16. Generative Model - VAE

Dongwoo Kim

dongwoo.kim@postech.ac.kr

CSED515 - 2022 Spring

Announcement

- Assignment 4 has been released (final assignment, total score = 150)
  - Due: June 2 (Friday), 23:59
- Final exam: June 8 (Thursday), 11:00∼12:20
  - Off-line, closed book
  - Topics: 10. Evaluation ∼ 19. Reinforcement Learning

Examples of Unsupervised Learning

In fact, we've focused on supervised learning tasks, in particular, regression and classification. In next several lectures, we will study about unsupervised learning:

- Clustering, e.g., image segmentation
- Feature selection or dimensionality reduction, e.g., PCA
- **Generative model, e.g., VAE, GAN**

Generative Models

Training data ~ $p_{data}(x)$

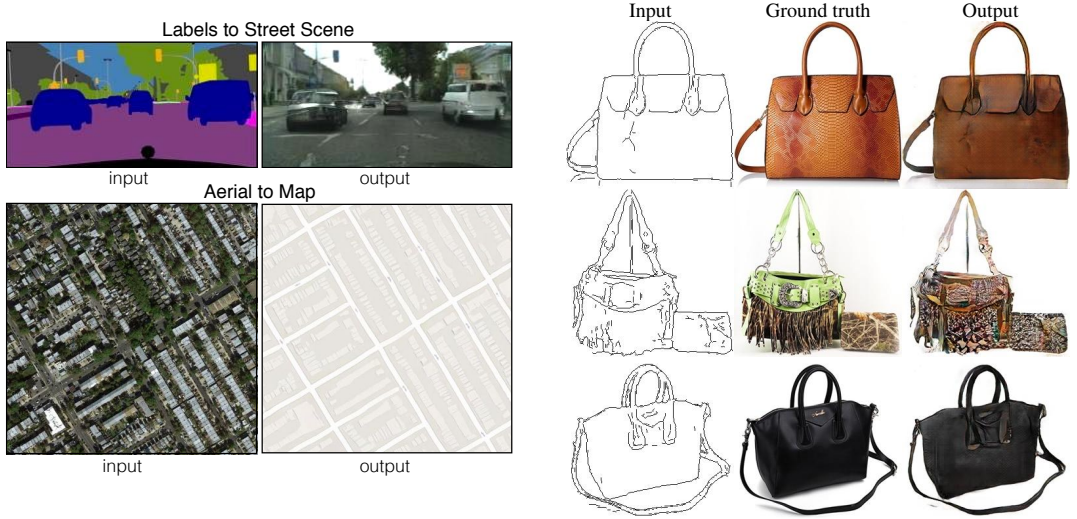
Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

- Given training dataset, we want to generate new samples from the same distribution
- In other words, we want to estimate density of data

Applications of Generative Model

- Generative model often provides useful approaches for other machine learning tasks, e.g., density estimation for regression, classification, out-of-distribution detection, ...
- Beside this, there are a number of interesting applications
  - e.g., Image-to-Image translation: super-resolution, style change, colorization, ...



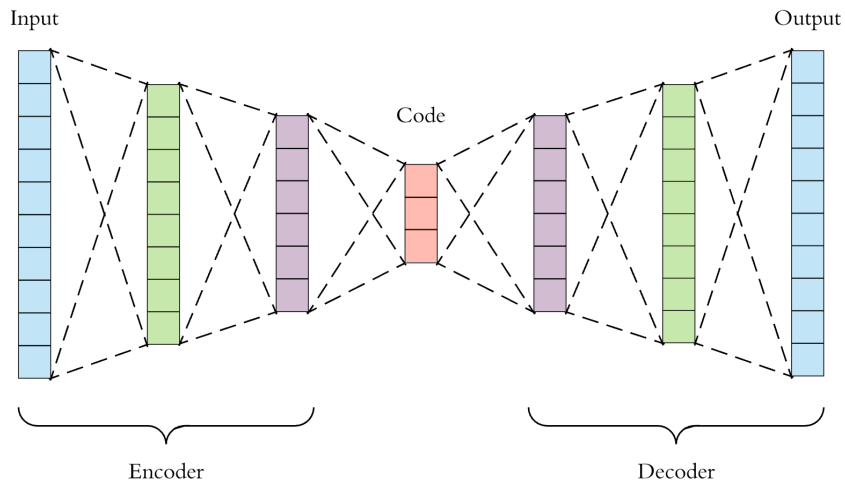
[Isola et al 16]

Applications of Generative Model

- Beside this, there are a number of interesting applications
  - e.g., POSCO steel plate with random pattern
  - More examples can be found at https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

Ordinary Auto-encoder (AE) for Data Generation?

