Generative Models I

Jisang Han

onground@korea.ac.kr

KUGODS

Department of Computer Science and Engineering, Korea University

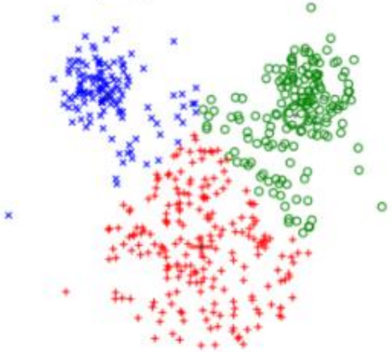
Korea University GOogle Developer Students

KUGODS

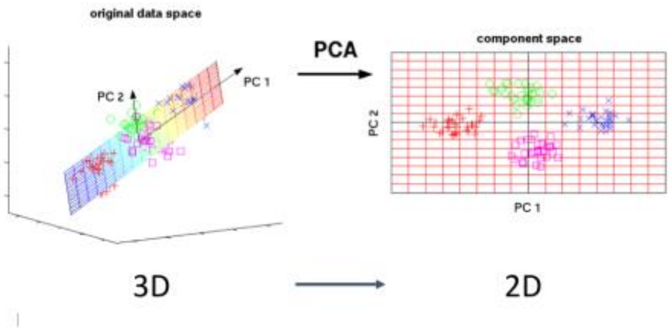
Supervised vs Unsupervised

	Supervised Learning	Unsupervised Learning
Data	(x, y) : x is data, y is label	x
Goal	$x \rightarrow y$ 로 가는 function을 학습	Learn some underlying hidden structure of the data
Examples	Classification, regression, object detection, semantic segmentation, image captioning, etc.	Clustering, dimensionality reduction, feature learning, density estimation, etc.

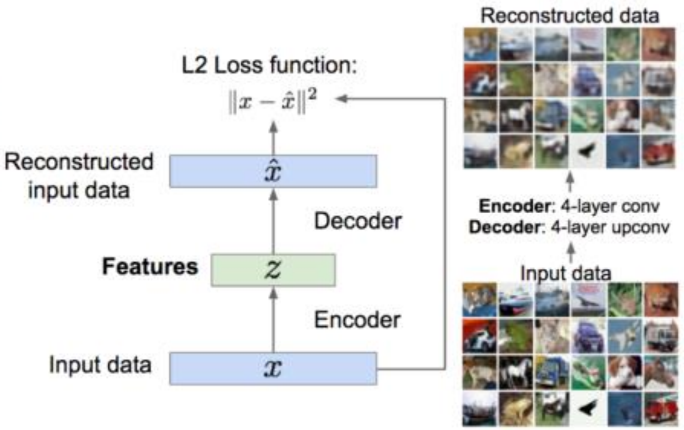
Clustering
(e.g. K-Means)



Dimensionality Reduction
(e.g. Principal Components Analysis)



Feature Learning
(e.g. autoencoders)



Discriminative vs **Generative** Models

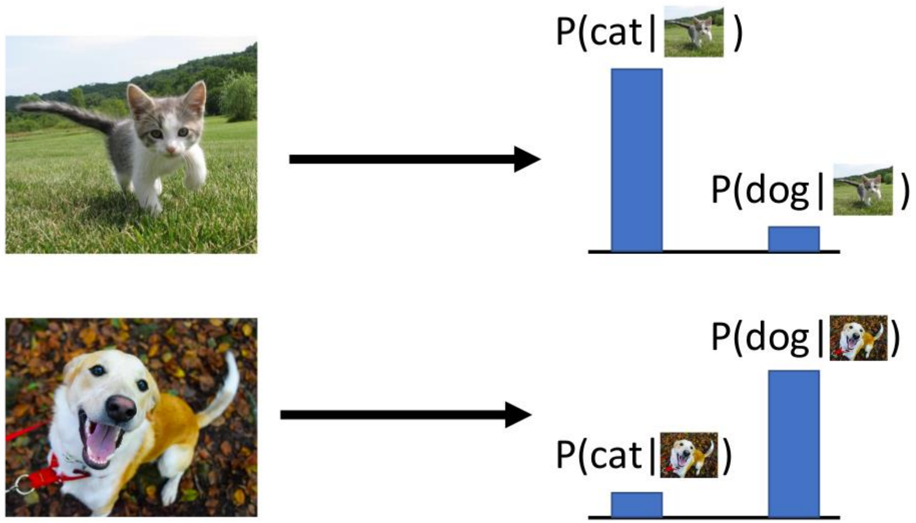
Density Function

$$\int_X p(x)\, dx = 1$$

- Density Functions are normalized.
- Sum of probabilities are 1. If one is bigger then the other is smaller(compete each other).

