequantization

$$\log P(\mathbf{x}|\theta) \geq \mathcal{L}(\lambda, \theta) = \int p(\epsilon) \log \left(\frac{p(\mathbf{x} + g(\epsilon, \mathbf{x}, \lambda)|\theta)}{p(\epsilon) \cdot |\det \mathbf{J}_g|^{-1}} \right) d\epsilon.$$

If $p(\mathbf{x} + \mathbf{u}|\theta)$ is also a flow model, it is straightforward to calculate stochastic gradient of this ELBO.

Flow-based variational dequantization

$$\log P(\mathbf{x}|\theta) \geq \int p(\epsilon) \log \left(\frac{p(\mathbf{x} + g(\epsilon, \mathbf{x}, \lambda))}{p(\epsilon) \cdot |\det \mathbf{J}_g|^{-1}} \right) d\epsilon.$$

Table 1. Unconditional image modeling results in bits/dim

Model family	Model	CIFAR10	ImageNet 32x32	ImageNet 64x64
Non-autoregressive	RealNVP (Dinh et al., 2016)	3.49	4.28	–
	Glow (Kingma & Dhariwal, 2018)	3.35	4.09	3.81
	IAF-VAE (Kingma et al., 2016)	3.11	–	–
	Flow++ (ours)	3.08	3.86	3.69
Autoregressive	Multiscale PixelCNN (Reed et al., 2017)	–	3.95	3.70
	PixelCNN (van den Oord et al., 2016b)	3.14	–	–
	PixelRNN (van den Oord et al., 2016b)	3.00	3.86	3.63
	Gated PixelCNN (van den Oord et al., 2016c)	3.03	3.83	3.57
	PixelCNN++ (Salimans et al., 2017)	2.92	–	–
	Image Transformer (Parmar et al., 2018)	2.90	3.77	–
	PixelSNAIL (Chen et al., 2017)	2.85	3.80	3.52

Outline

1. Data dequantization
2. ELBO surgery
3. VAE prior
4. VAE posterior

Likelihood-based models

Exact likelihood evaluation

- ▶ Autoregressive models (WaveNet, PixelCNN, PixelCNN++);
- ▶ Flow models (RealNVP, IAF, Glow).

Approximate likelihood evaluation

- ▶ Latent variable models (VAE).

What are the pros and cons of each of them?

VAE limitations

- ▶ Poor generative distribution (decoder)

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\sigma}_{\boldsymbol{\theta}}^2(\mathbf{z})) \quad \text{or} \quad = \text{Softmax}(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{z})).$$

- ▶ Loose lower bound

$$\log p(\mathbf{x}|\boldsymbol{\theta}) - \mathcal{L}(q, \boldsymbol{\theta}) = (?).$$

- ▶ **Poor prior distribution**

$$p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I}).$$

- ▶ Poor variational posterior distribution (encoder)

$$q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2(\mathbf{x})).$$

ELBO surgery

$$\frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(q, \theta) = \frac{1}{n} \sum_{i=1}^n [\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \theta) - KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z}))].$$

Theorem

$$\frac{1}{n} \sum_{i=1}^n KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z})) = KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) + \mathbb{I}_q[\mathbf{x}, \mathbf{z}],$$

- ▶ $q_{\text{agg}}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}_i)$ – **aggregated** posterior distribution.
- ▶ $\mathbb{I}_q[\mathbf{x}, \mathbf{z}]$ – mutual information between \mathbf{x} and \mathbf{z} under empirical data distribution and distribution $q(\mathbf{z}|\mathbf{x})$.
- ▶ First term pushes $q_{\text{agg}}(\mathbf{z})$ towards the prior $p(\mathbf{z})$.
- ▶ Second term reduces the amount of information about \mathbf{x} stored in \mathbf{z} .

