Deep Generative Models

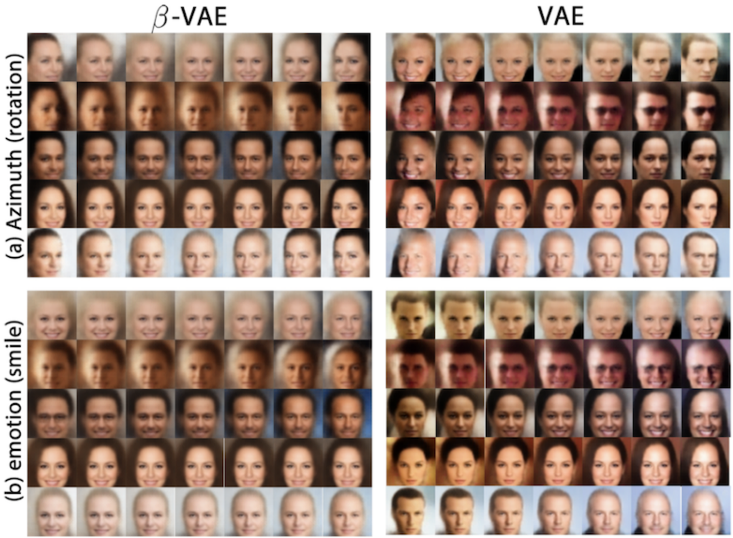
Lecture 9

Roman Isachenko

Moscow Institute of Physics and Technology

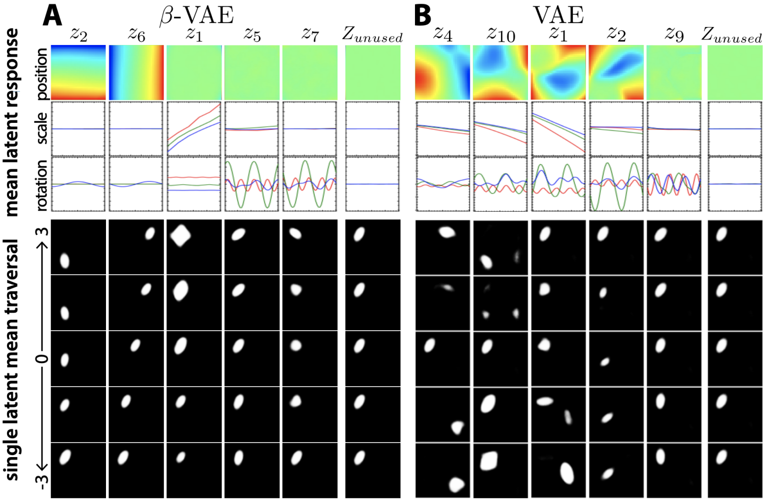
Autumn, 2021

β-VAE



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

β-VAE



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

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

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 aggregated variational posterior

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

DIP-VAE Objective

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

$$= \frac{1}{n}\sum_{i=1}^{n} \left[\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z}))\right] - \lambda \cdot KL(q(\mathbf{z})||p(\mathbf{z})) =$$

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

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

DIP-VAE

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

Let match the moments of $q(\mathbf{z})$ and $p(\mathbf{z})$:

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

DIP-VAE regularizes $\text{cov}_{q(\mathbf{z})}(\mathbf{z})$ to be close to the identity matrix.