Discriminative vs **Generative** Models

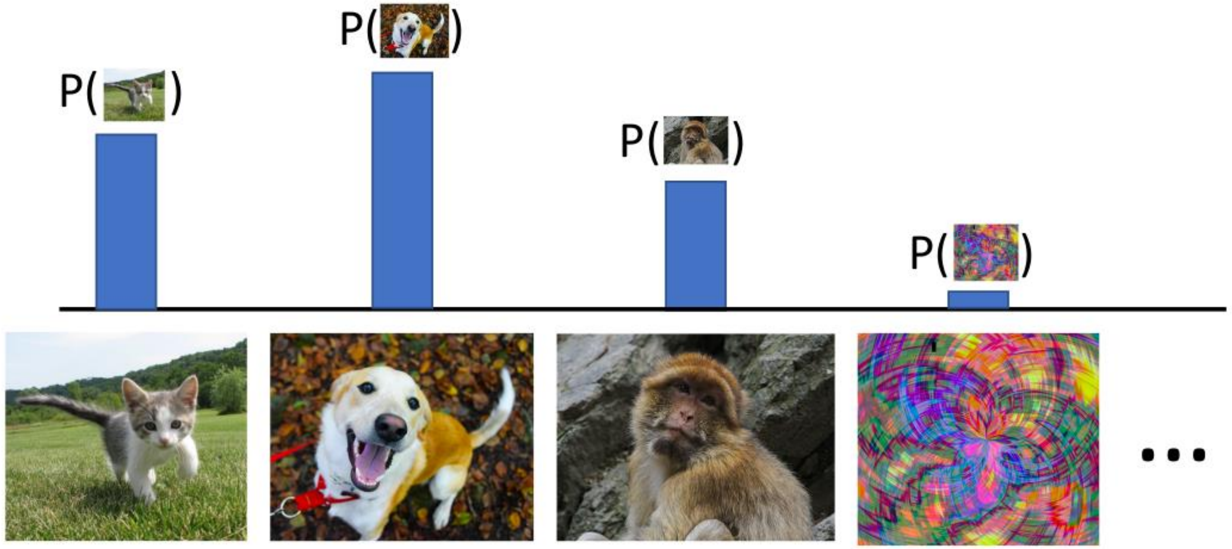
Discriminative Model



- **Learn a probability distribution p(y|x) that predicts probability of the label y conditioned on the input image x**
- Input is x and Output is the probability. (x에 대한 label이 나올 확률)
- The probability is calculated even if an input different from the specified labels is given.

