Deep Generative Models

Lecture 8

Roman Isachenko

Ozon Masters

Spring, 2022

Recap of previous lecture

Theorem

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

ELBO surgery

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

Optimal prior

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

The optimal prior distribution $p(\mathbf{z})$ is aggregated posterior $q(\mathbf{z})$.

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

Recap of previous lecture

Optimal prior

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

The optimal prior distribution $p(\mathbf{z})$ is aggregated posterior $q(\mathbf{z})$.

VampPrior

$$p(\mathbf{z}|\boldsymbol{\lambda}) = \frac{1}{K}\sum_{k=1}^{K} q(\mathbf{z}|\mathbf{u}_k),$$

where $\boldsymbol{\lambda} = \{\mathbf{u}_1, \dots, \mathbf{u}_K\}$ are trainable pseudo-inputs.

Flow-based VAE prior

$$\log p(\mathbf{z}|\boldsymbol{\lambda}) = \log p(\boldsymbol{\epsilon}) + \log \det \left| \frac{d\boldsymbol{\epsilon}}{d\mathbf{z}} \right| = \log p(\boldsymbol{\epsilon}) + \log \det \left| \frac{\partial f(\mathbf{z}, \boldsymbol{\lambda})}{\partial \mathbf{z}} \right|$$

Tomczak J. M., Welling M. VAE with a VampPrior, 2017
Chen X. et al. Variational Lossy Autoencoder, 2016

Outline

Flows in VAE posterior

Apply a sequence of transformations to the random variable

$$\mathbf{z}_0 \sim q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}^2_\phi(\mathbf{x})).$$

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

Flow model in latent space

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

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

Here $f(\mathbf{z}, \boldsymbol{\lambda})$ is a flow model (e.g. stack of planar/coupling layers) parameterized by $\boldsymbol{\lambda}$.
Let use $q(\mathbf{z}^*|\mathbf{x}, \phi, \boldsymbol{\lambda})$ as a variational distribution. Here ϕ – encoder parameters, $\boldsymbol{\lambda}$ – flow parameters.

Rezende D. J., Mohamed S. Variational Inference with Normalizing Flows, 2015

Flows-based VAE posterior

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

Flow model in latent space

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

ELBO with flow-based VAE posterior

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

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

Rezende D. J., Mohamed S. Variational Inference with Normalizing Flows, 2015

Flows-based VAE posterior

Flow model in latent space

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

ELBO objective

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \phi, \boldsymbol{\lambda})} \bigl[\log p(\mathbf{x}, \mathbf{z}^*|\boldsymbol{\theta}) - \log q(\mathbf{z}^*|\mathbf{x}, \phi, \boldsymbol{\lambda})\bigr] =$$
$$= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \left[\log p(\mathbf{x}, \mathbf{z}^*|\boldsymbol{\theta}) - \log q(\mathbf{z}^*|\mathbf{x}, \phi, \boldsymbol{\lambda})\right]\Big|_{\mathbf{z}^* = f(\mathbf{z}, \boldsymbol{\lambda})} =$$
$$= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \left[\log p(\mathbf{x}, \mathbf{z}^*|\boldsymbol{\theta}) - \log q(\mathbf{z}|\mathbf{x}, \phi) + \log \left| \det \left(\frac{\partial f(\mathbf{z}, \boldsymbol{\lambda})}{\partial \mathbf{z}} \right) \right| \right].$$

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

Rezende D. J., Mohamed S. Variational Inference with Normalizing Flows, 2015

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

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

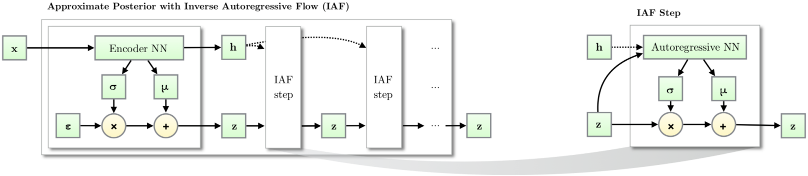
- We don't need to think about computing the function $f(\mathbf{x}, \boldsymbol{\theta})$.
- Inverse autoregressive flow is a natural choice for using flows in VAE:

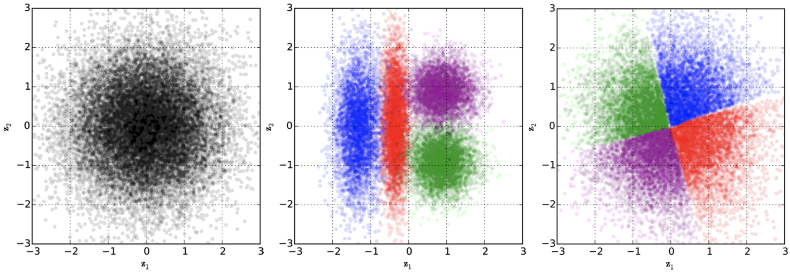
$$\mathbf{z} = \boldsymbol{\sigma}(\mathbf{x}) \odot \boldsymbol{\epsilon} + \boldsymbol{\mu}(\mathbf{x}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1); \quad \sim q(\mathbf{z}|\mathbf{x}, \phi).$$

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

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

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

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

(Here we use Flow KL duality theorem from Lecture 4)

Flows in VAE posterior

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \left[\log p(\mathbf{x}, \mathbf{z}^*|\boldsymbol{\theta}) - \log q(\mathbf{z}|\mathbf{x}, \phi) + \log \left| \det \left(\frac{\partial f(\mathbf{z}, \boldsymbol{\lambda})}{\partial \mathbf{z}} \right) \right| \right].$$

Chen X. et al. Variational Lossy Autoencoder, 2016

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: $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$, $\mathbf{z} = g(\boldsymbol{\epsilon}, \boldsymbol{\lambda})$, $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$.

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.

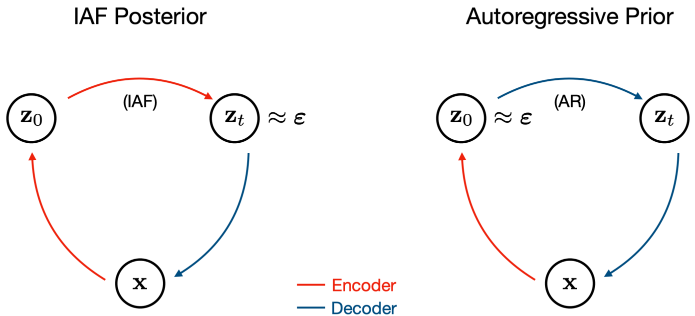


image credit: https://courses.cs.washington.edu/courses/cse599i/20au

VAE limitations

- Poor variational posterior distribution (encoder)

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

- Poor prior distribution

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

- Poor probabilistic model (decoder)

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

- Loose lower bound

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

Dequantization

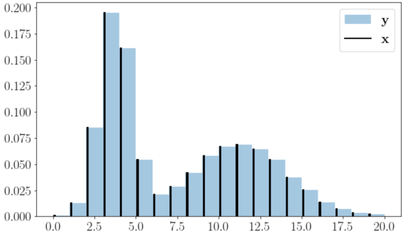
- Images are discrete data, pixels lie in the $\{0, 255\}$ integer domain (the model is $P(\mathbf{x}|\boldsymbol{\theta}) = \text{Categorical}(\boldsymbol{\pi}(\boldsymbol{\theta}))$).
- Flow is a continuous model (it works with continuous data $\mathbf{x}$).