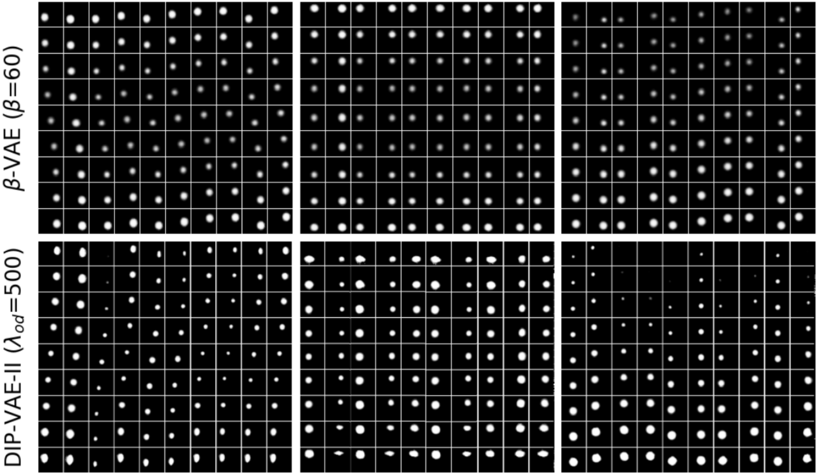
Objective

$$\max_{q, \boldsymbol{\theta}} \Big[\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i(q, \boldsymbol{\theta}) - \\ - \lambda_1 \sum_{i \neq j} \left[\text{cov}_{q(\mathbf{z})}(\mathbf{z}) \right]_{ij}^2 - \lambda_2 \sum_{i} \left(\left[\text{cov}_{q(\mathbf{z})}(\mathbf{z}) \right]_{ii} - 1 \right)^2 \Big]$$

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

DIP-VAE

Reconstructions become better.



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

Challenging Disentanglement Assumptions

Theorem

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

- $\frac{\partial f_i(\mathbf{z})}{\partial z_j} \neq 0$ 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 $\mathbf{z}$.

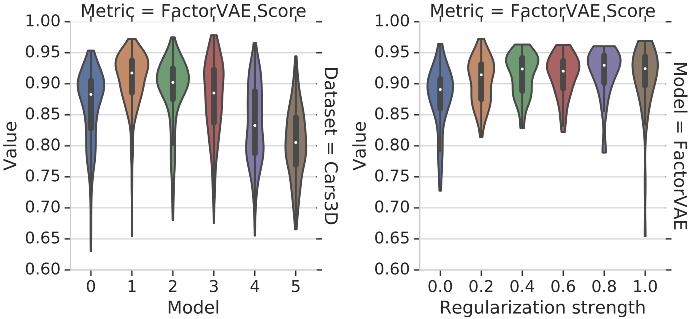
- $\exists\ \hat{\mathbf{z}} = f(\mathbf{z})$ where $\hat{\mathbf{z}}$ is completely entangled with respect to $\mathbf{z}$.
- The disentanglement method cannot distinguish between the two equivalent generative models:

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

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

Challenging Disentanglement Assumptions



Dataset = Noisy-dSprites

	(A)	(B)	(C)	(D)	(E)	(F)
BetaVAE Score (A)	100	80	44	41	46	37
FactorVAE Score (B)	80	100	49	52	25	38
MIG (C)	44	49	100	76	6	42
DCI Disentanglement (D)	41	52	76	100	-8	38
Modularity (E)	46	25	6	-8	100	13
SAP (F)	37	38	42	38	13	100

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

Recap of previous lecture

Autoregressive flow prior

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

Theorem

VAE with the AF prior for latent code $\mathbf{z}$ is equivalent to using the IAF posterior for latent code $\boldsymbol{\epsilon}$.

$$\mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \underbrace{\left(\log p(f(\mathbf{z}, \boldsymbol{\lambda})) + \log \left| \det \frac{\partial f(\mathbf{z}, \boldsymbol{\lambda})}{\partial \mathbf{z}} \right| \right)}_{\text{AF prior}} - \log q(\mathbf{z}|\mathbf{x}) \right]$$

$$= \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \log p(f(\mathbf{z}, \boldsymbol{\lambda})) - \underbrace{\left(\log q(\mathbf{z}|\mathbf{x}) - \log \left| \det \frac{\partial f(\mathbf{z}, \boldsymbol{\lambda})}{\partial \mathbf{z}} \right| \right)}_{\text{IAF posterior}} \right]$$

Chen X. et al. Variational Lossy Autoencoder, 2016

Recap of previous lecture

ELBO

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

- Normal variational distribution $q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}^2_\phi(\mathbf{x}))$ is poor (e.g. has only one mode).
- Flows models convert a simple base distribution to a compex one using an invertible transformation with simple Jacobian.

Flow model in latent space

$$\log q_K(\mathbf{z}_K|\mathbf{x}, \phi_*) = \log q(\mathbf{z}_0|\mathbf{x}, \phi) - \sum_{k=1}^{K} \log \left| \det \left(\frac{\partial g_k(\mathbf{z}_{k-1}, \phi_k)}{\partial \mathbf{z}_{k-1}} \right) \right|.$$

Let's use $q_K(\mathbf{z}_K|\mathbf{x}, \phi_*)$, $\phi_* = \{\phi, \phi_1, \dots, \phi_K\}$ as a variational distribution. Here, ϕ – encoder parameters, $\{\phi_k\}_{k=1}^K$ – flow parameters.

