Deep Generative Models

Lecture 8

Roman Isachenko

AI Masters

Autumn, 2022

Recap of previous lecture

Images are discrete data, flow is a continuous model. We need to convert a discrete data distribution to a continuous one.

Uniform dequantization bound

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

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

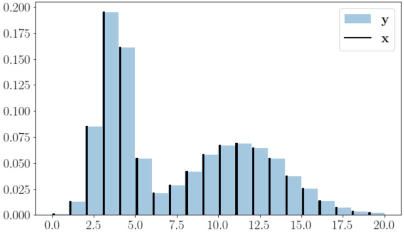
Variational dequantization bound

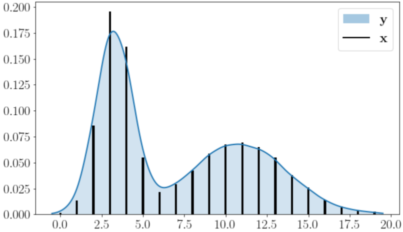
Introduce variational dequantization noise distribution $q(\mathbf{u}|\mathbf{x})$ and treat it as an approximate posterior.

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

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

Recap of previous lecture





Flow model for dequantization

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

Variational dequantization bound

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

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

Recap of previous lecture

Theorem

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

ELBO surgery

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

Optimal 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 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

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

The optimal prior 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

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

Tomczak J. M., Welling M. VAE with a VampPrior, 2017

Chen X. et al. Variational Lossy Autoencoder, 2016

Recap of previous lecture

Standart ELBO

$$p(\mathbf{x}|\boldsymbol{\theta}) \geq \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\boldsymbol{\phi})} \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})} \rightarrow \max_{\boldsymbol{\phi},\boldsymbol{\theta}}.$$

Expressive flow-based VAE posterior

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

ELBO with flow-based posterior

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x},\phi,\boldsymbol{\lambda})} \big[\log p(\mathbf{x}, \mathbf{z}^*|\boldsymbol{\theta}) - \log q(\mathbf{z}^*|\mathbf{x}, \phi, \boldsymbol{\lambda}) \big] =$$

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

- Obtain samples $\mathbf{z}$ from the encoder $q(\mathbf{z}|\mathbf{x}, \phi)$.
- Apply flow model $\mathbf{z}^* = g(\mathbf{z}, \boldsymbol{\lambda})$.
- 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

Recap of previous lecture

Expressive flow-based VAE posterior

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

Expressive flow-based VAE prior

$$\log p(\mathbf{z}|\boldsymbol{\lambda}) = \log p(\mathbf{z}^*) + \log \left| \det \left(\frac{d\mathbf{z}^*}{d\mathbf{z}} \right) \right|; \quad \mathbf{z} = f(\mathbf{z}^*, \boldsymbol{\lambda}) = g^{-1}(\mathbf{z}^*, \boldsymbol{\lambda})$$

Theorem

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

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

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

Chen X. et al. Variational Lossy Autoencoder, 2016

Outline

1. Disentanglement learning

2. Likelihood-free learning

3. Generative adversarial networks

Flows in 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.
- Here ϕ – encoder parameters, $\boldsymbol{\lambda}$ – flow parameters.

Flow model in latent space

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

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

ELBO with flow-based VAE posterior

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

The second term in ELBO is reverse KL divergence with respect to $q(\mathbf{z}^*|\mathbf{x}, \phi, \boldsymbol{\lambda})$.

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

Flow-based VAE posterior

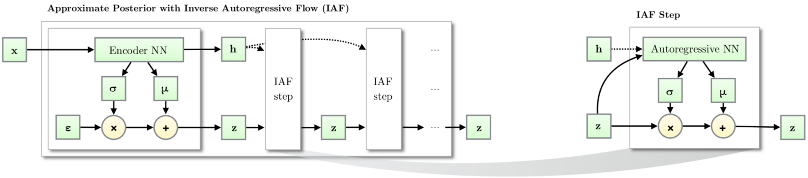
ELBO objective

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

$$= \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x},\phi,\boldsymbol{\lambda})} \Bigg[\log p(\mathbf{x}|\mathbf{z}^*, \boldsymbol{\theta}) + \log p(\mathbf{z}^*) -$$

$$- \Big(\log q(g(\mathbf{z}^*, \boldsymbol{\lambda})|\mathbf{x}, \phi) + \log |\det(\mathbf{J}_g)| \Big) \Bigg].$$

- RealNVP with coupling layers.
- Inverse autoregressive flow (slow $f(\mathbf{z}, \boldsymbol{\lambda})$, fast $g(\mathbf{z}^*, \boldsymbol{\lambda})$).
- Is it OK to use AF for VAE posterior?