ELBO surgery

Theorem

$$\frac{1}{n} \sum_{i=1}^n KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z})) = KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) + \mathbb{I}_q[\mathbf{x}, \mathbf{z}].$$

Proof

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z})) &= \frac{1}{n} \sum_{i=1}^n \int q(\mathbf{z}|\mathbf{x}_i) \log \frac{q(\mathbf{z}|\mathbf{x}_i)}{p(\mathbf{z})} d\mathbf{z} = \\ &= \frac{1}{n} \sum_{i=1}^n \int q(\mathbf{z}|\mathbf{x}_i) \log \frac{q_{\text{agg}}(\mathbf{z})q(\mathbf{z}|\mathbf{x}_i)}{p(\mathbf{z})q_{\text{agg}}(\mathbf{z})} d\mathbf{z} = \int \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}_i) \log \frac{q_{\text{agg}}(\mathbf{z})}{p(\mathbf{z})} d\mathbf{z} + \\ &+ \frac{1}{n} \sum_{i=1}^n \int q(\mathbf{z}|\mathbf{x}_i) \log \frac{q(\mathbf{z}|\mathbf{x}_i)}{q_{\text{agg}}(\mathbf{z})} d\mathbf{z} = KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) + \frac{1}{n} \sum_{i=1}^n KL(q(\mathbf{z}|\mathbf{x}_i)||q_{\text{agg}}(\mathbf{z})) \end{aligned}$$

Without proof:

$$\mathbb{I}_q[\mathbf{x}, \mathbf{z}] = \frac{1}{n} \sum_{i=1}^n KL(q(\mathbf{z}|\mathbf{x}_i)||q_{\text{agg}}(\mathbf{z})) \in [0, \log n].$$

ELBO surgery

ELBO revisiting

$$\begin{aligned}\frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(q, \theta) &= \frac{1}{n} \sum_{i=1}^n [\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \theta) - KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z}))] = \\ &= \underbrace{\frac{1}{n} \sum_{i=1}^n \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \theta)}_{\text{Reconstruction loss}} - \underbrace{\mathbb{I}_q[\mathbf{x}, \mathbf{z}]}_{\text{MI}} - \underbrace{KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z}))}_{\text{Marginal KL}}\end{aligned}$$

Prior distribution $p(\mathbf{z})$ is only in the last term.

Optimal VAE prior

$$KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) = 0 \quad \Leftrightarrow \quad p(\mathbf{z}) = q_{\text{agg}}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}_i).$$

The optimal prior $p(\mathbf{z})$ is the aggregated posterior $q_{\text{agg}}(\mathbf{z})$.

Hoffman M. D., Johnson M. J. *ELBO surgery: yet another way to carve up the variational evidence lower bound*, 2016

Outline

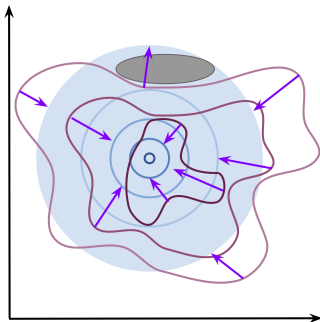
1. Data dequantization
2. ELBO surgery
3. VAE prior
4. VAE posterior

Optimal VAE prior

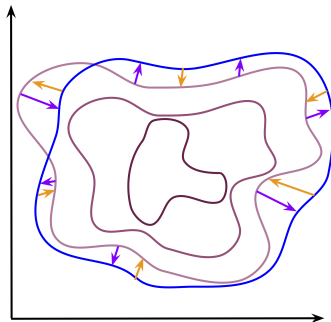
How to choose the optimal $p(\mathbf{z})$?

- ▶ Standard Gaussian $p(\mathbf{z}) = \mathcal{N}(0, I) \Rightarrow$ over-regularization;
- ▶ $p(\mathbf{z}) = q_{\text{agg}}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}_i) \Rightarrow$ overfitting and highly expensive.

Non learnable prior $p(\mathbf{z})$



Learnable prior $p(\mathbf{z}|\lambda)$



Flows-based VAE prior

Flow model in latent space

$$\log p(\mathbf{z}|\boldsymbol{\lambda}) = \log p(\mathbf{z}^*) + \log \left| \det \left(\frac{d\mathbf{z}^*}{d\mathbf{z}} \right) \right| = \log p(g(\mathbf{z}, \boldsymbol{\lambda})) + \log |\det(\mathbf{J}_g)|$$

$$\mathbf{z} = f(\mathbf{z}^*, \boldsymbol{\lambda}) = g^{-1}(\mathbf{z}^*, \boldsymbol{\lambda})$$

- ▶ RealNVP flow.
- ▶ Autoregressive flow (MAF).