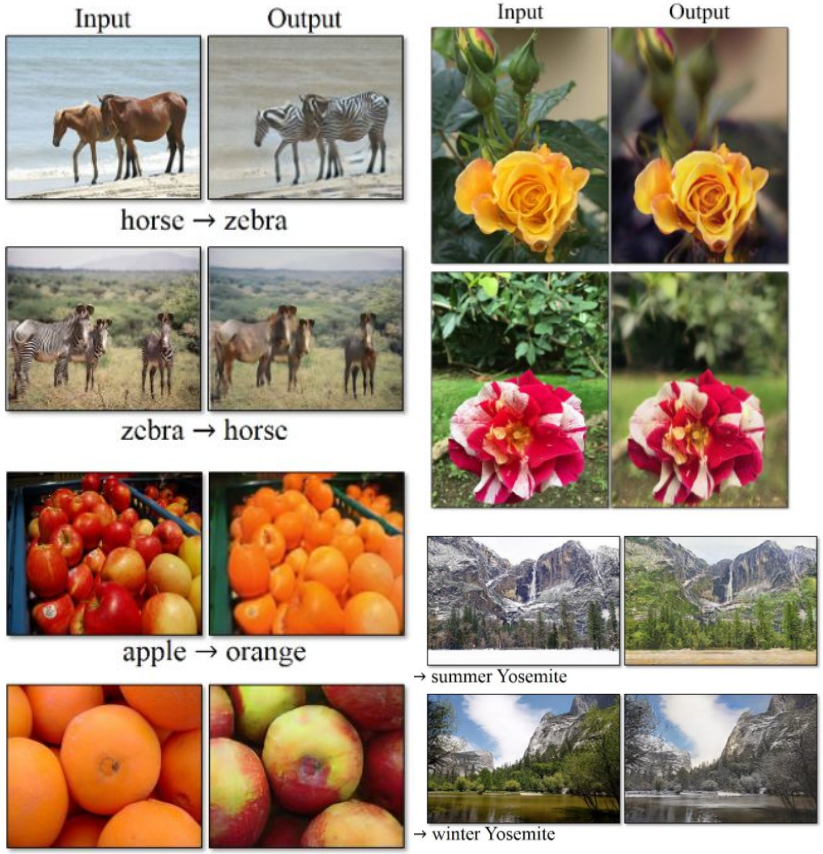
- We can use the ordinary autoencoder (in the previous lecture) to generate data
- However, we cannot control the shape of distribution of latent variable in AE
- Hence, it is hard to interpret or manipulate latent feature to generate desired data, i.e., we need some probabilistic approach



What we can do from manipulating latent feature?

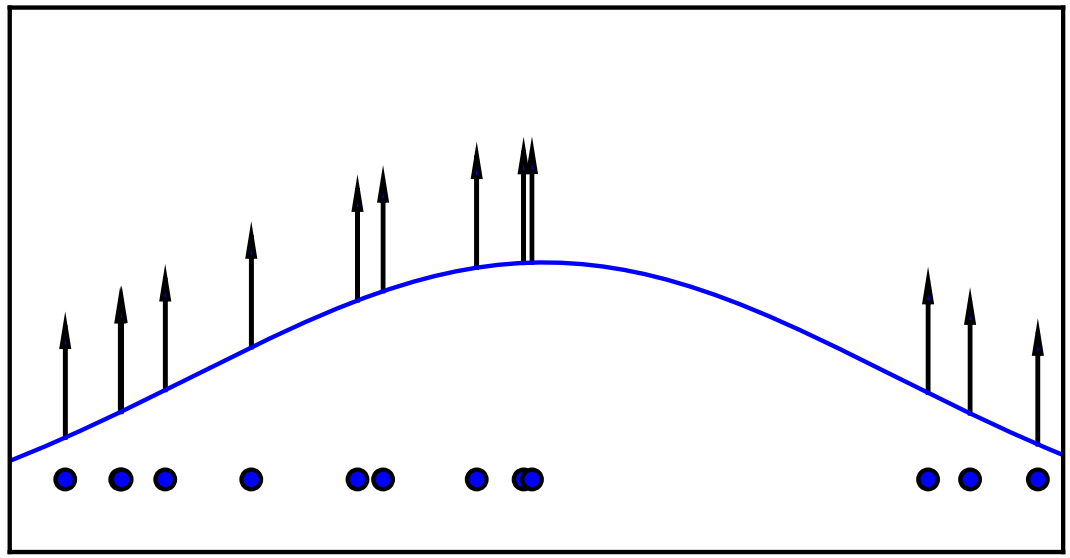
What we can do from manipulating latent feature?