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

Flows-based VAE prior vs posterior

Theorem

VAE with the flow-based prior $p(\mathbf{z}|\boldsymbol{\lambda})$ for latent code $\mathbf{z}^*$ is equivalent to VAE with flow-based posterior $q(\mathbf{z}|\mathbf{x}, \phi, \boldsymbol{\lambda})$ for latent code $\mathbf{z}$.

Proof

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

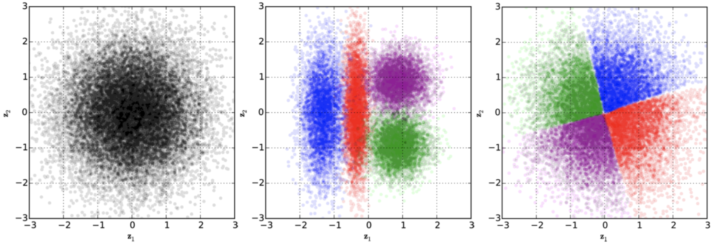
$$= \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x},\phi,\boldsymbol{\lambda})} \log p(\mathbf{x}|f(\mathbf{z}^*, \boldsymbol{\lambda}), \boldsymbol{\theta}) - \underbrace{KL(q(\mathbf{z}^*|\mathbf{x}, \phi, \boldsymbol{\lambda})||p(\mathbf{z}^*))}_{\text{flow-based posterior}}$$

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

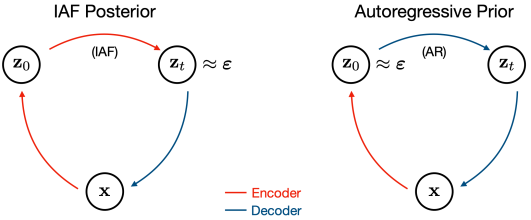
- IAF posterior decoder path: $\mathbf{z} \sim p(\mathbf{z})$, $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.
- AF prior decoder path: $\mathbf{z}^* \sim p(\mathbf{z}^*)$, $\mathbf{z} = f(\mathbf{z}^*, \boldsymbol{\lambda})$, $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

Chen X. et al. Variational Lossy Autoencoder, 2016

Flows-based VAE prior vs posterior



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

VAE limitations

- Poor generative distribution (decoder)

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

- 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)

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

Outline

1. Disentanglement learning
2. Likelihood-free learning
3. Generative adversarial networks

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.

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 $\beta = 1$?

