Generative Deep Learning

Sudong Lee

sudonglee@ulsan.ac.kr

https://dais.ulsan.ac.kr/

Contents

- Introduction to Generative Deep Learning
- Variational Autoencoders (VAEs)
- Generative Adversarial Networks (GANs)
- Diffusion Models

Introduction to Generative Deep Learning

A Probabilistic View of Deep Learning: Discriminative vs. Generative Models

"Discriminative and generative models are two fundamental approaches in deep learning, each with distinct characteristics and applications."

- **Discriminative models**
  - Learn the *decision boundary (e.g.*, classification and regression)
  - Focus on the inference of the output, Y, given the input, X.

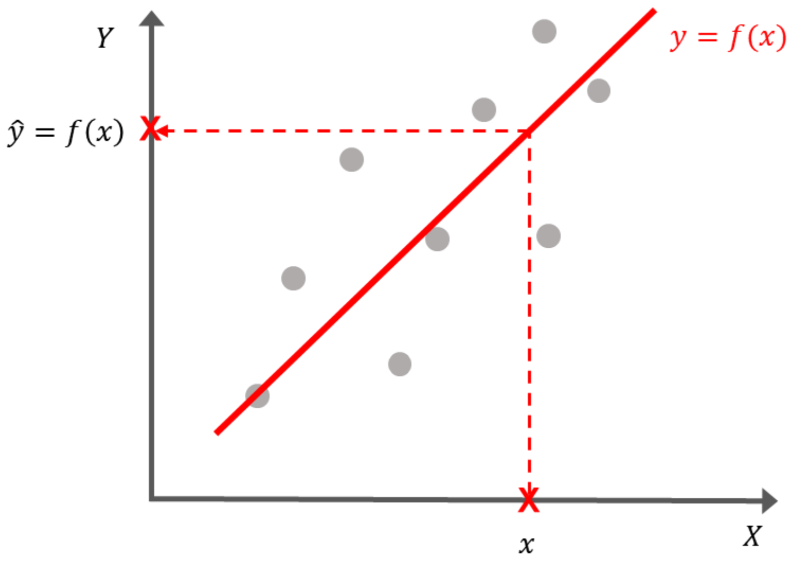
  ⇒ Model the *conditional probability* of the output given the input, $P(Y|X)$.

- **Generative models**
  - Learn the underlying distribution of data.
  - Focus on the generation of new data instances.

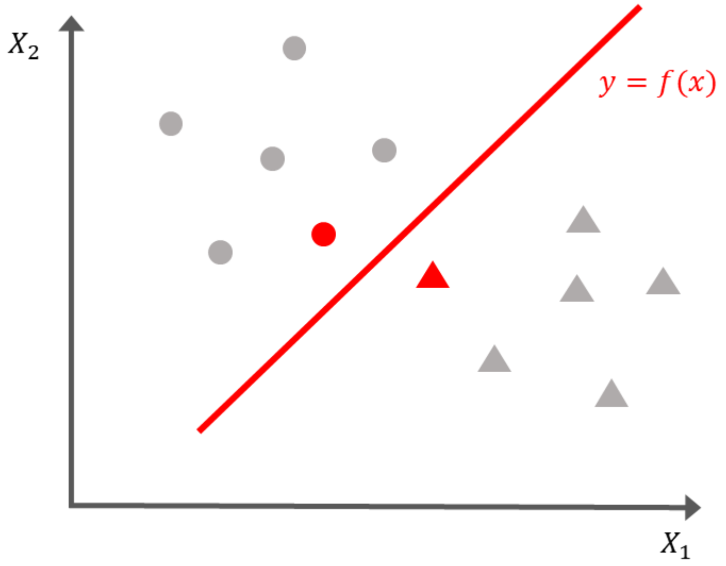
  ⇒ Model the *joint probability* $P(X,Y)$. Depending on the problem, $P(X|Y)$ or $P(X)$ can be of interest.

Discriminative Models

"Discriminative models learn the decision boundary to infer the output given the input."



