Deep Generative Models
Lecture 9

Roman Isachenko

Ozon Masters

Spring, 2021

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 $p(\mathbf{z})$:

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

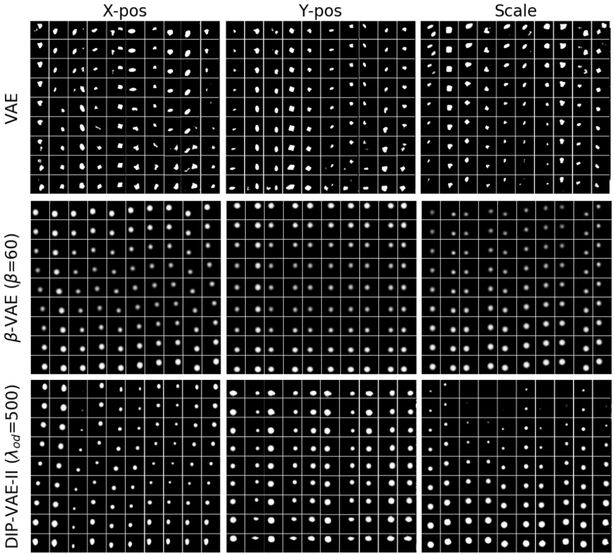
DIP-VAE regularizes $\text{cov}_{q(\mathbf{z})}(\mathbf{z})$ to be close to the identity matrix.

Objective

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

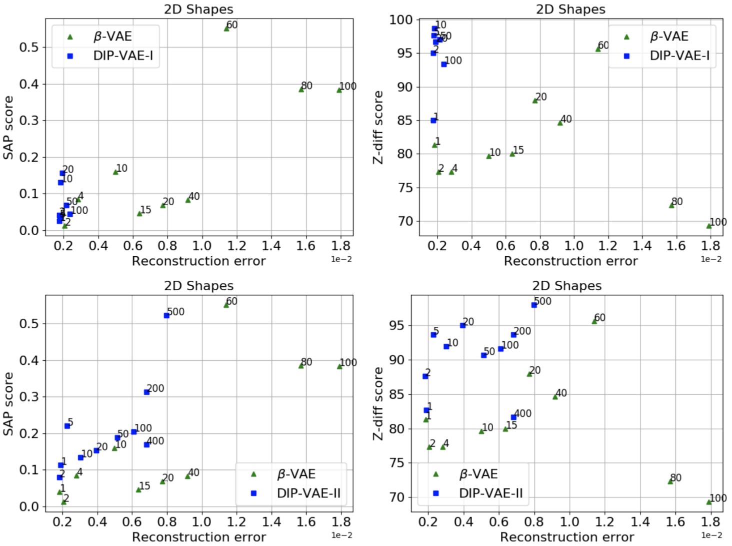
Kumar A., Sattigeri P., Balakrishnan A. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations, 2017

DIP-VAE



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

DIP-VAE



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

Challenging Disentanglement Assumptions

Whether unsupervised disentanglement learning is even possible for arbitrary generative models?

Theorem

For $d > 1$, let $\mathbf{z} \sim P$ denote any distribution which admits a density $p(\mathbf{z}) = \prod_{i=1}^{d} p(z_i)$. Then, there exists an infinite family of bijective functions $f : \text{supp}(\mathbf{z}) \to \text{supp}(\mathbf{z})$ such that

- $\frac{\partial f_i(\mathbf{u})}{\partial u_j} \neq 0$ almost everywhere for all i and j (i.e., $\mathbf{z}$ and $f(\mathbf{z})$ are completely entangled);
- and $P(\mathbf{z} \leq \mathbf{u}) = P(f(\mathbf{z}) \leq \mathbf{u})$ for all $\mathbf{u} \in \text{supp}(\mathbf{z})$ (i.e., they have the same marginal distribution).