Source->Target domain transfer



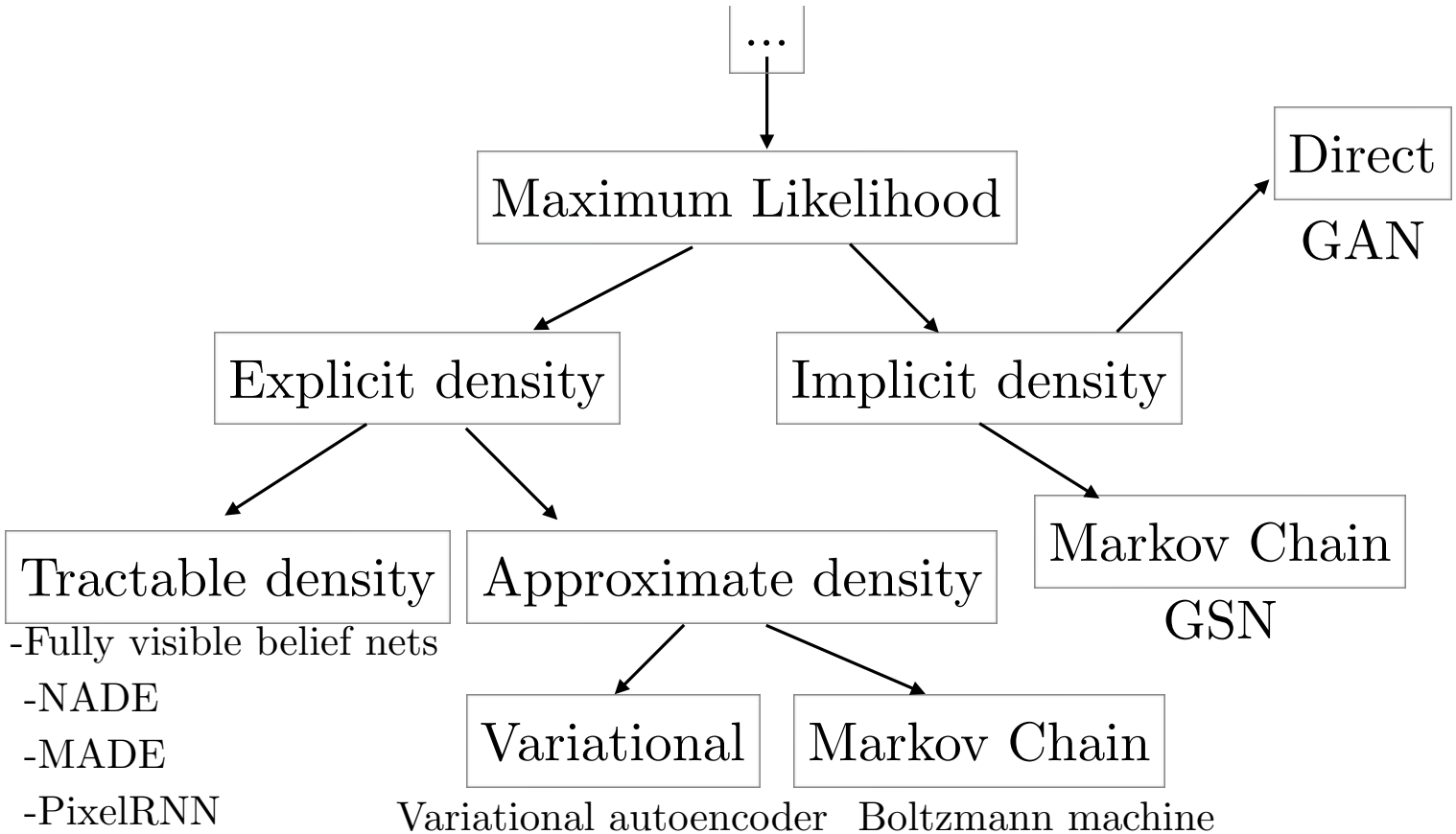
CycleGAN. Zhu et al. 2017.

A General Approach: Maximum Likelihood



$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{\theta})$$

Taxonomy of Generative Models



(Goodfellow 2016)

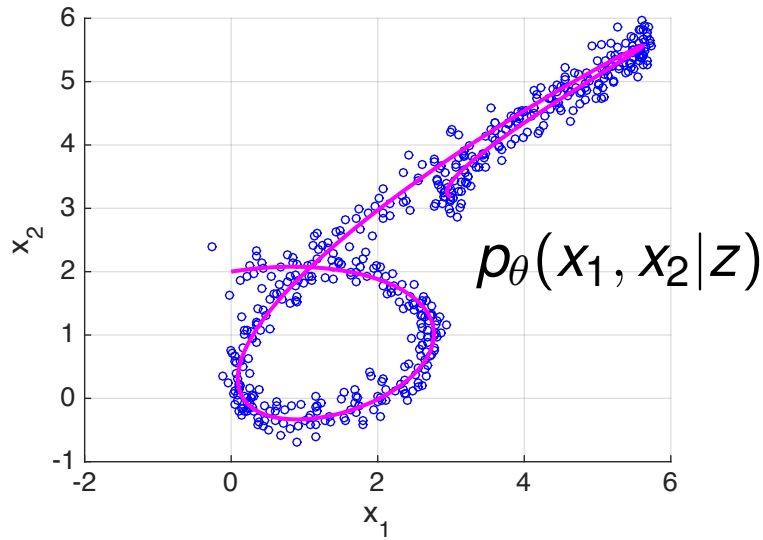
Outline

- Variational Auto-Encoder (VAE) [Kingma & Welling 14]
- Generative Adversarial Network (GAN) [Goodfellow et al 14]