Regression



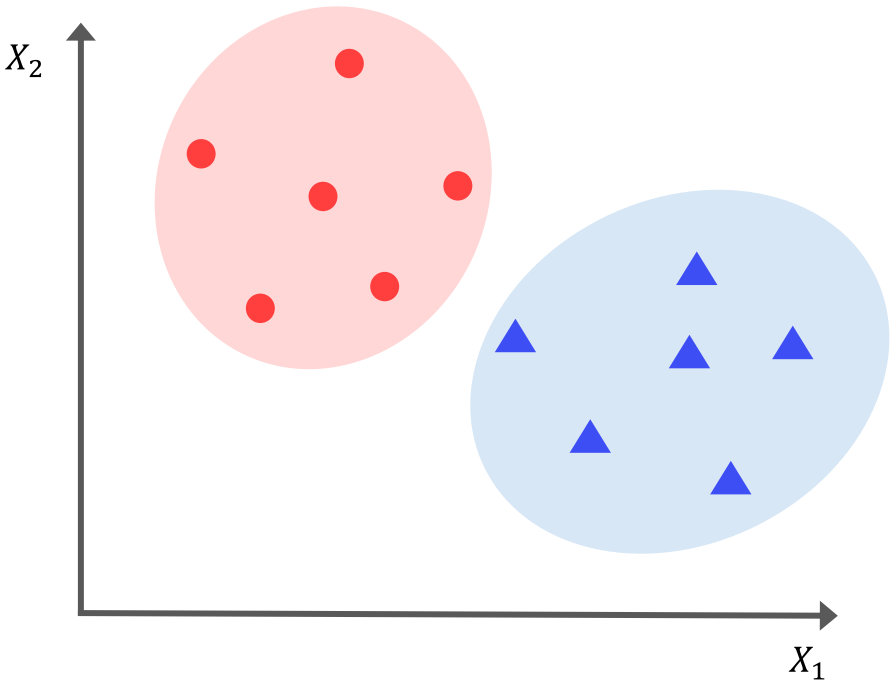
Classification

Applications of Discriminative Models

- ANY classification and regression
- Image classification, object detection, and segmentation
- Natural Language Processing tasks (e.g., sentiment analysis, text classification, …)
- Anomaly detection
- Medical diagnosis and disease prediction
- …

Generative Models

"Generative models learn the underlying distribution of data enabling them to generate new data instances."



Applications of Generative Models

- **Data generation**:

  sample unobserved but realistic new data points (e.g., image/text generation)

- **Density estimation**:

  predict the rareness of events (e.g., anomaly detection)

- **Latent variable extraction**:

  extract low-dimensional latent space for dimensionality reduction (e.g., feature embedding)

- **Simulation and forecasting**

  in various fields (e.g., digital twins, weather prediction, financial modeling, etc.)

- …

A Different Perspective on Generative Models

"While you learn to generate, you can learn the patterns in your data."

Richard Phillips Feynman

The esteemed scientist *Richard Feynman* once said,

"What I cannot create, I do not understand."

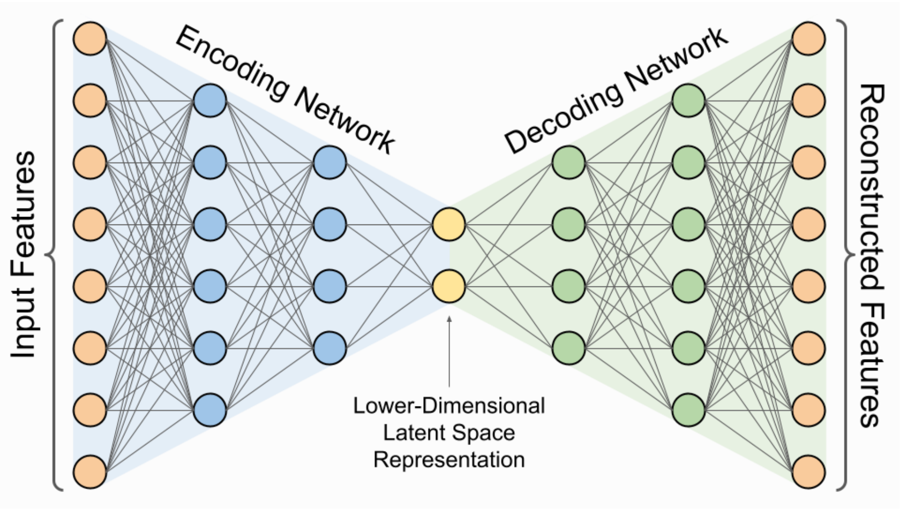
Examples of Deep Generative Models

- Variational Autoencoders (VAEs)
- Generative Adversarial Networks (GANs)
- Autoregressive models (e.g., Transformers like GPT)
- Flow-based models
- Diffusion models
- …