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

Recap of previous lecture

Variational distribution

$$\log q_K(\mathbf{z}_K|\mathbf{x}, \phi_*) = \log q(\mathbf{z}_0|\mathbf{x}, \phi) - \sum_{k=1}^{K} \log \left| \det \left(\frac{\partial g_k(\mathbf{z}_{k-1}, \phi_k)}{\partial \mathbf{z}_{k-1}} \right) \right| .$$

ELBO objective

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{z}_0|\mathbf{x},\phi)} \left[\log p(\mathbf{x}, \mathbf{z}_K|\boldsymbol{\theta}) - \log q(\mathbf{z}_0|\mathbf{x}, \phi) + \right.$$
$$\left. + \sum_{k=1}^{K} \log \left| \det \left(\frac{\partial g_k(\mathbf{z}_{k-1}, \phi_k)}{\partial \mathbf{z}_{k-1}} \right) \right| \right] .$$

- Obtain samples $\mathbf{z}_0$ from the encoder.
- Apply flow model $\mathbf{z}_K = g(\mathbf{z}_0, \{\phi_k\}_{k=1}^K)$.
- Compute likelihood for $\mathbf{z}_K$ using the decoder, base distribution for $\mathbf{z}_0$ and the Jacobian.
- We do not need an inverse flow function if we use flows in variational inference.

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

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

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

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

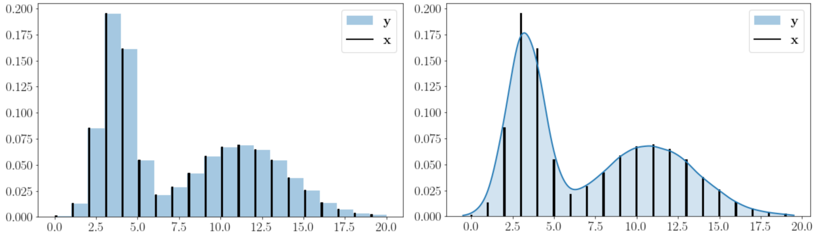
Variational dequantization bound

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

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

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

Disentanglement learning

A disentangled representation is a one where single latent units are sensitive to changes in single generative factors, while being invariant to changes in other factors.

β-VAE

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

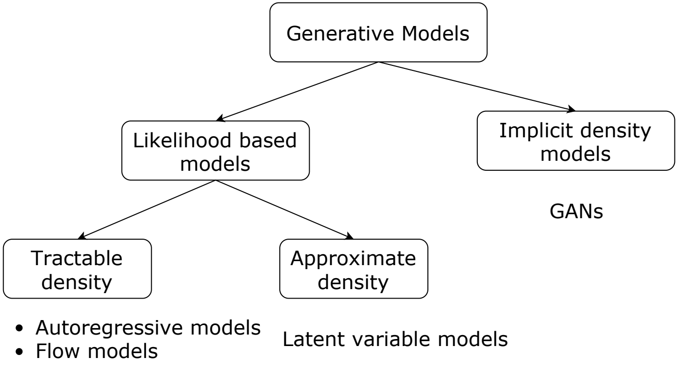
Representations becomes disentangled by setting a stronger constraint with $\beta > 1$. However, it leads to poorer reconstructions and a loss of high frequency details.

ELBO surgery

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

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

Generative models zoo



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 measure 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

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

- **Generator:** 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.

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 (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 confitnued (fixed $D = D^*$)

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

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

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

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 $D = D^*$:

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

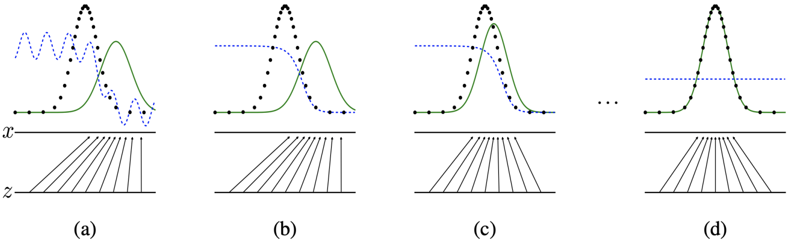
If the generator could be any function and the discriminator is optimal at every step, then the generator is guaranteed to converge to the data distribution.

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

- 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 suggest to solve minimax problem to match the distributions.
- Vanilla GAN tries to optimize Jensen-Shannon divergence (in theory).