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

Assume we have $p(\mathbf{z})$ and some $p(\mathbf{x}|\mathbf{z})$ defining a generative model. Consider any unsupervised disentanglement method and assume that it finds a representation that is perfectly disentangled with respect to $\mathbf{z}$ in the generative model.

- Theorem claims that $\exists\ \hat{\mathbf{z}} = f(\mathbf{z})$ where $\hat{\mathbf{z}}$ is completely entangled with respect to $\mathbf{z}$.
- Since the (unsupervised) disentanglement method only has access to observations $\mathbf{x}$, it hence cannot distinguish between the two equivalent generative models and thus has to be entangled to at least one of them

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

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

Challenging Disentanglement Assumptions

Proof (1)

1. Consider the function $g : \text{supp}(\mathbf{z}) \to [0, 1]^d$:

$$g_i(\mathbf{v}) = P(z_i \le v_i), \quad i = 1, \ldots, d.$$

   - g is bijective (since $p(\mathbf{z}) = \prod_{i=1} dp(z_i)$).
   - $\frac{\partial g_i(\mathbf{u})}{\partial u_i} \neq 0$, for all i and $\frac{\partial g_i(\mathbf{u})}{\partial u_j} = 0$ for all $i \neq j$.
   - $g(\mathbf{z})$ is an independent d-dimensional uniform distribution.

2. Consider $h : (0, 1]^d \to \mathbb{R}^d$

$$h_i(\mathbf{v}) = \psi^{-1}(v_i), \quad i = 1, \ldots, d.$$

   Here ψ denotes the CDF of a standard normal distribution.

   - h is bijective.
   - $\frac{\partial g_i(\mathbf{u})}{\partial u_i} \neq 0$, for all i and $\frac{\partial g_i(\mathbf{u})}{\partial u_j} = 0$ for all $i \neq j$.
   - $h(g(\mathbf{z}))$ is a d-dimensional standard normal distribution.

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

Challenging Disentanglement Assumptions

Proof (2)

Let $\mathbf{A} \in \mathbb{R}^{d \times d}$ be an arbitrary orthogonal matrix with $A_{ij} \neq 0$ for all i, j. The family of such matrices is infinite.

- $\mathbf{A}$ is orthogonal, it is invertible and thus defines a bijective linear operator.
- $\mathbf{A}h(g(\mathbf{z})) \in \mathbb{R}^d$ is hence an independent, multivariate standard normal distribution.
- $h^{-1}(\mathbf{A}h(g(\mathbf{z}))) \in \mathbb{R}^d$ is an independent d-dimensional uniform distribution.

Define $f : \mathrm{supp}(\mathbf{z}) \to \mathrm{supp}(\mathbf{z})$:

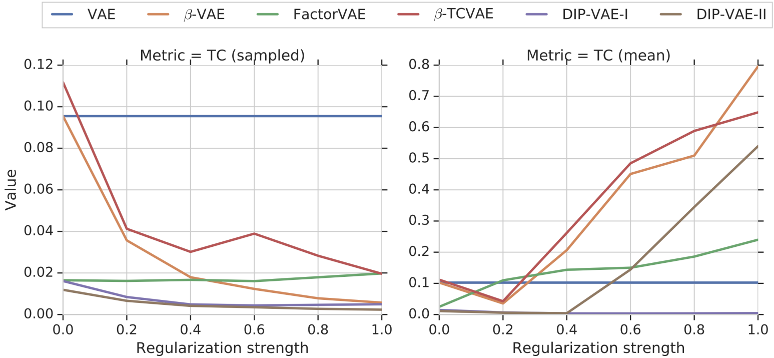
$$f(\mathbf{u}) = g^{-1}(h^{-1}(\mathbf{A}h(g(\mathbf{z})))).$$

By definition $f(\mathbf{z})$ has the same marginal distribution as $\mathbf{z}$: $\underline{P(\mathbf{z} \leq \mathbf{u}) = P(f(\mathbf{z}) \leq \mathbf{u}})$ and $\frac{\partial f_i(\mathbf{u})}{\partial u_j} \neq 0$.