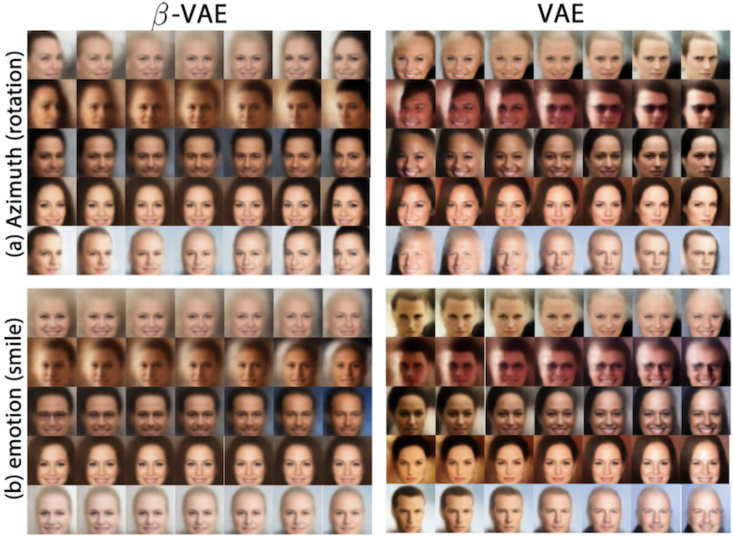
Constrained optimization

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

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

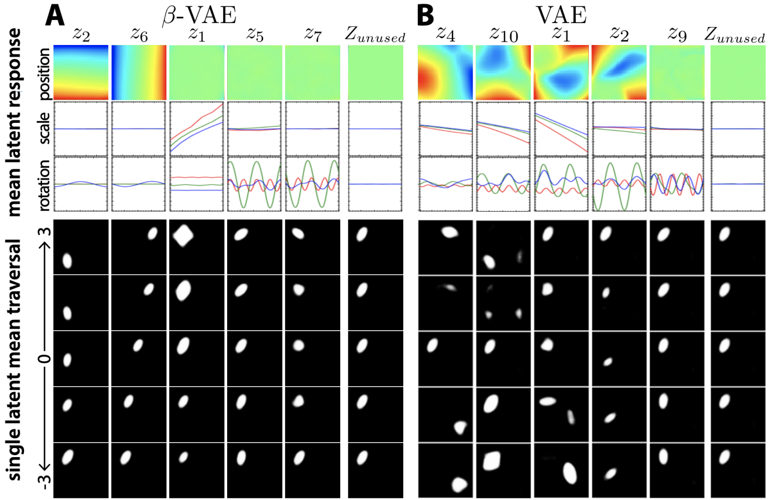
Higgins I. et al. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, 2017

β-VAE samples



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

β-VAE analysis



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

β-VAE

ELBO

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

ELBO surgery

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

Minimization of MI

- It is not necessary and not desirable for disentanglement.
- It hurts reconstruction.

Burgess C. P. et al. Understanding disentangling in β-VAE, 2018

DIP-VAE: disentangled posterior

Disentangled aggregated variational posterior

$$q_{\text{agg}}(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} q(\mathbf{z}|\mathbf{x}) = \prod_{j=1}^{d} q_{\text{agg}}(z_j)$$

DIP-VAE objective

$$\mathcal{L}_{\text{DIP}}(q, \boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_i(q, \boldsymbol{\theta}) - \lambda \cdot KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) =$$

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

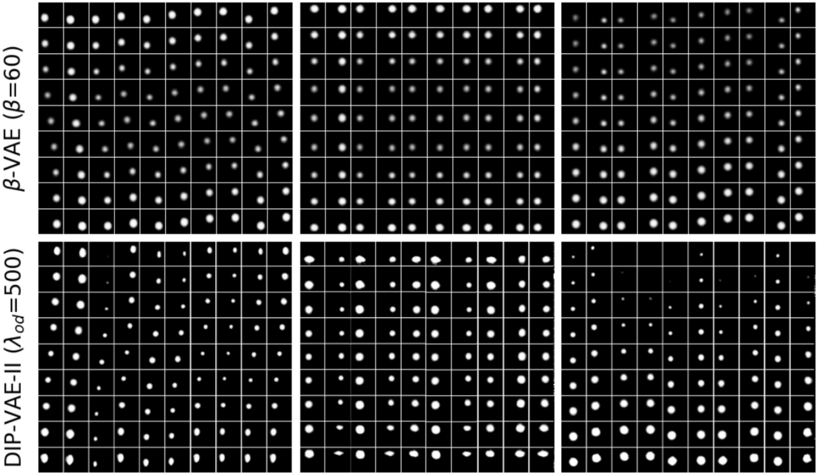
Marginal KL term is intractable. $\Rightarrow$ Let match the moments of $q_{\text{agg}}(\mathbf{z})$ and $p(\mathbf{z})$:

$$\text{cov}_{q_{\text{agg}}(\mathbf{z})}(\mathbf{z}) = \mathbb{E}_{q_{\text{agg}}(\mathbf{z})}\left[(\mathbf{z} - \mathbb{E}_{q_{\text{agg}}(\mathbf{z})}(\mathbf{z}))(\mathbf{z} - \mathbb{E}_{q_{\text{agg}}(\mathbf{z})}(\mathbf{z}))^T\right].$$

Kumar A., Sattigeri P., Balakrishnan A. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations, 2017

DIP-VAE: analysis

Reconstructions become better.



Kumar A., Sattigeri P., Balakrishnan A. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations, 2017

Challenging disentanglement assumptions

Theorem

Let $\mathbf{z}$ has density $p(\mathbf{z}) = \prod_{i=1}^{d} p(z_i)$. Then, there exists an **infinite** family of bijective functions $f : \text{supp}(\mathbf{z}) \to \text{supp}(\mathbf{z})$:

- $\frac{\partial f_i(\mathbf{z})}{\partial z_j} \neq 0$ for all i and j ($\mathbf{z}$ and $f(\mathbf{z})$ are completely entangled);
- $P(\mathbf{z} \leq \mathbf{u}) = P(f(\mathbf{z}) \leq \mathbf{u})$ for all $\mathbf{u} \in \text{supp}(\mathbf{z})$.

Consider a generative model with disentangled representation $\mathbf{z}$.