Discriminative vs **Generative** Models

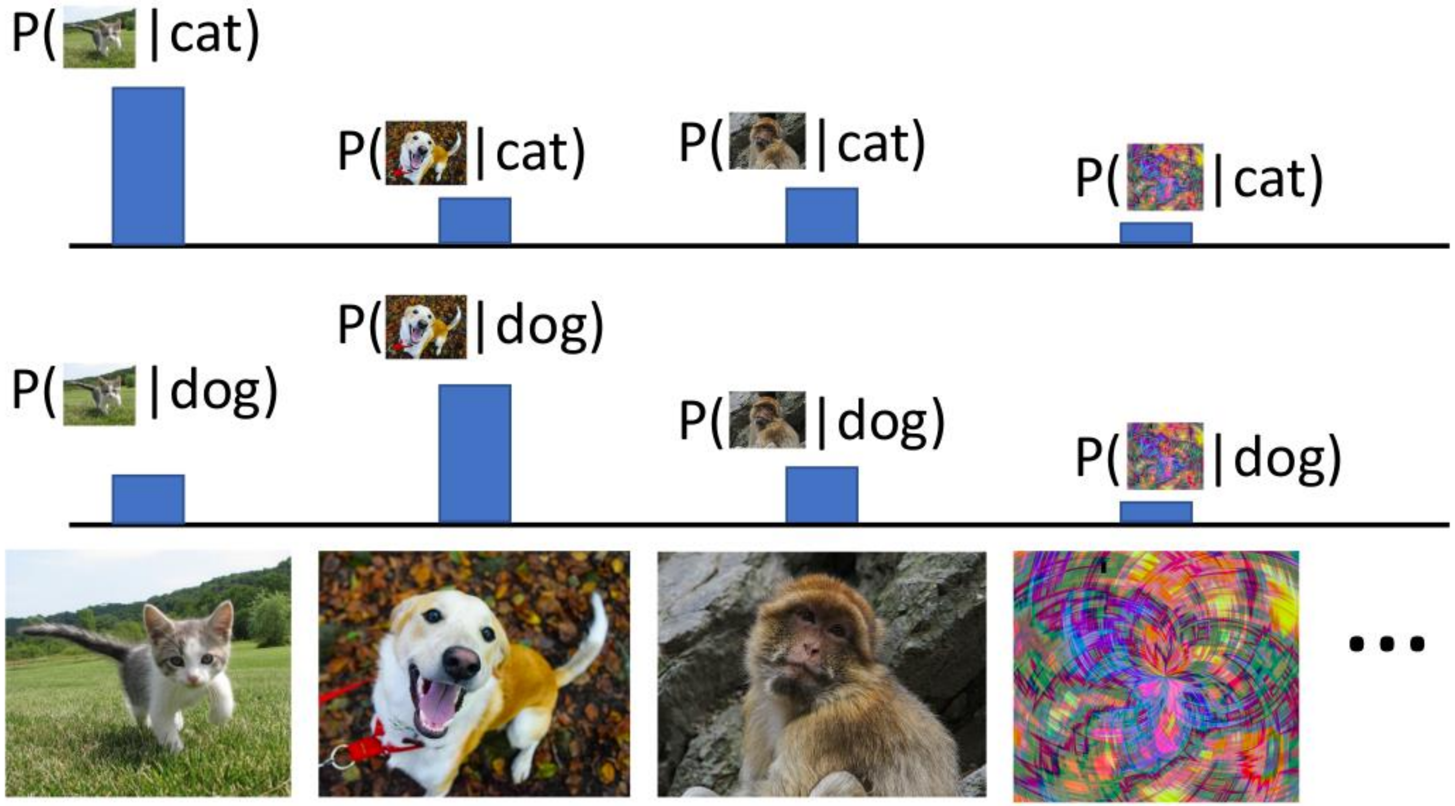
Generative Model



- **Learn a probability distribution p(x)**
- All possible images compare to the probability mass.
- A deep understanding of the image is needed. (Which is more plausible, sitting a dog or standing up?)
- The model can reject irrational inputs by giving them a very small value.

Discriminative vs **Generative** Models

Conditional Generative Model



Recall **Bayes' Rule:**

$$\underbrace{\boxed{P(x \mid y)}}_{\text{Conditional Generative Model}} = \frac{\overbrace{\boxed{P(y \mid x)}}^{\text{Discriminative Model}}}{\underbrace{\boxed{P(y)}}_{\text{Prior over labels}}} \overbrace{\boxed{P(x)}}^{\text{(Unconditional) Generative Model}}$$

- **Learn p(x|y)**

Discriminative vs **Generative** Models

Discriminative Model:
Learn a probability distribution p(y|x) → Assign labels to data
Feature learning (supervised)

Generative Model:
Learn a probability distribution p(x) → Detect outliers
Feature learning (unsupervised)
Sample to **generate** new data

Conditional Generative Model: Learn p(x|y) → Assign labels, while rejecting outliers!
Generate new data conditioned on input labels

Autoregressive Models

Explicit Density Estimation

Goal : Write down an explicit function for $p(x) = f(x, W)$ (x: data, W: learnable weight matrix)

If dataset is $x^{(1)}, x^{(2)}, \dots, x^{(N)}$,

$$W^* = \arg\max_W \prod_i p(x^{(i)})$$

Maximize the probability of training data

$$= \arg\max_W \sum_i \log p(x^{(i)})$$

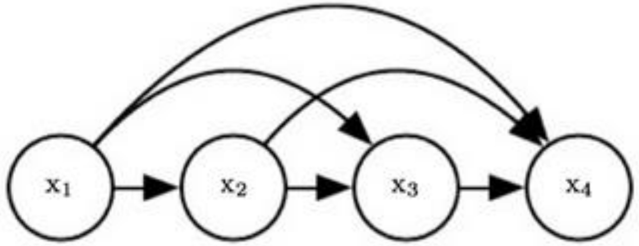
$$= \arg\max_W \sum_i \log f(x^{(i)}, W)$$

Loss Function. (Gradient Descent)

Autoregressive Models

Explicit Density: Autoregressive Models

- 자기 자신을 입력으로 하여 자기 자신을 예측하는 모형



- x가 여러 subparts로 이루어져있다고 가정. x가 이미지라고 하면 subparts는 각 픽셀이다 $x = (x_1, x_2, x_3, ..., x_T)$
- Chain rule을 사용하여 probability를 계산한다.

$$
\begin{aligned}
p(x) &= p(x_1, x_2, ..., x_T) \\
&= p(x_1)\, p(x_2|x_1)\, p(x_3|x_1, x_2) \\
&= \prod_{t=1}^{T} p(x_t|x_1, ..., x_{t-1})
\end{aligned}
$$