Why it is not a good idea to use IAF for VAE prior?

ELBO with flow-based VAE prior

$$\begin{aligned} \mathcal{L}(\phi, \theta) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} [\log p(\mathbf{x}|\mathbf{z}, \theta) + \log p(\mathbf{z}|\boldsymbol{\lambda}) - \log q(\mathbf{z}|\mathbf{x}, \phi)] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \left[\log p(\mathbf{x}|\mathbf{z}, \theta) + \underbrace{\left(\log p(g(\mathbf{z}, \boldsymbol{\lambda})) + \log |\det(\mathbf{J}_g)| \right)}_{\text{flow-based prior}} - \log q(\mathbf{z}|\mathbf{x}, \phi) \right] \end{aligned}$$

VAE limitations

- ▶ Poor generative distribution (decoder)

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\sigma}_{\boldsymbol{\theta}}^2(\mathbf{z})) \quad \text{or} \quad = \text{Softmax}(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{z})).$$

- ▶ Loose lower bound

$$\log p(\mathbf{x}|\boldsymbol{\theta}) - \mathcal{L}(q, \boldsymbol{\theta}) = (?).$$

- ▶ Poor prior distribution

$$p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I}).$$

- ▶ **Poor variational posterior distribution (encoder)**

$$q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2(\mathbf{x})).$$

Outline

1. Data dequantization
2. ELBO surgery
3. VAE prior
4. VAE posterior

Variational posterior

ELBO

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

- ▶ In E-step of EM-algorithm we wish $KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) = 0$.
(In this case the lower bound is tight $\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta})$).
- ▶ Normal variational distribution $q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi^2(\mathbf{x}))$ is poor (e.g. has only one mode).
- ▶ Flows models convert a simple base distribution to a complex one using invertible transformation with simple Jacobian. How to use flows in VAE posterior?

Flows in VAE posterior

Apply a sequence of transformations to the random variable

$$\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\mu_{\phi}(\mathbf{x}), \sigma_{\phi}^2(\mathbf{x})).$$

Let $q(\mathbf{z}|\mathbf{x}, \phi)$ (VAE encoder) be a base distribution for a flow model.

Flow model in latent space

$$\log q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda) = \log q(\mathbf{z}|\mathbf{x}, \phi) + \log \left| \det \left(\frac{\partial g(\mathbf{z}, \lambda)}{\partial \mathbf{z}} \right) \right|$$

$$\mathbf{z}^* = g(\mathbf{z}, \lambda) = f^{-1}(\mathbf{z}, \lambda)$$

Here $g(\mathbf{z}, \lambda)$ is a flow model (e.g. stack of planar/coupling/AR layers) parameterized by λ .

Let use $q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)$ as a variational distribution. Here ϕ – encoder parameters, λ – flow parameters.

Flows-based VAE posterior

- ▶ Encoder outputs base distribution $q(\mathbf{z}|\mathbf{x}, \phi)$.
- ▶ Flow model $\mathbf{z}^* = g(\mathbf{z}, \lambda)$ transforms the base distribution $q(\mathbf{z}|\mathbf{x}, \phi)$ to the distribution $q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)$.
- ▶ Distribution $q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)$ is used as a variational distribution for ELBO maximization.

Flow model in latent space

$$\log q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda) = \log q(\mathbf{z}|\mathbf{x}, \phi) + \log \left| \det \left(\frac{\partial g(\mathbf{z}, \lambda)}{\partial \mathbf{z}} \right) \right|$$

ELBO with flow-based VAE posterior

$$\begin{aligned} \mathcal{L}(\phi, \theta, \lambda) &= \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)} [\log p(\mathbf{x}, \mathbf{z}^*|\theta) - \log q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)] \\ &= \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)} \log p(\mathbf{x}|\mathbf{z}^*, \theta) - KL(q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda) || p(\mathbf{z}^*)). \end{aligned}$$

The second term in ELBO is reverse KL divergence. Planar flows was originally proposed for variational inference in VAE.