- $\exists\ \hat{\mathbf{z}} = f(\mathbf{z})$ where $\hat{\mathbf{z}}$ is completely entangled with respect to $\mathbf{z}$.
- The disentanglement method cannot distinguish between the two equivalent generative models:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} = \int p(\mathbf{x}|\hat{\mathbf{z}})p(\hat{\mathbf{z}})d\hat{\mathbf{z}}.$$

Theorem claims that unsupervised disentanglement learning is impossible for arbitrary generative models with a factorized prior.

Locatello F. et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, 2018

Outline

1. Disentanglement learning
2. Likelihood-free learning
3. Generative adversarial networks

Likelihood based models

Is likelihood a good measure of model quality?

Poor likelihood
Great samples

$$p_1(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \epsilon \mathbf{I})$$

For small ϵ this model will generate samples with great quality, but likelihood will be very poor.

Great likelihood
Poor samples

$$p_2(\mathbf{x}) = 0.01 p(\mathbf{x}) + 0.99 p_{\text{noise}}(\mathbf{x})$$

$$\log\left[0.01 p(\mathbf{x}) + 0.99 p_{\text{noise}}(\mathbf{x})\right] \geq$$
$$\geq \log\left[0.01 p(\mathbf{x})\right] = \log p(\mathbf{x}) - \log 100$$

Noisy irrelevant samples, but for high dimensions $\log p(\mathbf{x})$ becomes proportional to m.

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

Likelihood-free learning

- Likelihood is not a perfect quality measure for generative model.
- Likelihood could be intractable.

Where did we start

We would like to approximate true data distribution $\pi(\mathbf{x})$. Instead of searching true $\pi(\mathbf{x})$ over all probability distributions, learn function approximation $p(\mathbf{x}|\boldsymbol{\theta}) \approx \pi(\mathbf{x})$.

Imagine we have two sets of samples

- $\mathcal{S}_1 = \{\mathbf{x}_i\}_{i=1}^{n_1} \sim \pi(\mathbf{x})$ – real samples;
- $\mathcal{S}_2 = \{\mathbf{x}_i\}_{i=1}^{n_2} \sim p(\mathbf{x}|\boldsymbol{\theta})$ – generated (or fake) samples.

Two sample test

$$H_0 : \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}), \quad H_1 : \pi(\mathbf{x}) \neq p(\mathbf{x}|\boldsymbol{\theta})$$

Define test statistic $T(\mathcal{S}_1, \mathcal{S}_2)$. The test statistic is likelihood free. If $T(\mathcal{S}_1, \mathcal{S}_2) < \alpha$, then accept H_0, else reject it.

Likelihood-free learning

Two sample test

$$H_0 : \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}), \quad H_1 : \pi(\mathbf{x}) \neq p(\mathbf{x}|\boldsymbol{\theta})$$

Desired behaviour

- $p(\mathbf{x}|\boldsymbol{\theta})$ minimizes the value of test statistic $T(\mathcal{S}_1, \mathcal{S}_2)$.
- It is hard to find an appropriate test statistic in high dimensions. $T(\mathcal{S}_1, \mathcal{S}_2)$ could be learnable.

GAN objective

- **Generator:** generative model $\mathbf{x} = G(\mathbf{z})$, which makes generated sample more realistic.
- **Discriminator:** a classifier $D(\mathbf{x}) \in [0, 1]$, which distinguishes real samples from generated samples.

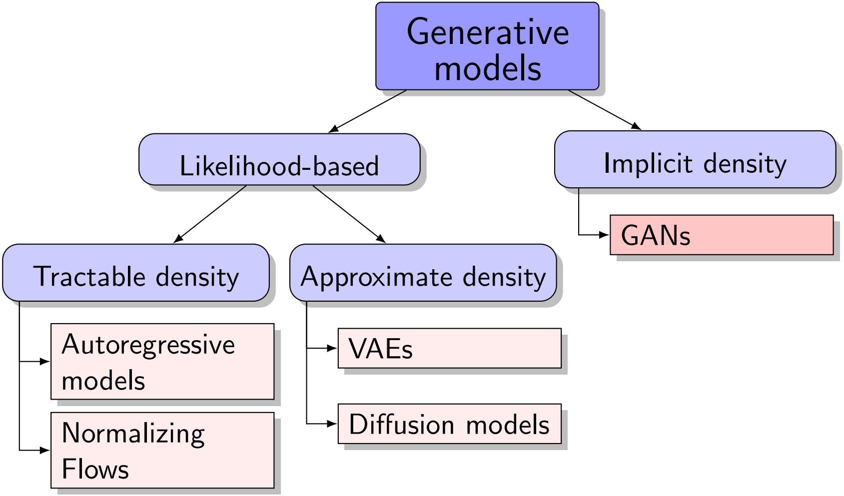
$$\min_G \max_D V(G, D) = \min_G \max_D \left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))\right]$$

Goodfellow I. J. et al. Generative Adversarial Networks, 2014

Outline

1. Disentanglement learning
2. Likelihood-free learning
3. Generative adversarial networks

Generative models zoo



Vanilla GAN optimality

Theorem

The minimax game

$$\min_G \max_D V(G, D) = \min_G \max_D \left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))\right]$$

has the global optimum $\pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta})$, in this case $D^*(\mathbf{x}) = 0.5$.