Autoregressive Models

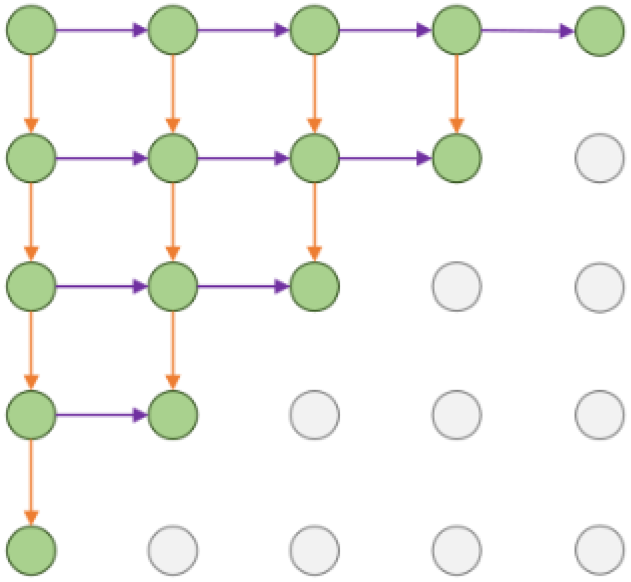
Explicit Density: Autoregressive Models

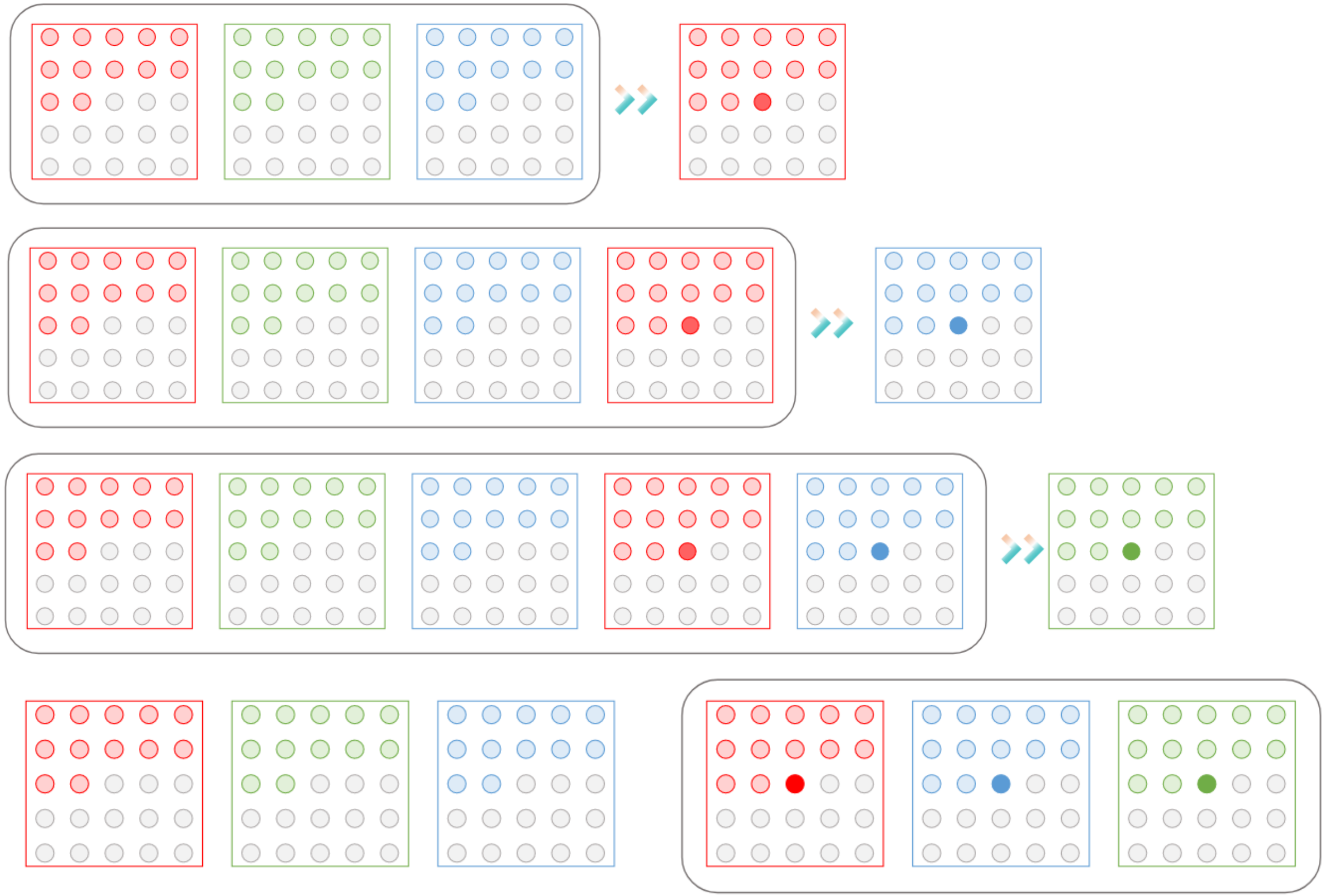
$$
\begin{aligned}
p(x) &= p(x_1, x_2, \dots, x_T) \\
&= p(x_1)\, p(x_2|x_1)\, p(x_3|x_1, x_2) \\
&= \prod_{t=1}^{T} p(x_t|x_1, \dots, x_{t-1})
\end{aligned}
$$

- 즉, sequence의 확률 $p(x)$는 이전 sequence가 주어졌을 때 다음 sequence가 나올 확률을 전부 곱한 것이다.
- https://wikidocs.net/22034