Manifold Hypothesis

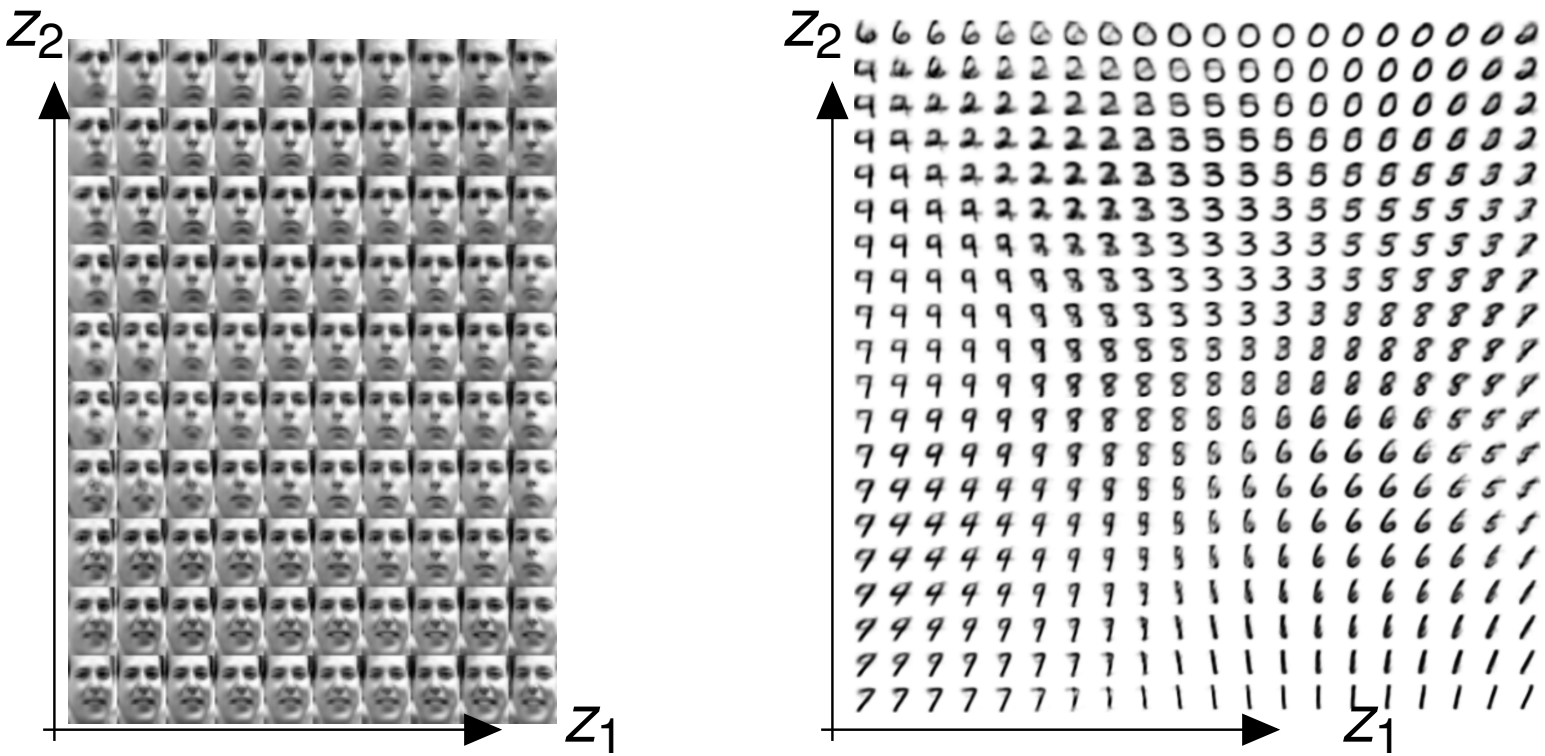
- x is a high dimensional vector
- Data is concentrated around a low dimensional manifold

$$z \in [0, 1] \quad \rightarrow$$



Manifold Hypothesis

- $x \in \mathbb{R}^D$ is a high dimensional vector
- Data is concentrated around a low dimensional manifold ($z \in \mathbb{R}^M$ with $M \ll D$)



[Kingma and Welling 14]

A Probabilistic Approach for Generative Model

Recalling MLE, our objective is maximizing

$$p_\theta(x) = \int p(z) p_\theta(x \mid z) dz$$

where generative model is $p_\theta(x \mid z)$

- Recalling manifold hypothesis, choose prior $p(z)$ to be simple, e.g., Gaussian distribution of reasonable latent attributes z, e.g., pose, degree of smile, ...
- As conditional $p_\theta(x \mid z)$ is anticipated to be complex, neural network is widely selected