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

Challenging Disentanglement Assumptions

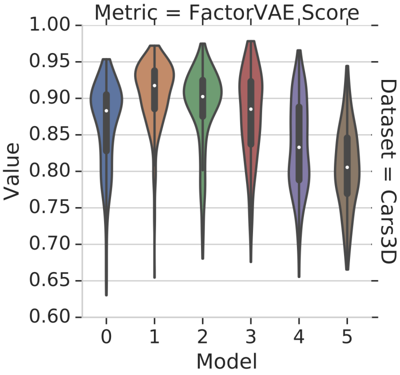
- **Training:** Factorizing **samples** from aggregated posterior $q(\mathbf{z}) = \prod_{i=1}^{d} q(z_i)$.
- **Inference:** Use a **mean** vector (usually mean of Gaussian encoder) as a representation.

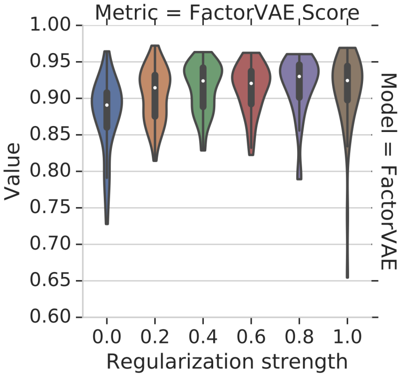


Locatello F. et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, 2018

Challenging Disentanglement Assumptions

Importance of different models and hyperparameters for disentanglement

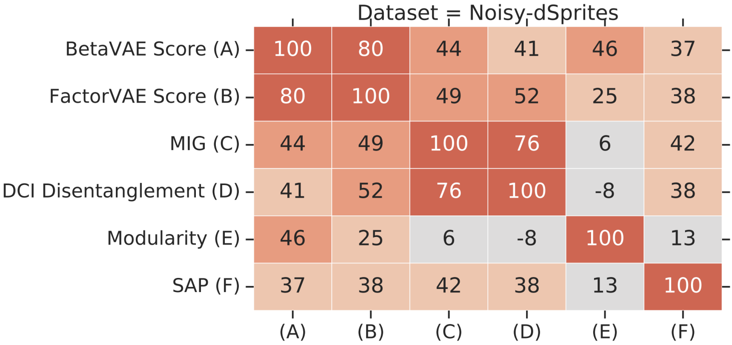




Locatello F. et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, 2018

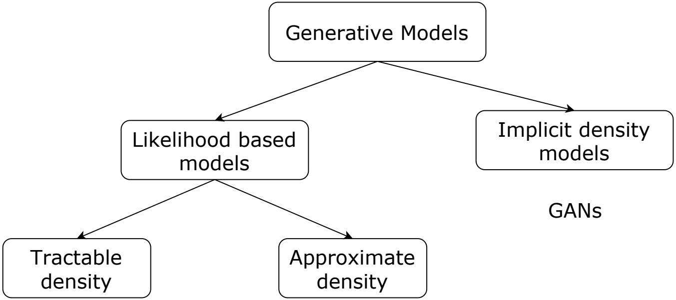
Challenging Disentanglement Assumptions

Agreement of different disentanglement metrics



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

Generative models zoo



- Autoregressive models
- Flow models

Latent variable models

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

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

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

Noisy irrelevant samples, but for high dimensions $\log p(\mathbf{x})$ becames larger.

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

Likelihood-free learning

- Likelihood is not a perfect measure for 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

- $\mathcal{S}_1 = \{\mathbf{x}_i\}_{i=1}^{n_1} \sim \pi(\mathbf{x})$
- $\mathcal{S}_2 = \{\mathbf{x}_i\}_{i=1}^{n_2} \sim p(\mathbf{x}|\boldsymbol{\theta})$

Two sample test

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

Likelihood-free learning

Two sample test

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

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

Desired behaviour

- The generative model $p(\mathbf{x}|\boldsymbol{\theta})$ 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. We could try to learn the appropriate $T(\mathcal{S}_1, \mathcal{S}_2)$.