Autoregressive Models

PixelRNN

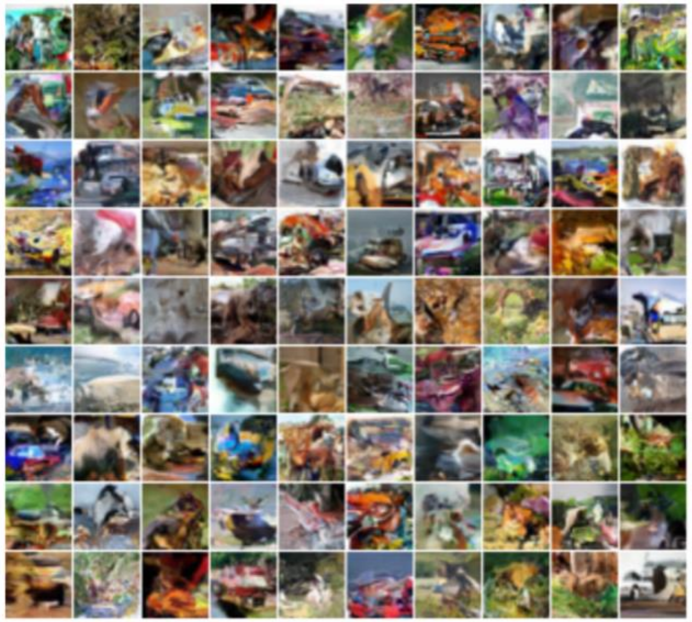




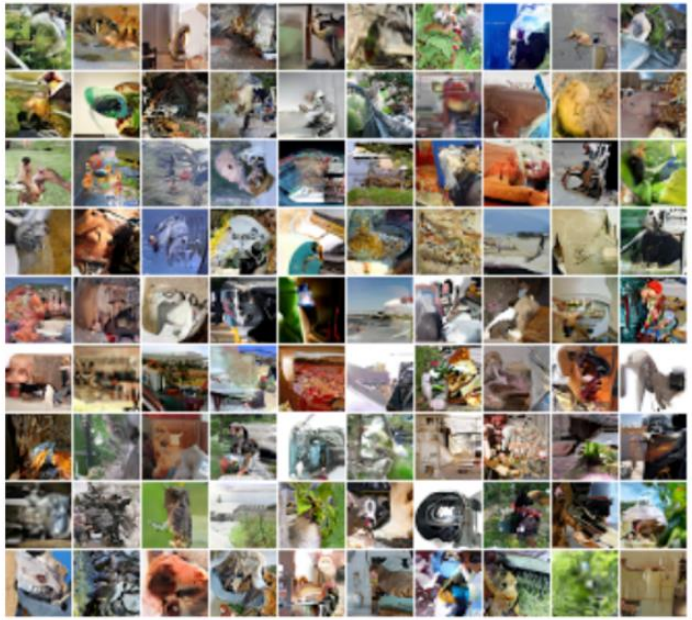
http://dmqm.korea.ac.kr/uploads/seminar/20190705_Autoregressive.pdf

Autoregressive Models

PixelRNN



32x32 CIFAR-10



32x32 ImageNet

- 겉으로 보기에는 합리적으로 보이지만 사실 자세히 보면 거지같다..
- 그래도 edges, colors를 꽤나 그럴싸하게 생성하는 것에 의의를 가진다.
- 이는 unconditional generation이므로 test time에 내가 무엇을 생성하고 있는지 정할 수 없다

Autoregressive Models

Pros and Cons

Pros

- Likelihood p(x)를 명시적으로 계산할 수 있다.
- 위의 장점 덕분에 좋은 evaluation metric을 얻는다. (training data와 유사한, 즉 얼마나 그럴듯한지에 대한 p(x) 값을 얻을 수 있기 때문이다.)
- 나름 괜찮은 결과를 얻을 수 있다. (진짜 사진같지는 않지만..)

Cons

- Sequential Generation 이므로 느리다.

Variational Autoencoders (VAE)

- In PixelRNN, PixelCNN, they defined parametric density function $p(x) = f(x, W)$ and calculate for each input. And train the model to maximize this output.
- In VAE, Instead of maximizing the actual density value, **maximize the lower bound of the density.**
- $p_\theta(x) = \prod_{i=1}^{n} p_\theta(x_i | x_1, \ldots, x_{i-1})$