A Probabilistic Approach for Generative Model

Recalling MLE, our objective is maximizing

$$p_\theta(x) = \int p(z)p_\theta(x \mid z)dz$$

where generative model is $p_\theta(x \mid z)$

- Recalling manifold hypothesis, choose prior $p(z)$ to be simple, e.g., Gaussian distribution of reasonable latent attributes z, e.g., pose, degree of smile, ...
- As conditional $p_\theta(x \mid z)$ is anticipated to be complex, neural network is widely selected
- The marginalization $\int$ is intractable $\rightarrow$ variational inference

Intractability

- Data likelihood is intractable due to the integral:

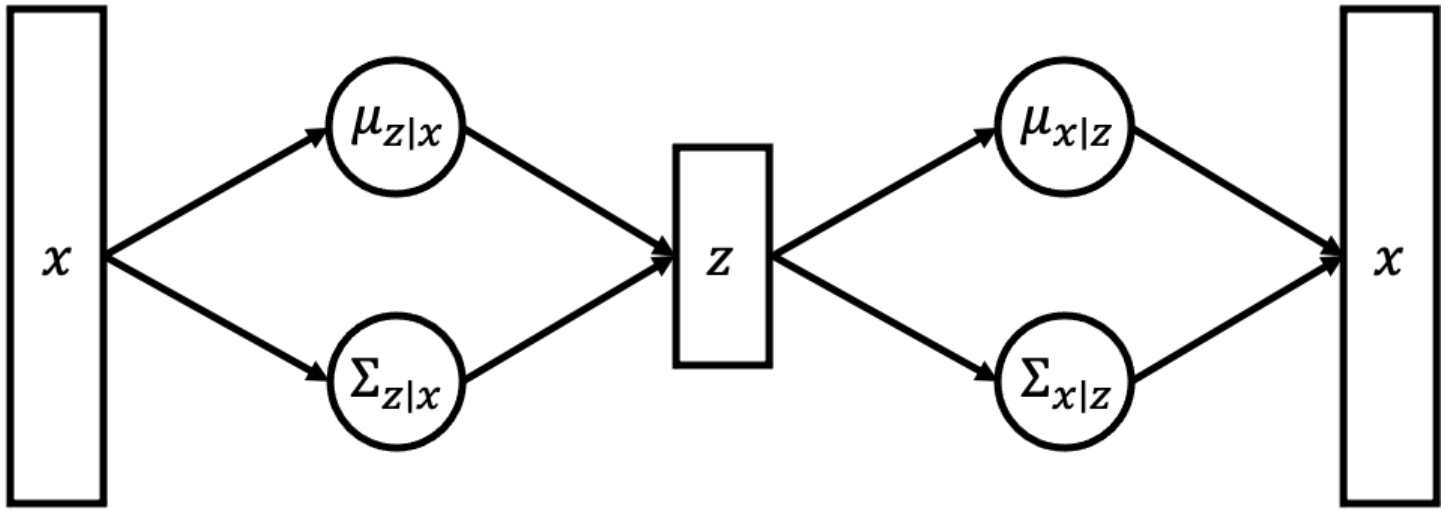
$$p_\theta(x) = \int p(z) p_\theta(x \mid z) dz$$

- Posterior density is also intractable due to the data likelihood:

$$p_\theta(z \mid x) = \frac{p_\theta(x \mid z) p(z)}{p_\theta(x)}$$

- A solution: approximate the posterior $p_\theta(z \mid x)$ using another (encoder) network $q_\phi(z \mid x)$

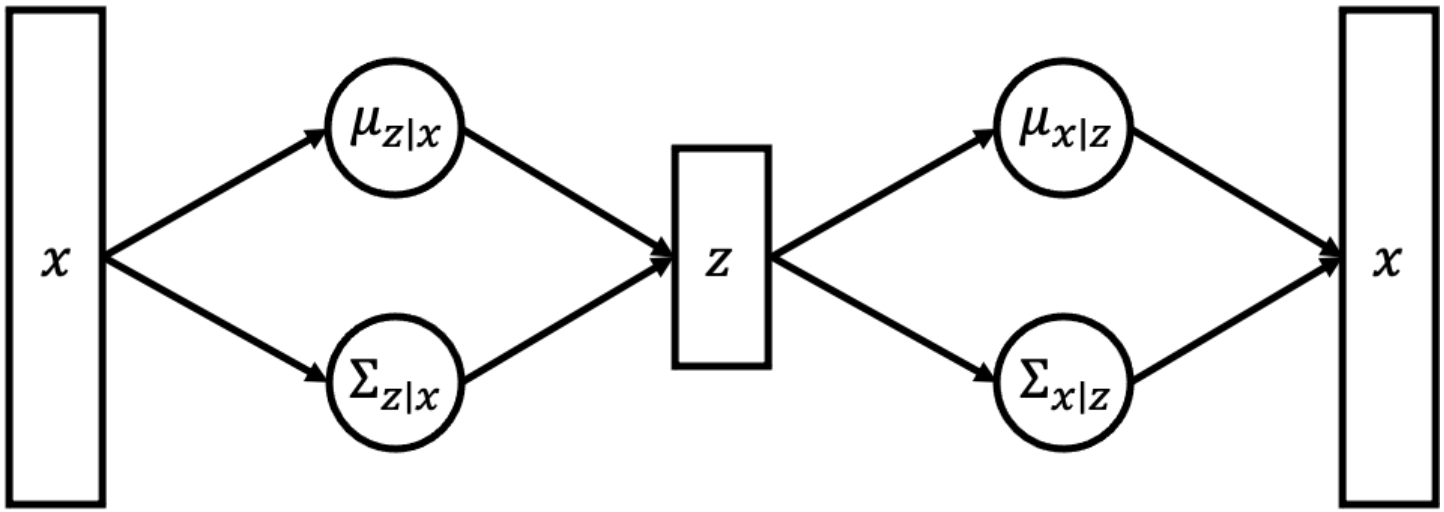
A Probabilistic Framework of Auto-encoder