Flows-based VAE posterior

Flow model in latent space

$$\log q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda) = \log q(\mathbf{z}|\mathbf{x}, \phi) + \log \left| \det \left(\frac{\partial g(\mathbf{z}, \lambda)}{\partial \mathbf{z}} \right) \right|$$

ELBO objective

$$\begin{aligned} \mathcal{L}(\phi, \theta, \lambda) &= \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)} [\log p(\mathbf{x}, \mathbf{z}^*|\theta) - \log q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)] = \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} [\log p(\mathbf{x}, \mathbf{z}^*|\theta) - \log q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)] \Big|_{\mathbf{z}^*=g(\mathbf{z}, \lambda)} = \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \left[\log p(\mathbf{x}, g(\mathbf{z}, \lambda)|\theta) - \log q(\mathbf{z}|\mathbf{x}, \phi) - \log |\det(\mathbf{J}_g)| \right]. \end{aligned}$$

- ▶ Obtain samples \mathbf{z} from the encoder $q(\mathbf{z}|\mathbf{x}, \phi)$.
- ▶ Apply flow model $\mathbf{z}^* = g(\mathbf{z}, \lambda)$.
- ▶ Compute likelihood for \mathbf{z}^* using the decoder, base distribution for \mathbf{z}^* and the Jacobian.

Inverse autoregressive flow (IAF)

$$\mathbf{x} = g(\mathbf{z}, \boldsymbol{\theta}) \quad \Rightarrow \quad x_i = \tilde{\sigma}_i(\mathbf{z}_{1:i-1}) \cdot z_i + \tilde{\mu}_i(\mathbf{z}_{1:i-1}).$$

$$\mathbf{z} = f(\mathbf{x}, \boldsymbol{\theta}) \quad \Rightarrow \quad z_i = (x_i - \tilde{\mu}_i(\mathbf{z}_{1:i-1})) \cdot \frac{1}{\tilde{\sigma}_i(\mathbf{z}_{1:i-1})}.$$

Reverse KL for flow model

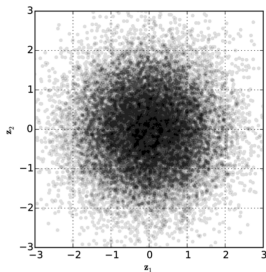
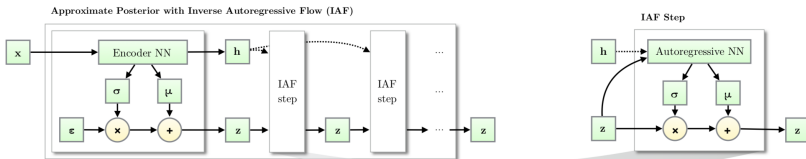
$$KL(p||\pi) = \mathbb{E}_{p(\mathbf{z})} \left[\log p(\mathbf{z}) - \log \left| \det \left(\frac{\partial g(\mathbf{z}, \boldsymbol{\theta})}{\partial \mathbf{z}} \right) \right| - \log \pi(g(\mathbf{z}, \boldsymbol{\theta})) \right]$$

- ▶ We don't need to think about computing the function $f(\mathbf{x}, \boldsymbol{\theta})$.
- ▶ Inverse autoregressive flow is a natural choice for using flows in VAE:

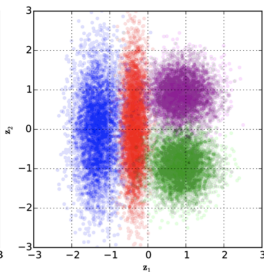
$$\mathbf{z} = \boldsymbol{\sigma}(\mathbf{x}) \odot \boldsymbol{\epsilon} + \boldsymbol{\mu}(\mathbf{x}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1); \quad \sim q(\mathbf{z}|\mathbf{x}, \phi).$$

$$\mathbf{z}_k = \tilde{\boldsymbol{\sigma}}_k(\mathbf{z}_{k-1}) \odot \mathbf{z}_{k-1} + \tilde{\boldsymbol{\mu}}_k(\mathbf{z}_{k-1}), \quad k \geq 1; \quad \sim q_k(\mathbf{z}_k|\mathbf{x}, \phi, \{\boldsymbol{\lambda}_j\}_{j=1}^k).$$