Vanila GAN

- **Generator:** latent variable model $\mathbf{x} = G(\mathbf{z})$, which minimizes two-sample test objective.
- **Discriminator:** function $D(\mathbf{x})$, which distinguishes real samples from model samples and maximizes a two-sample test statistic.

Objective

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

For fixed generator $G(\mathbf{z})$ the discriminator is performing a binary classification with a cross entropy loss.
This minimax game has the global optimum $\pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta})$.

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

Vanila GAN optimality

Objective (fixed G)

$$\max_D V(D) = \mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))$$

Optimal discriminator

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

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

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

Vanila GAN optimality

Objective (fixed D)

$$\max_G V(G, D^*) = \mathbb{E}_{\pi(\mathbf{x})} \log D^*(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D^*(G(\mathbf{z})))$$

Optimal generator

$$V(G, D^*) = \mathbb{E}_{\pi(\mathbf{x})} \log \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})} + \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}$$
$$= KL\left(\pi(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2}\right) + KL\left(p(\mathbf{x}|\boldsymbol{\theta}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2}\right) - 2\log 2$$
$$= 2JSD(\pi(\mathbf{x}) || p(\mathbf{x}|\boldsymbol{\theta})) - 2\log 2.$$

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

Vanila GAN optimality

Optimal generator

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

Jensen-Shannon divergence

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

Also called symmetric KL divergence. Could be used as a distance measure!

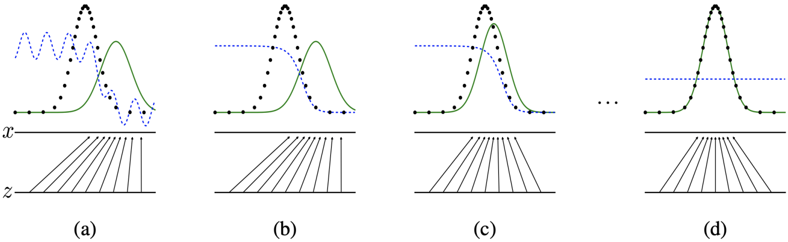
$$V(G^*, D^*) = -2\log 2, \quad \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}).$$

If the generator updates are made in a function space and the discriminator is optimal at every step, then the generator is guaranteed to converge to the data distribution.

Vanila GAN

Objective

$$V(G, D) = \min_{G} \max_{D} \mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))$$



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

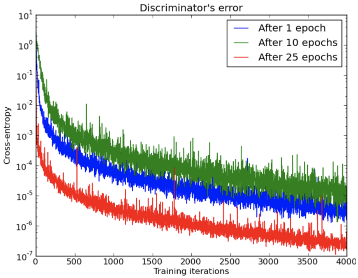
Goodfellow I. J. et al. Generative Adversarial Networks, 2014

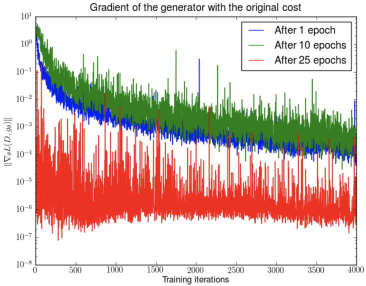
Vanishing gradients

Objective

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

Early in learning, G is poor, D can reject samples with high confidence. In this case, $\log(1 - D(G(\mathbf{z})))$ saturates.