You can imagine that there is a neural network f_ϕ for encoder, so that

$$f_\phi(x) = \begin{bmatrix} \mu_{z|x} \\ \sigma_{z|x} \end{bmatrix}, \quad \text{and} \quad q_\phi(z|x) = \mathcal{N}\left(f_\phi(x)_1, (f_\phi(x)_2)^2\right)$$

A Probabilistic Framework of Auto-encoder



For decoder p_θ,

$$g_\theta(z) = \begin{bmatrix} \mu_{x|z} \\ \sigma_{x|z} \end{bmatrix}, \quad \text{and} \quad p_\theta(x|z) = \mathcal{N}\left(g_\theta(z)_1, (g_\theta(z)_2)^2\right)$$

Variational Autoencoder

Recalling we aim at MLE: for given x [1],

$$
\begin{aligned}
\log p_\theta(x) &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x)\right] \\
&= \mathbb{E}_z\left[\log \frac{p_\theta(x \mid z)p(z)}{p_\theta(z \mid x)}\right] \\
&= \mathbb{E}_z\left[\log \frac{p_\theta(x \mid z)p(z)}{p_\theta(z \mid x)}\frac{q_\phi(z \mid x)}{q_\phi(z \mid x)}\right] \\
&= \mathbb{E}_z\left[\log p_\theta(x \mid z)\right] - \mathbb{E}_z\left[\log \frac{q_\phi(z \mid x)}{p(z)}\right] + \mathbb{E}_z\left[\log \frac{q_\phi(z \mid x)}{p_\theta(z \mid x)}\right] \\
&= \mathbb{E}_z\left[\log p_\theta(x \mid z)\right] - \mathrm{KL}(q_\phi(z \mid x)\|p(z)) + \mathrm{KL}(q_\phi(z \mid x)\|p_\theta(z \mid x))
\end{aligned}
$$

where the KL divergences take the expectation w.r.t. $z \sim q_\phi(\cdot \mid x)$.

[1] $p(x)p(z \mid x) = p(x \mid z)p(z)$

Variational Autoencoder

Recalling we aim at MLE: for given x,

$$\log p_\theta(x) = \underbrace{\mathbb{E}_z\left[\log p_\theta(x \mid z)\right]}_{\text{(A)}} - \underbrace{\mathrm{KL}(q_\phi(z \mid x) \| p(z))}_{\text{(B)}} + \underbrace{\mathrm{KL}(q_\phi(z \mid x) \| p_\theta(z \mid x))}_{\text{(C)}}$$

- Term (A) is tractable as we can sample $z \sim q_\phi(\cdot \mid x)$ from the encoder, and compute $p_\theta(x \mid z)$ from the decoder.
- Term (B) is tractable as the KL divergence between Gaussians has a closed-form
- Term (C) is intractable, while we know it is non-negative thanks to Gibbs' inequality ($KL \geq 0$)
- Hence, define (A)+(B) as variational lower bound $\mathcal{L}(x, \theta, \phi)$ (ELBO: Evidence Lower BOund [2]) and maximize it

[2] c.f. EM slides p. 12

Training VAE

Training VAE:

$$\arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)}, \theta, \phi)$$

Understanding ELBO:

$$\begin{aligned}\log p_\theta(x) &\geq \mathcal{L}(x, \theta, \phi) \\ &= \mathbb{E}_z \left[\log p_\theta(x \mid z)\right] - \mathrm{KL}(q_\phi(z \mid x) \| p_\theta(z))\end{aligned}$$

- $\mathbb{E}_z \left[\log p_\theta(x \mid z)\right]$ for reconstruction
- $\mathrm{KL}(q_\phi(z \mid x) \| p(z))$ for regularization to make the approximate posterior close to the prior

Training VAE: Monte Carlo Method

Let's simplify the model by assuming $x, z \in \mathbb{R}$,

$$q_\phi(z|x) \sim \mathcal{N}(z \mid f_\phi(x), \sigma_z^2) \quad \text{and} \quad p_\theta(x|z) \sim \mathcal{N}(x|g_\theta(z), \sigma_x^2)$$

where $f_\phi(x)$ is a function of x parameterized by ϕ, and $g_\theta(z)$ is a function of z parameterized by θ.