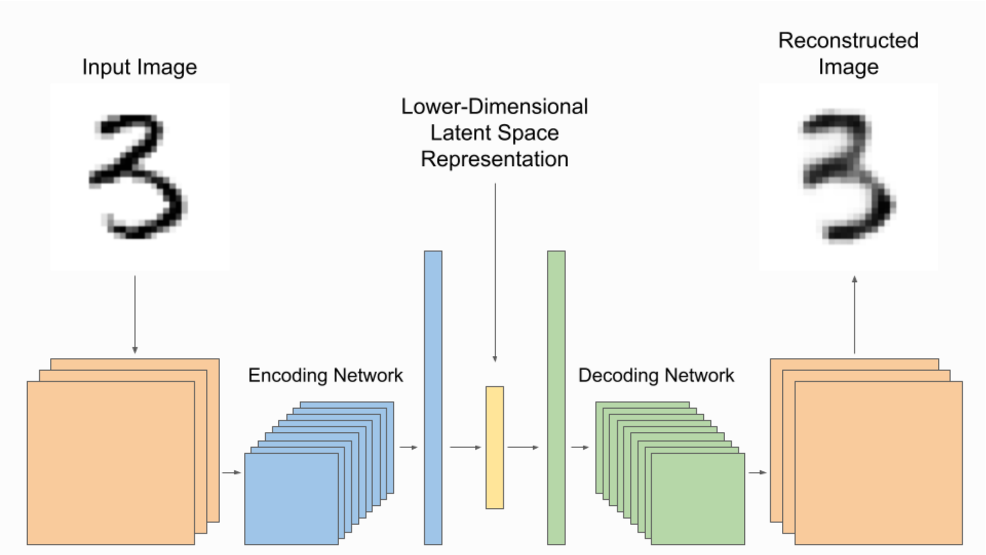
Variational Autoencoders (VAEs)

Autoencoder

"An autoencoder compresses the features into a lower dimension by reconstruction."



Network architecture for an ordinary autoencoder

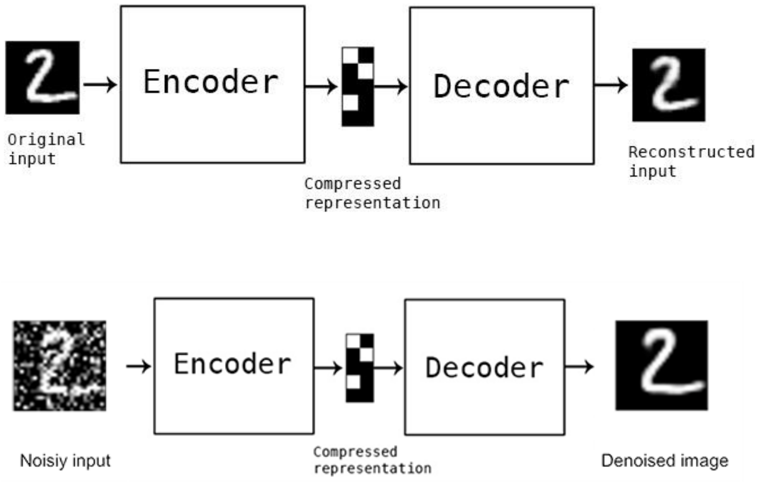


Network architecture for an ordinary convolutional autoencoder

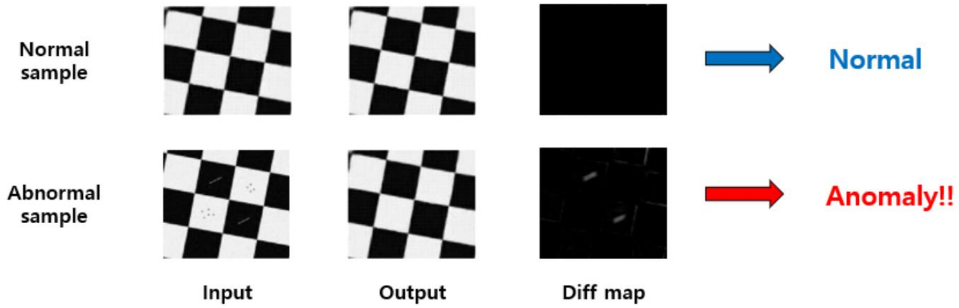
Image Source | https://www.assemblyai.com/blog/introduction-to-variational-autoencoders-using-keras/