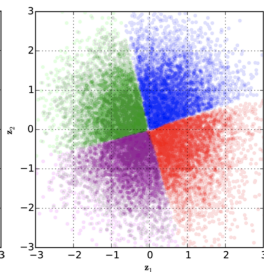
Inverse autoregressive flow (IAF)



(a) Prior distribution



(b) Posteriors in standard VAE



(c) Posteriors in VAE with IAF

Kingma D. P. et al. *Improving Variational Inference with Inverse Autoregressive Flow*, 2016

Flows-based VAE prior vs posterior

Theorem

VAE with the flow-based prior for latent code \mathbf{z} is equivalent to VAE with flow-based posterior for latent code \mathbf{z} .

Proof

$$\begin{aligned}\mathcal{L}(\phi, \theta, \lambda) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \log p(\mathbf{x}|\mathbf{z}, \theta) - \underbrace{KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}|\lambda))}_{\text{flow-based prior}} \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \log p(\mathbf{x}|\mathbf{z}, \theta) - \underbrace{KL(q(\mathbf{z}|\mathbf{x}, \phi, \lambda) || p(\mathbf{z}))}_{\text{flow-based posterior}}\end{aligned}$$

(Here we use Flow KL duality theorem from Lecture 5)

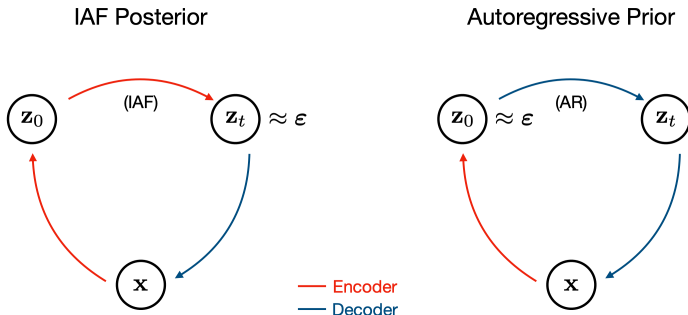
Flows in VAE posterior

$$\mathcal{L}(\phi, \theta, \lambda) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \left[\log p(\mathbf{x}, g(\mathbf{z}, \lambda) | \theta) - \log q(\mathbf{z}|\mathbf{x}, \phi) - \log |\det(\mathbf{J}_g)| \right].$$

Flows-based VAE prior vs posterior

- ▶ IAF posterior decoder path: $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$, $\mathbf{z} \sim p(\mathbf{z})$.
- ▶ AF prior decoder path: $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$, $\mathbf{z} = f(\mathbf{z}^*, \boldsymbol{\lambda})$, $\epsilon \sim p(\mathbf{z}^*)$.

The AF prior and the IAF posterior have the same computation cost, so using the AF prior makes the model more expressive at no training time cost.



VAE limitations

- ▶ Poor generative distribution (decoder)

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\sigma}_{\boldsymbol{\theta}}^2(\mathbf{z})) \quad \text{or} \quad = \text{Softmax}(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{z})).$$

- ▶ Loose lower bound

$$\log p(\mathbf{x}|\boldsymbol{\theta}) - \mathcal{L}(q, \boldsymbol{\theta}) = (?).$$

- ▶ Poor prior distribution

$$p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I}).$$

- ▶ Poor variational posterior distribution (encoder)

$$q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2(\mathbf{x})).$$

Summary

- ▶ PixelCNN++ proposes to use discretized mixture of logistics for output distribution.
- ▶ Dequantization allows to fit discrete data using continuous model.
- ▶ Uniform dequantization is the simplest form of dequantization. Variational dequantization is a more natural type that was proposed in Flow++ model.
- ▶ The ELBO surgery reveals insights about a prior distribution in VAE. The optimal prior is the aggregated posterior.
- ▶ We could use flow-based prior in VAE (moreover, autoregressive).
- ▶ We could use flows to make variational posterior more expressive. This is equivalent to the flow-based prior.