The first term of ELBO has no analytic solution:

$$\mathbb{E}_z \left[\log p_\theta(x \mid z)\right] = \int q_\phi(z|x) \log p_\theta(x \mid z) dz$$

We can approximate the expectation with Monte-Carlo method:

$$\mathbb{E}_z \left[\log p_\theta(x \mid z)\right] \approx \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x \mid z^{(i)})$$

where $z^{(1)}, z^{(2)}, ..., z^{(N)}$ are samples drawn from $q_\phi(z|x)$.

Training Decoder p

Given the Monte Carlo approximation

$$\mathbb{E}_z \left[\log p_\theta(x \mid z)\right] \approx \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x \mid z^{(i)})$$

$$= -\log \sigma_x^2 \sqrt{2\pi} - \frac{1}{2N} \sum_{i=1}^{N} \frac{(x - g_\theta(z^{(i)}))^2}{\sigma_x^2}$$

we can approximate the derivative w.r.t θ[3].

For example, if $g_\theta(z) = \theta_1 z + \theta_0$ where $\theta_1, \theta_0 \in \mathbb{R}$,

$$\frac{\partial \, \mathbb{E}_z \left[\log p_\theta(x \mid z)\right]}{\partial \theta_1} \approx \frac{1}{N} \sum_{i=1}^{N} \frac{(x - g_\theta(z^{(i)}))z^{(i)}}{\sigma_x^2}$$

[3] The same procedure can be applied if g_θ is a NN parameterized by θ.

Training Encoder q

Again, given the Monte Carlo approximation

$$\mathbb{E}_z \left[\log p_\theta(x \mid z)\right] \approx \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x \mid z^{(i)})$$

$$= -\log \sigma_x^2 \sqrt{2\pi} - \frac{1}{2N} \sum_{i=1}^{N} \frac{(x - g_\theta(z^{(i)}))^2}{\sigma_x^2}$$

we cannot approximate the derivative w.r.t ϕ in this case.

Why? the distribution q is replaced by its samples!

$\Rightarrow$ Reparameterization is a key trick to train VAE!

Reparameterization Trick

Some random variables can be represented as a function of another variable.

For example, assume $Z \sim \mathcal{N}(\mu, \sigma^2)$.

The distribution of Z can be explained by the standard normal distribution as

$$Z = \sigma\epsilon + \mu, \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, 1)$$

We can also take a sample of Z using the sample from $\mathcal{N}(0, 1)$ via $z^{(i)} = \sigma\epsilon^{(i)} + \mu$

Training Encoder with Reparameterization

Recall $q_\phi(z|x) \sim \mathcal{N}(z \mid f_\phi(x), \sigma_z^2)$.

Using reparam $z^{(i)} = \epsilon^{(i)}\sigma_z + f_\phi(x)$, the expectation can be rewritten as

$$\begin{aligned}
\mathbb{E}_z\left[\log p_\theta(x \mid z)\right] &\approx \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(x \mid z^{(i)}) \\
&= -\log \sigma_x^2\sqrt{2\pi} - \frac{1}{2N}\sum_{i=1}^{N}\frac{(x - g_\theta(z^{(i)}))^2}{\sigma_x^2} \\
&= -\log \sigma_x^2\sqrt{2\pi} - \frac{1}{2N}\sum_{i=1}^{N}\frac{(x - g_\theta(\epsilon^{(i)}\sigma_z + f_\phi(x)))}{\sigma_x^2}
\end{aligned}$$

Then the partial derivative w.r.t. ϕ can be computed via

$$\frac{\partial\, \mathbb{E}_z\left[\log p_\theta(x \mid z)\right]}{\partial \phi} \approx \frac{1}{N}\sum_{i=1}^{N}\frac{(x - g_\theta(z^{(i)}))}{\sigma_x^2}\frac{\partial g_\theta}{\partial \phi}$$

KL Divergence

The second term in ELBO, i.e., $\mathrm{KL}(q_\phi(z \mid x) \| p(z))$, has an analytic solution if both q and p follows the normal distribution:

$$\int q_\theta(z|x) \log p(z) dz = \int \mathcal{N}(z; \mu, \sigma^2) \log \mathcal{N}(z; 0, I) dz$$
$$= -\frac{J}{2} \log 2\pi - \frac{1}{2} \sum_{j=1}^{J} (\mu_j^2 + \sigma_j^2)$$

where J is a dimensionality of z, μ and σ is a function of x.

We can easily compute the derivatives w.r.t μ and σ.