Autoencoder

"We can use autoencoders for various applications such as denoising and anomaly detection."



Denoising autoencoder



Autoencoder for anomaly detection

Autoencoders for Data Generation

"Can autoencoders be used for data generation?"

- "Nope. They optimize for *faithful reconstructions*."

  Ordinary autoencoders learn to use the latent space as an embedding space to create **optimal compressions**.

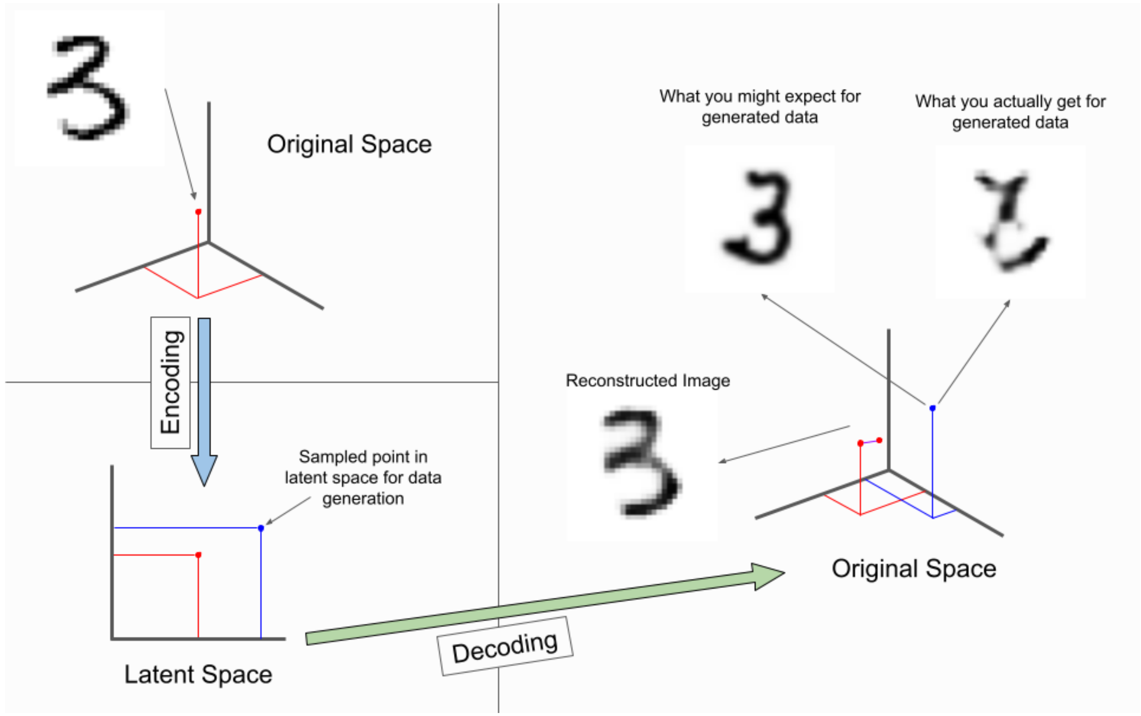


Image Source | https://www.assemblyai.com/blog/introduction-to-variational-autoencoders-using-keras/

Manifold Learning

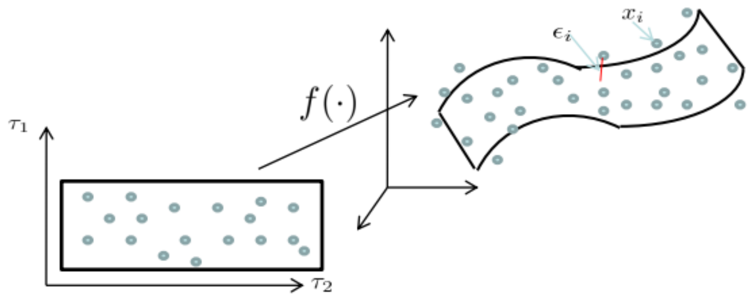
"Manifold learning aims to learn the mapping of high-dimensional data, potentially existing non-linear manifolds,
onto lower-dimensional latent space."