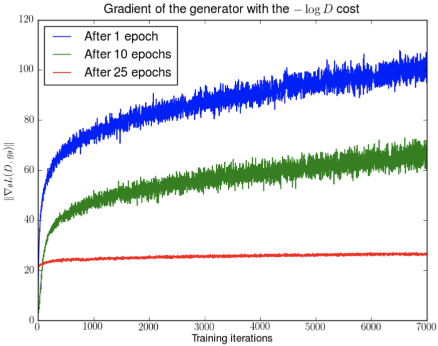
Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017

Vanishing gradients

Objective

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

Maximize $\log D(G(z))$ instead of $\log(1 - D(G(\mathbf{z})))$.
Gradients are getting much stronger, but the training is instable (with increasing mean and variance).



Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017

Likelihood-based vs GAN

GAN objective (for optimal discriminator)

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

Likelihood model objective

$$\max_{\boldsymbol{\theta}} \log p(\mathbf{X}|\boldsymbol{\theta}) \approx \max_{\boldsymbol{\theta}} \mathbb{E}_{\pi(\mathbf{x})} \log p(\mathbf{x}|\boldsymbol{\theta}) =$$

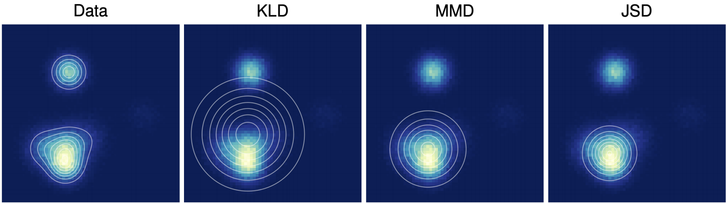
$$= \max_{\boldsymbol{\theta}} \mathbb{E}_{\pi(\mathbf{x})} \log p(\mathbf{x}|\boldsymbol{\theta}) - \mathbb{E}_{\pi(\mathbf{x})} \log \pi(\mathbf{x}) =$$

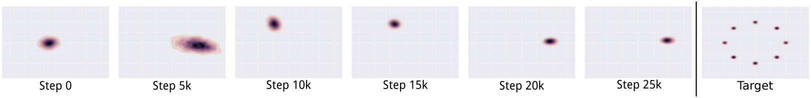
$$= \max_{\boldsymbol{\theta}} \mathbb{E}_{\pi(\mathbf{x})} \log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{\pi(\mathbf{x})} = \min_{\boldsymbol{\theta}} KL(\pi(\mathbf{x})||p(\mathbf{x}|\boldsymbol{\theta}))$$

What is the difference between *JS* and *KL* divergences?

Mode collapse

The phenomena where the generator of a GAN collapses to one or few distribution modes.





Alternate architectures, adding regularization terms, injecting small noise perturbations and other millions bags and tricks are used to avoid the mode collapse.

Goodfellow I. J. et al. Generative Adversarial Networks, 2014
Metz L. et al. Unrolled Generative Adversarial Networks, 2016

Mode collapse

Mode covering vs mode seeking

$$KL(\pi||p) = \int \pi(\mathbf{x}) \log \frac{\pi(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x}, \quad KL(p||\pi) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\pi(\mathbf{x})} d\mathbf{x}$$