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

$$\mathbf{x} \sim \text{Categorical}(\boldsymbol{\pi})$$
$$\mathbf{u} \sim U[0, 1]$$

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



Theis L., Oord A., Bethge M. A note on the evaluation of generative models. 2015

Uniform dequantization

Statement

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

$$\mathbb{E}_{\pi} \log p(\mathbf{y}|\boldsymbol{\theta}) = \int \pi(\mathbf{y}) \log p(\mathbf{y}|\boldsymbol{\theta}) d\mathbf{y} =$$

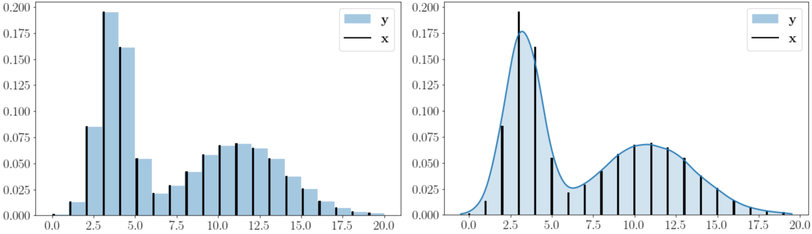
$$= \sum \pi(\mathbf{x}) \int_{U[0,1]} \log p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u} \leq$$

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

$$= \sum \pi(\mathbf{x}) \log P(\mathbf{x}|\boldsymbol{\theta}) = \mathbb{E}_{\pi} \log P(\mathbf{x}|\boldsymbol{\theta}).$$

Theis L., Oord A., Bethge M. A note on the evaluation of generative models. 2015

Variational dequantization



- $p(\mathbf{y}|\boldsymbol{\theta})$ assign unifrom 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?

Flow++

Variational dequantization

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

Variational lower bound

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

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 ($q(\mathbf{u}|\mathbf{x}) = U[0, 1]$).

Ho J. et al. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design, 2019

Flow++

Variational lower bound

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

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

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

Flow-based variational dequantization

$$\log P(\mathbf{x}|\boldsymbol{\theta}) \geq \mathcal{L}(\phi, \boldsymbol{\theta}) = \int p(\boldsymbol{\epsilon}) \log \left(\frac{p(\mathbf{x} + h(\boldsymbol{\epsilon}, \phi)|\boldsymbol{\theta})}{p(\boldsymbol{\epsilon}) \cdot \left| \det \frac{\partial h(\boldsymbol{\epsilon}, \phi)}{\partial \boldsymbol{\epsilon}} \right|^{-1}} \right) d\boldsymbol{\epsilon}.$$

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

Ho J. et al. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design, 2019

Flow++

Flow-based variational dequantization

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

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

Ho J. et al. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design, 2019

Disentangled representations

Representation learning is looking for an interpretable representation of the independent data generative factors.

Disentanglement informal definition

Every single latent unit are sensitive to changes in a single generative factor, while being invariant to changes in other factors.

Generative process

- $\pi(\mathbf{x}|\mathbf{v}, \mathbf{w}) = \text{Sim}(\mathbf{v}, \mathbf{w})$ – true world simulator;
- $\mathbf{v}$ – conditionally independent factors: $\pi(\mathbf{v}|\mathbf{x}) = \prod_{j=1}^{d} \pi(v_j|\mathbf{x})$;
- $\mathbf{w}$ – conditionally dependent factors.

Unsupervised generative model

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) \approx \pi(\mathbf{x}|\mathbf{v}, \mathbf{w}).$$

The latent factors $q(\mathbf{z}|\mathbf{x})$ capture the factors $\mathbf{v}$ in a disentangled manner. The conditionally dependent factors $\mathbf{w}$ remains entangled in a subset of $\mathbf{z}$ that is not used for representing $\mathbf{v}$.

Higgins I. et al. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, 2017

β-VAE

ELBO objective

$$\mathcal{L}(q, \boldsymbol{\theta}, \beta) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \beta \cdot KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})).$$

What do we get at $\beta = 1$?

Constrained optimization

$$\max_{q, \boldsymbol{\theta}} \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}), \quad \text{subject to } KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) < \epsilon.$$

Hypothesis

We are able to learn disentangled representations of the independent factors $\mathbf{v}$ by setting a stronger constraint with $\beta > 1$.

Note: It leads to poorer reconstructions and a loss of high frequency details.

Higgins I. et al. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, 2017

Summary

- We could use flows to make variational posterior more expressive. This is equivalent to the flow-based prior.
- 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.
- Disentanglement learning tries to make latent components more informative.