Training VAE: Summary

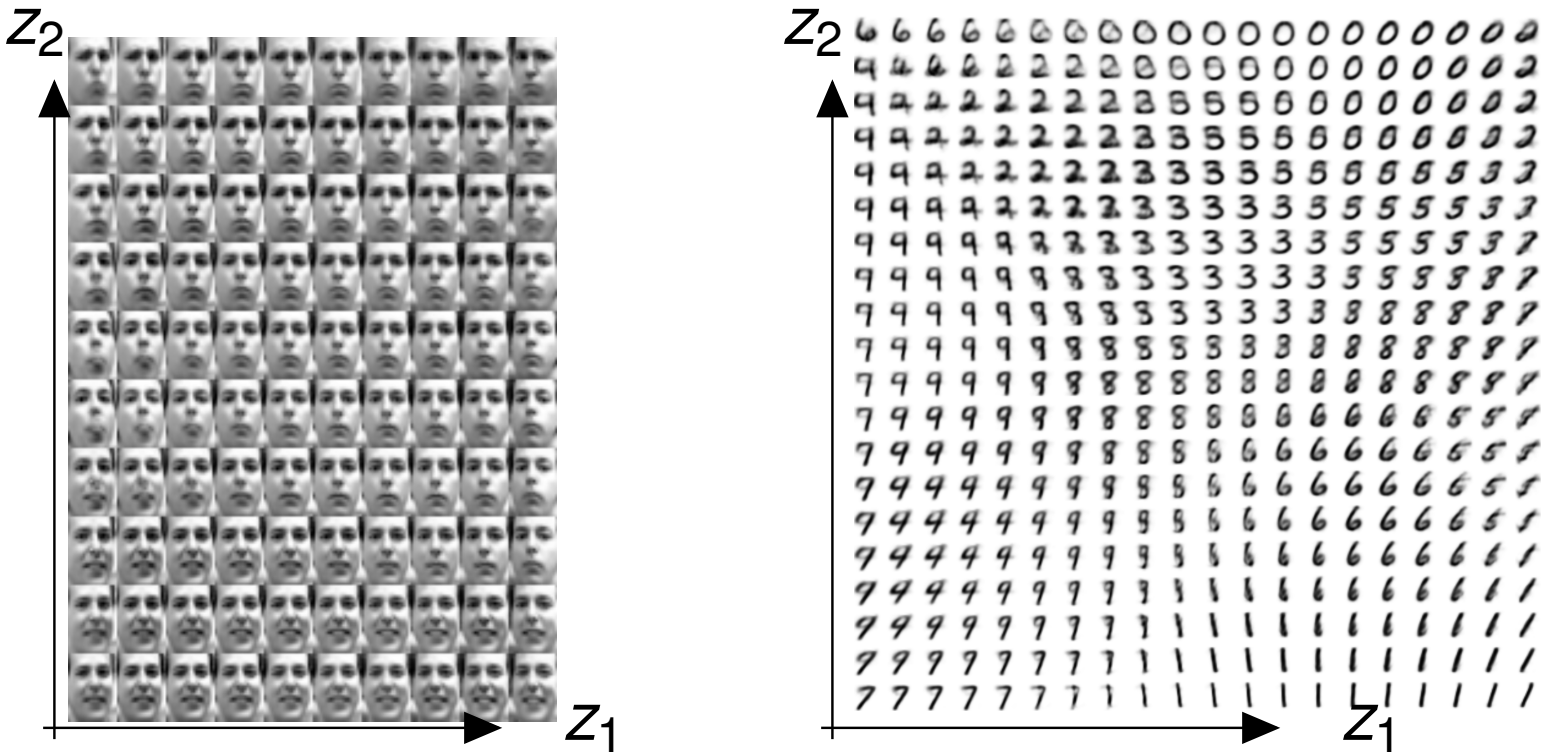
Training VAE via ELBO:

$$\begin{aligned}\log p_\theta(x) &\geq \mathcal{L}(x, \theta, \phi) \\ &= \mathbb{E}_z \left[\log p_\theta(x \mid z)\right] - \mathrm{KL}(q_\phi(z \mid x) \| p(z)) \\ &\approx \frac{1}{N} \sum_{n=1}^{N} \left[\log p_\theta(x \mid z^{(n)})\right] - \mathrm{KL}\left(q_\phi(z^{(n)} \mid x) \| p(z^{(n)})\right)\end{aligned}$$

- $\partial \mathcal{L}(x, \theta, \phi)/\partial \theta$ is simple given samples from $q(z|x)$
- $\partial \mathcal{L}(x, \theta, \phi)/\partial \phi$ requires reparameterization trick.

Generating Data from VAE

Use the decoder network with z sampled from prior $\mathcal{N}(0, I)$



[Kingma and Welling 14]

- Similar z implies similar output x
- It is interesting to see that in the left, $z_1 \approx$ head pose, and $z_2 \approx$ degree of smile

Pros and Cons of VAE

Pros:

- A principled approach to generative models
- Encoder $q_\theta(z \mid x)$, which can be useful for other tasks, e.g., semisupervised learning

Cons:

- A variational inference (approximating likelihood using ELBO)
- Blurry and low quality generations compared to state-of-the-art, e.g., GANs