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

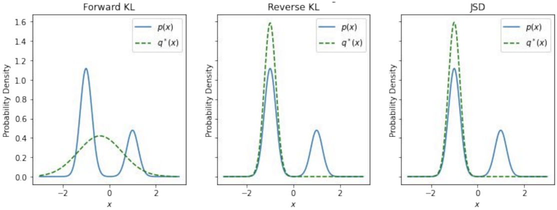
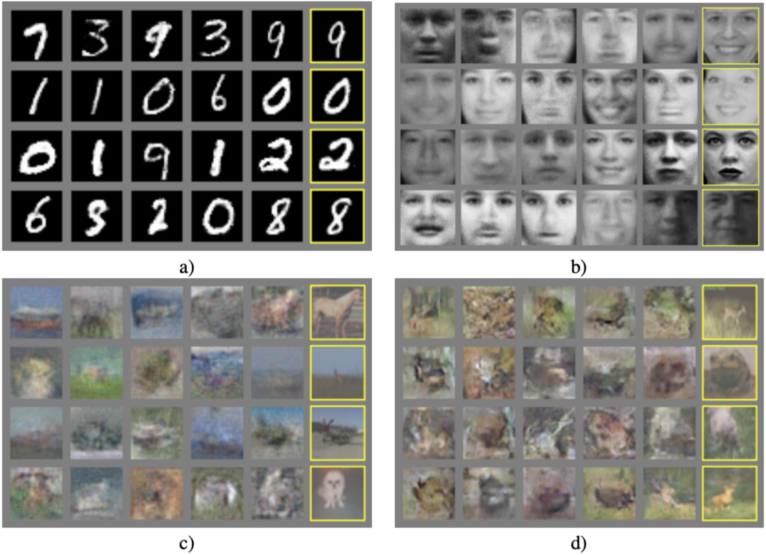


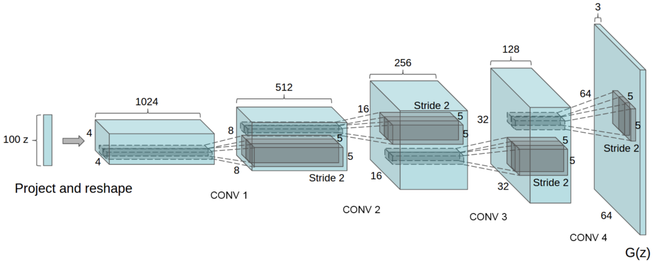
image credit: *https://sites.google.com/view/berkeley-cs294-158-sp20/home*

Vanila GAN results



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

Deep Convolutional GAN



- Mean-pooling instead of max-pooling.
- Transposed convolutions in the generator for upsampling.
- Downsample with strided convolutions and average pooling.
- ReLU for generator, Leaky-ReLU (0.2) for discriminator.
- Output nonlinearity: tanh for Generator, sigmoid for discriminator.
- Batch Normalization used to prevent mode collapse (not applied at the output of G and input of D).
- Adam: small LR = 2e-4; small momentum: 0.5, batch-size: 128.