- **Definition.** ***Manifold Learning***
  - A d dimensional manifold $\mathcal{M}$ is embedded in an m dimensional space,
    and there is an explicit mapping $f : \mathbb{R}^d \to \mathbb{R}^m$ where $d \le m$.
  - Given samples $x_i \in \mathbb{R}^m$, we assume that

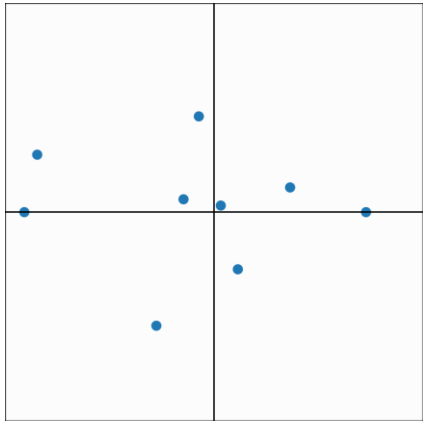
$$x_i = f(\tau_i) + \epsilon_i$$

where f is called mapping function, $\tau_i \in \mathbb{R}^d$ is the embedded feature of x_i, and ϵ_i is the random noise.

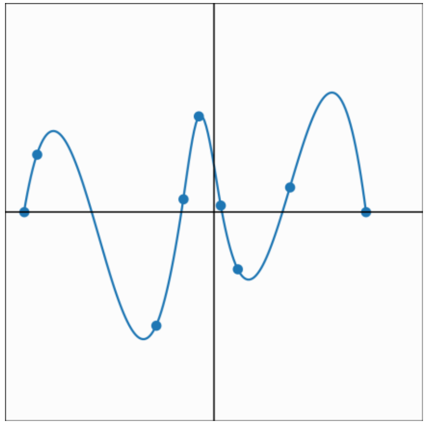
  - We assume $p(\tau)$ is smooth, is distributed uniformly, and noise is small.



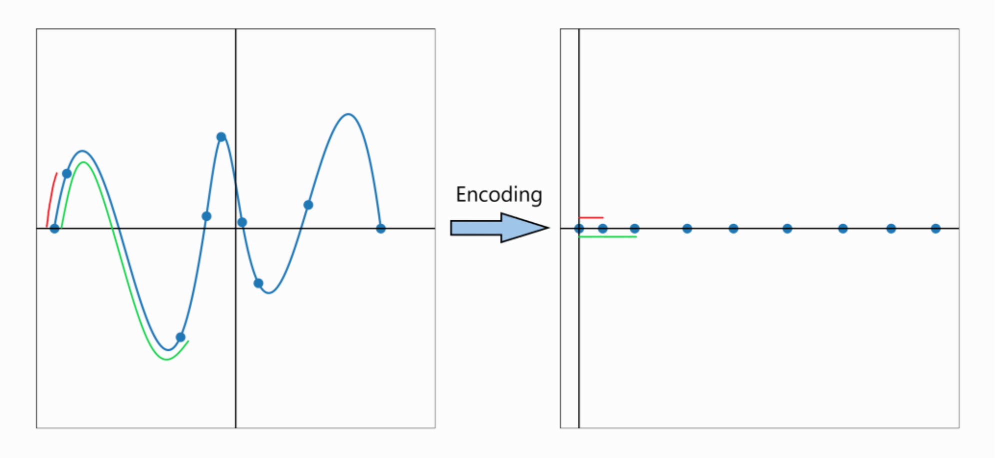
Manifold Learning by Autoencoders as a Compression