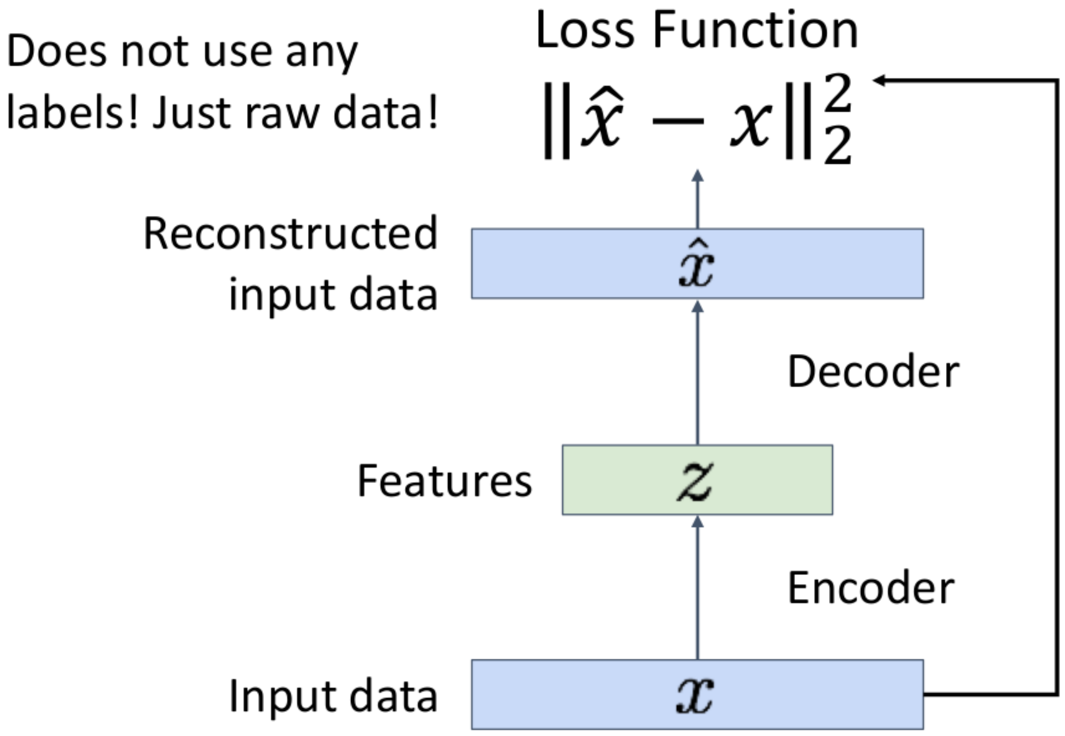
Variational Autoencoders (VAE)

(Regular, non-variational) Autoencoders

- Unsupervised method로, labels 없이 raw data x로부터 feature vectors를 학습한다.

(Unsupervised method for learning feature vectors from raw data x, without any labels)

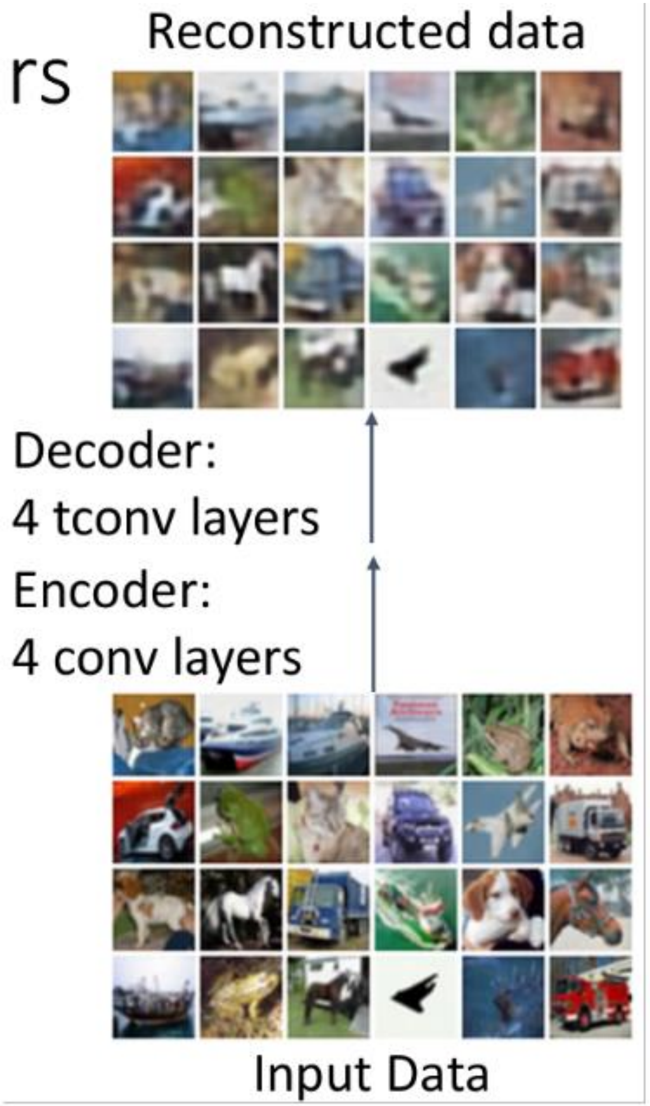
- **Features** extracts useful information that can be used for downstream tasks.
- **Encoder** extracts features from input data,
- **Decoder** reconstruct the input data from the features.
- Encoder:
  - Originally: Linear + nonlinearity (sigmoid)
  - Later: Deep, fully-connected
  - Later: ReLU CNN (upconv)



Variational Autoencoders (VAE)

(Regular, non-variational) Autoencoders

- We expect the effect of **compressing input data through Encoder.**
- After learning, discard the decoder and use it for the downstream task using the encoder.
- **Not probabilistic** : 학습하지 않은 new data를 sampling할 수 없다.