Radford A., Metz L., Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015

Deep Convolutional GAN

ImageNet samples

Radford A., Metz L., Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015

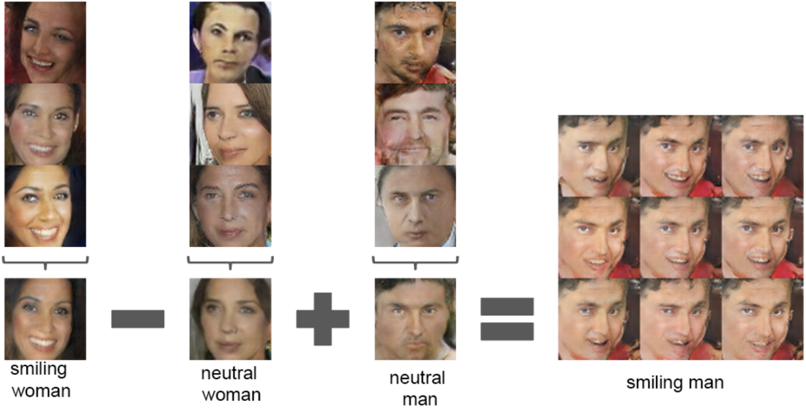
Deep Convolutional GAN

Smooth interpolations

Radford A., Metz L., Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015

Deep Convolutional GAN

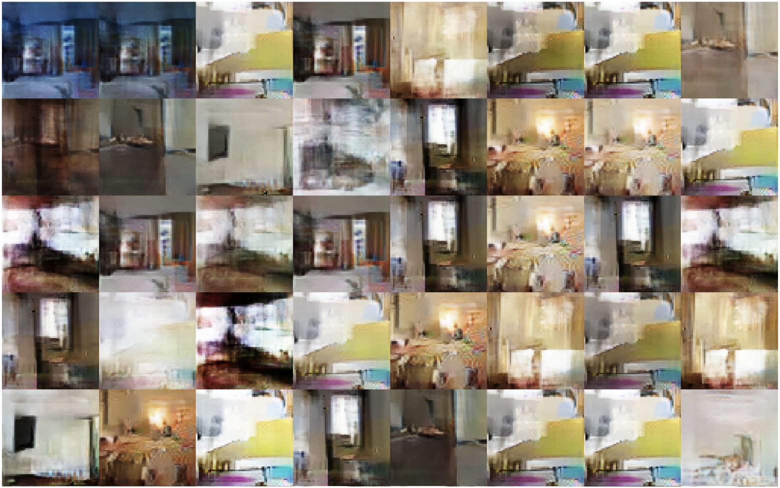
Vector arithmetic



Radford A., Metz L., Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015

Deep Convolutional GAN

Mode collapse



Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Improved techniques for training GANs

- Feature matching

$$\mathcal{L}_G = \|\mathbb{E}_{\pi(\mathbf{x})}\mathbf{d}(\mathbf{x}) - \mathbb{E}_{p(\mathbf{z})}\mathbf{d}(G(\mathbf{z}))\|_2^2$$

  Here $\mathbf{d}(\mathbf{x})$ – intermediate layer of discriminator. Matching the learned discriminator statistics instead of the output of the discriminator. Helps to avoid the vanishing gradients for sufficiently good discriminator.

- Historical averaging adds extra loss term for generator and discriminator losses

$$\|\boldsymbol{\theta} - \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\theta}_t\|_2^2.$$

  Here $\boldsymbol{\theta}_t$ – value of parameters at the previous step t. It allows to stabilize training procedure.

Salimans T. et al. Improved Techniques for Training GANs, 2016

Improved techniques for training GANs

- One-sided label smoothing. Instead of using one-hot labels in classification, use $(1-\alpha)$ for real data (the generated samples are not smoothed).

$$D^*(\mathbf{x}) = \frac{(1-\alpha)\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}$$

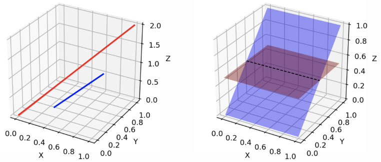
- Virtual batch normalization. BatchNorm makes samples within minibatch are highly correlated.

Use reference fixed batch to compute the normalization statistics. To avoid overfitting construct batch with the reference batch and the current sample.

Salimans T. et al. Improved Techniques for Training GANs, 2016

Informal theoretical results

- Since $\mathbf{z}$ usually has lower dimensionality compared to $\mathbf{x}$, manifold $G(\mathbf{z})$ has a measure 0 in $\mathbf{x}$ space. Hence, support of $p(\mathbf{x}|\boldsymbol{\theta})$ lies on low-dimensional manifold.
- Distribution of real images $\pi(\mathbf{x})$ is also concentrated on a low dimensional manifold.



- If $\pi(\mathbf{x})$ and $p(\mathbf{x}|\boldsymbol{\theta})$ have disjoint supports, then there is a smooth optimal discrimator. We are not able to learn anything by backproping through it.
- For such low-dimensional disjoint manifolds

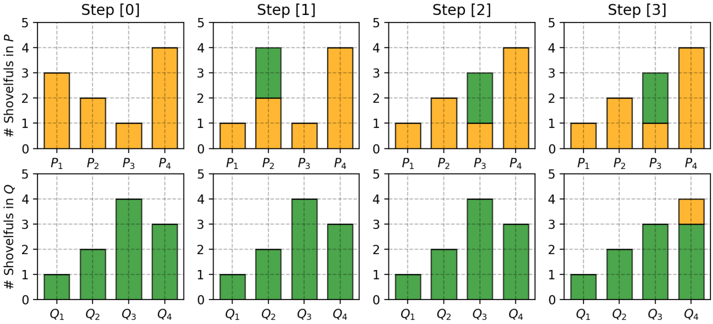
$$KL(\pi||p) = KL(p||\pi) = \infty, \quad JSD(\pi||p) = \log 2$$

- Adding continuous noise to the inputs of the discriminator smoothes the distributions of the probability mass.