Proof (fixed G)

$$\begin{aligned} V(G, D) &= \mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \log(1 - D(\mathbf{x})) \\ &= \int \underbrace{[\pi(\mathbf{x}) \log D(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta}) \log(1 - D(\mathbf{x})]}_{y(D)} d\mathbf{x} \end{aligned}$$

$$\frac{dy(D)}{dD} = \frac{\pi(\mathbf{x})}{D(\mathbf{x})} - \frac{p(\mathbf{x}|\boldsymbol{\theta})}{1 - D(\mathbf{x})} = 0 \quad \Rightarrow \quad D^*(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}$$

Goodfellow I. J. et al. Generative Adversarial Networks, 2014

Vanilla GAN optimality

Proof continued (fixed $D = D^*$)

$$V(G, D^*) = \mathbb{E}_{\pi(\mathbf{x})} \log \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})} + \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}$$

$$= KL \left(\pi(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2} \right) + KL \left(p(\mathbf{x}|\boldsymbol{\theta}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2} \right) - 2 \log 2$$

$$= 2 JSD(\pi(\mathbf{x}) || p(\mathbf{x}|\boldsymbol{\theta})) - 2 \log 2.$$

Jensen-Shannon divergence (symmetric KL divergence)

$$JSD(\pi(\mathbf{x}) || p(\mathbf{x}|\boldsymbol{\theta})) = \frac{1}{2} \left[KL \left(\pi(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2} \right) + KL \left(p(\mathbf{x}|\boldsymbol{\theta}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2} \right) \right]$$

Could be used as a distance measure!

$$V(G^*, D^*) = -2 \log 2, \quad \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}).$$

Goodfellow I. J. et al. Generative Adversarial Networks, 2014

Vanilla GAN optimality

Theorem

The minimax game

$$\min_G \max_D V(G, D) = \min_G \max_D \left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))\right]$$

has the global optimum $\pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta})$, in this case $D^*(\mathbf{x}) = 0.5$.

Proof

for fixed G:

$$D^*(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}$$

for fixed $D = D^*$:

$$\min_G V(G, D^*) = \min_G \left[2JSD(\pi||p) - \log 4\right] = -\log 4, \quad \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}).$$

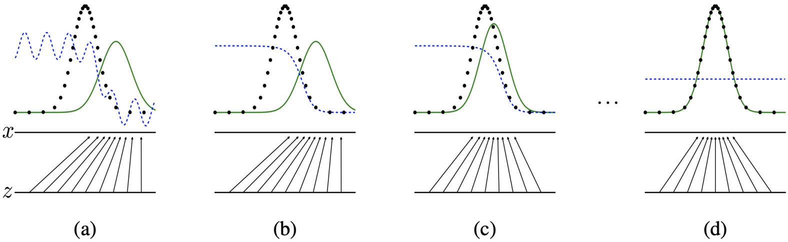
If the generator could be any function and the discriminator is optimal at every step, then the generator is guaranteed to converge to the data distribution.

Goodfellow I. J. et al. Generative Adversarial Networks, 2014

Vanilla GAN

Objective

$$\min_G \max_D V(G, D) = \min_G \max_D \left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))\right]$$



- Generator updates are made in parameter space.
- Discriminator is not optimal at every step.
- Generator and discriminator loss keeps oscillating during GAN training.

Goodfellow I. J. et al. Generative Adversarial Networks, 2014

Summary

- It is possible to use flows in VAE prior and posterior. This is almost the same.
- Disentanglement learning tries to make latent components more informative.
- β-VAE makes the latent components more independent, but the reconstructions get poorer. DIP-VAE does not make the reconstructions worse using ELBO surgery theorem.
- Majority of disentanglement learning models use heuristic objective or regularizers to achieve the goal, but the task itself could not be solved without good inductive bias.
- Likelihood is not a perfect criteria to measure quality of generative model.
- Adversarial learning suggests to solve minimax problem to match the distributions.
- Vanilla GAN tries to optimize Jensen-Shannon divergence (in theory).