Variational Autoencoders (VAE)

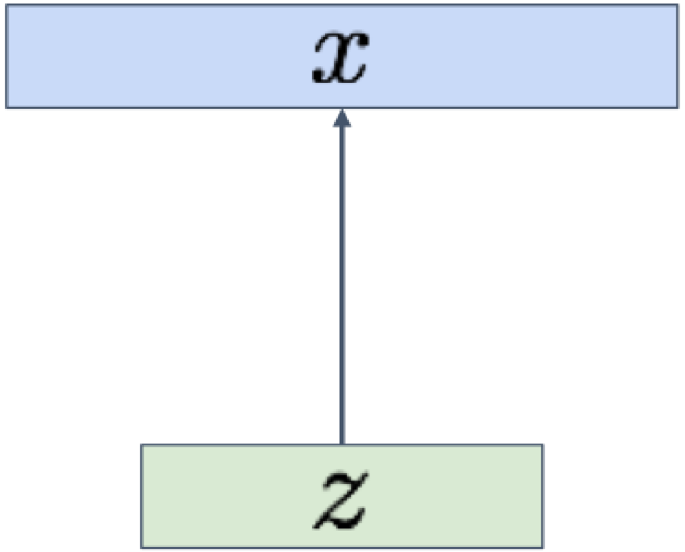
Variational Autoencoders

Sample from conditional

$$p_{\theta^*}(x \mid z^{(i)})$$

Sample z from prior

$$p_{\theta^*}(z)$$



- Autoencoder에 확률 개념을 도입하였다
- 1. raw data로부터 **latent features z**를 학습한다.
- 2. new data를 생성하기 위해 model로부터 **sampling**한다.

Variational Autoencoders (VAE)

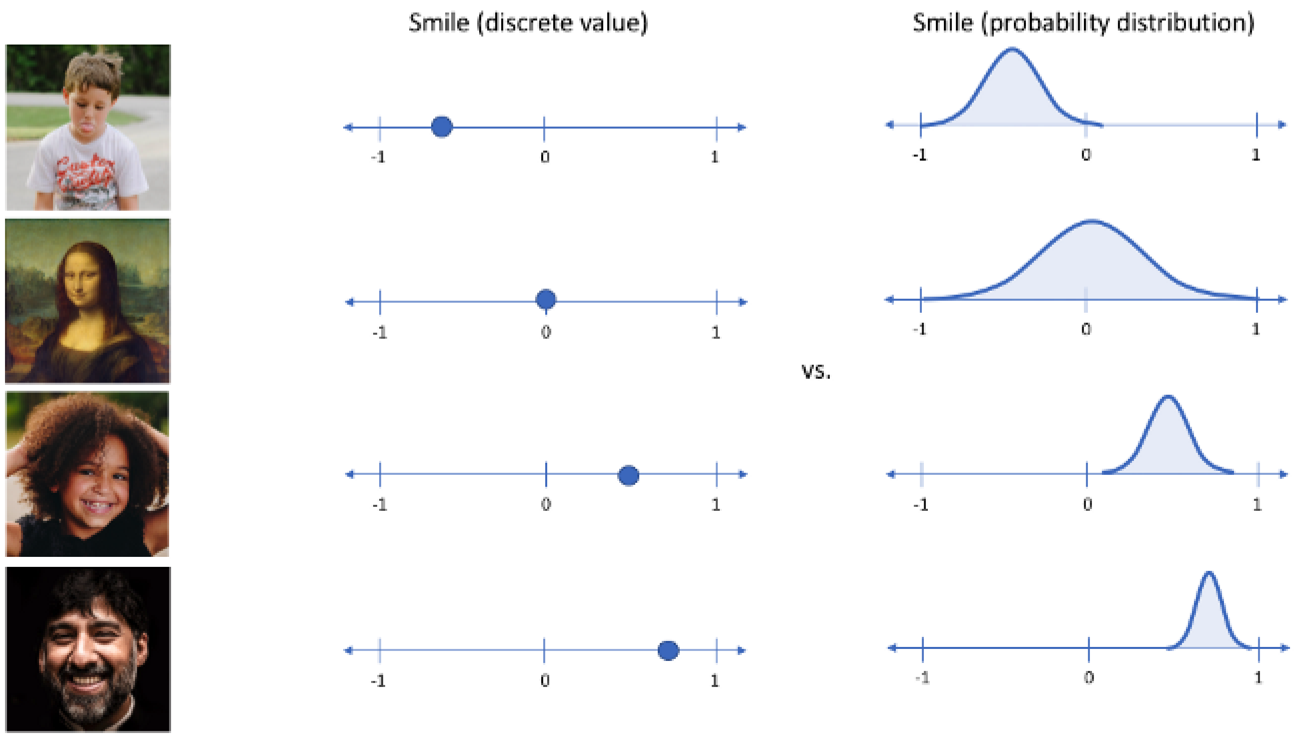
Variational Autoencoders

- **Decoder**: Generating new data x from latent features z that is similar to input data but completely new.
- Sampling latent variables from prior distribution $p_{\theta_*}(z)$, put sampled z into the decoder to predict image x
- At this time, output is not a single image but the distribution of images.