Weng L. From GAN to WGAN, 2019
Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017

Wasserstein distance (discrete)

Also called Earth Mover's distance. The minimum cost of moving and transforming a pile of dirt in the shape of one probability distribution to the shape of the other distribution.



$$W(P, Q) = 2(\text{step } 1) + 2(\text{step } 2) + 1(\text{step } 3) = 5$$

Weng L. From GAN to WGAN, 2019

Wasserstein distance

$$W(\pi, p) = \inf_{\gamma \in \prod(\pi, p)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \|\mathbf{x} - \mathbf{y}\| = \inf_{\gamma \in \prod(\pi, p)} \int \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

- $\prod(\pi, p)$ – the set of all joint distributions $\gamma(\mathbf{x}, \mathbf{y})$ with marginals π and p ($\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} = p(\mathbf{y})$, $\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y} = \pi(\mathbf{x})$)
- $\gamma(\mathbf{x}, \mathbf{y})$ – transportation plan (the amount of "dirt" that should be transported from point $\mathbf{x}$ to point $\mathbf{y}$).
- $\gamma(\mathbf{x}, \mathbf{y})$ – the amount, $\|\mathbf{x} - \mathbf{y}\|$– the distance.

For better understanding of transportation plan function γ, try to write down the plan for previous discrete case.

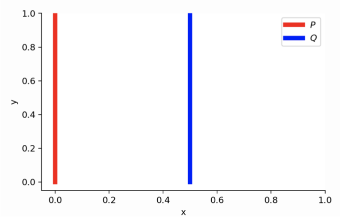
Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein distance vs KL vs JSD

Consider 2d distributions

$$\pi(x, y) = (0, U[0,1])$$

$$p(x, y|\theta) = (\theta, U[0,1])$$



- $\theta = 0$. Distributions are the same

$$KL(\pi||p) = KL(p||\pi) = JSD(p||\pi) = W(\pi, p) = 0$$

- $\theta \neq 0$

$$KL(\pi||p) = \int_{U[0,1]} 1 \log \frac{1}{0} dy = \infty = KL(p||\pi)$$

$$JSD(\pi||p) = \frac{1}{2}\left(\int_{U[0,1]} 1 \log \frac{1}{1/2} dy + \int_{U[0,1]} 1 \log \frac{1}{1/2} dy\right) = \log 2$$

$$W(\pi, p) = |\theta|$$

Weng L. From GAN to WGAN, 2019
Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Wasserstein distance vs KL vs JSD

Theorem 1

Let $G(\mathbf{z}, \boldsymbol{\theta})$ be any feedforward neural network, and $p(\mathbf{z})$ a prior over $\mathbf{z}$ such that $\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \|\mathbf{z}\| < \infty$. Then therefore $W(\pi, p)$ is continuous everywhere and differentiable almost everywhere.

Theorem 2

Let π be a distribution on a compact space $\mathcal{X}$ and $\{p_t\}_{t=1}^{\infty}$ be a sequence of distributions on $\mathcal{X}$.

$$KL(\pi || p_t) \to 0 \, (\text{or } KL(p_t || \pi) \to 0) \tag{1}$$

$$JSD(\pi || p_t) \to 0 \tag{2}$$

$$W(\pi || p_t) \to 0 \tag{3}$$

Then, considering limits as $t \to \infty$, (1) implies (2), (2) implies (3).

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017

Summary

- Majority of disentanglement learning models use heuristic objective or regularizers to achieve the goal.