- $p_{\theta_*}(z)$: prior distribution. PDF of x. (Gaussian distribution)
- $p_{\theta_*}(x|z^{(i)})$: 주어진 z에서 특정 x가 나올 조건부 확률에 대한 PDF
- θ : parameter

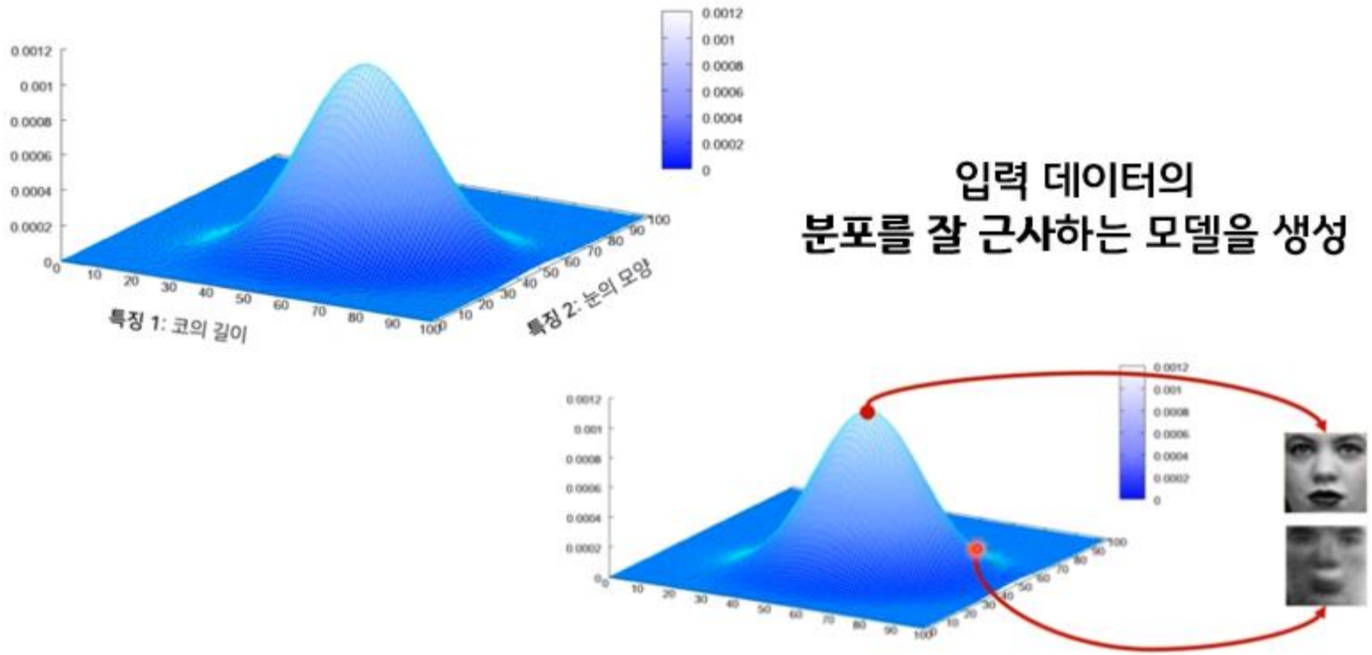
Variational Autoencoders (VAE)

Variational Autoencoders



- VAE에서의 z는 AE에서의 z(a value: low dimension of input data)와 다르게, 가우시안 확률분포에 기반한 확률값으로 나타낸다.
- input image가 들어오면 그 이미지의 다양한 특징들이 각각의 확률변수가 되는 확률분포를 만든다. 이 확률 분포를 잘 찾아내어 확률값이 높은 부분을 사용하면 그럴듯한 새로운 이미지를 생성할 수 있다.

Variational Autoencoders (VAE)



- 이때 각 feature가 가우시안 분포를 따른다고 가정하고 latent z는각 feature의 평균과 분산값을 나타낸다.
- 예를 들어 한국인의 얼굴을 그리기 위해 눈, 코, 입 등의 feature를 Latent vector z에 담고, 그 z를 이용해 그럴듯한 한국인의 얼굴을 그려내는 것이다. latent vector z는 한국인 눈 모양의 평균 및 분산, 한국인 코 길이의 평균 및 분산, 한국인 머리카락 길이의 평균 및 분산 등등의 정보를 담고 있다고 생각할 수 있다.

Thank you!
Q & A