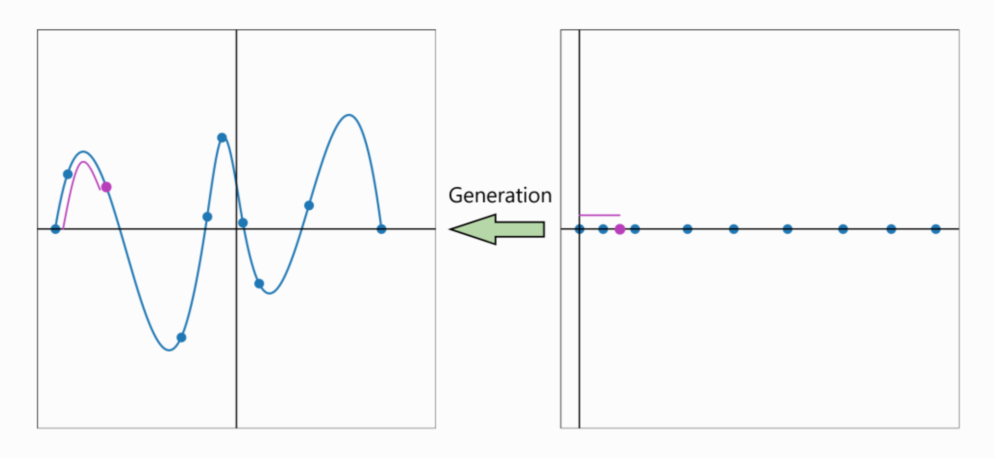
Observations



Hypothesized Manifold

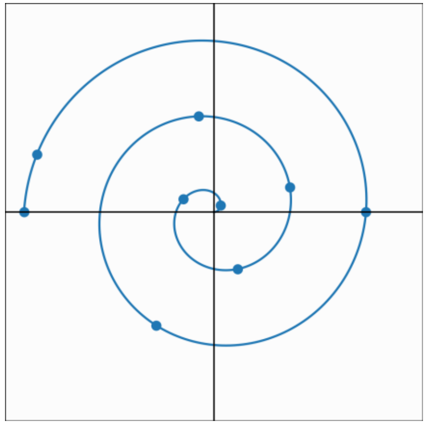


Encoder

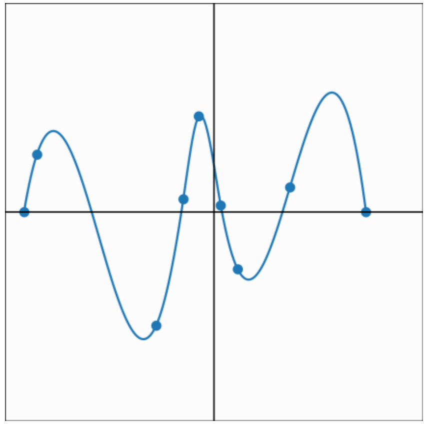


Decoder

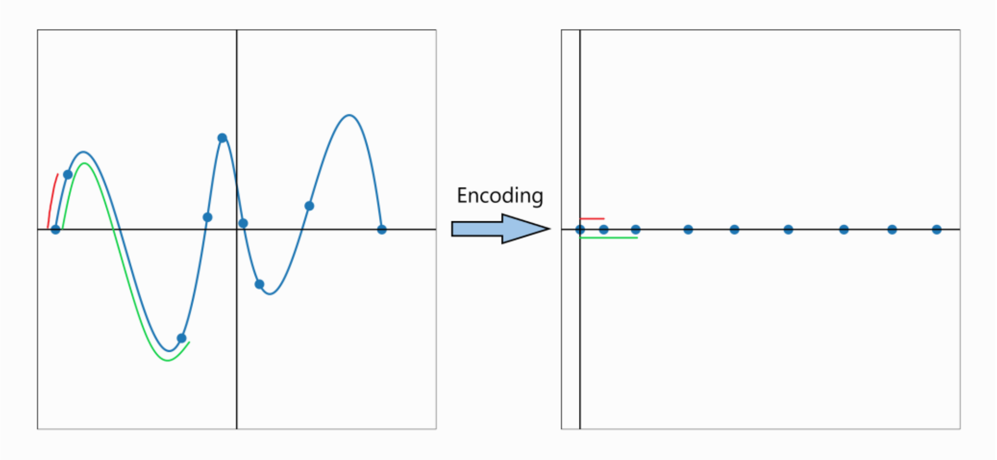
Manifold Learning by Autoencoders as a Compression



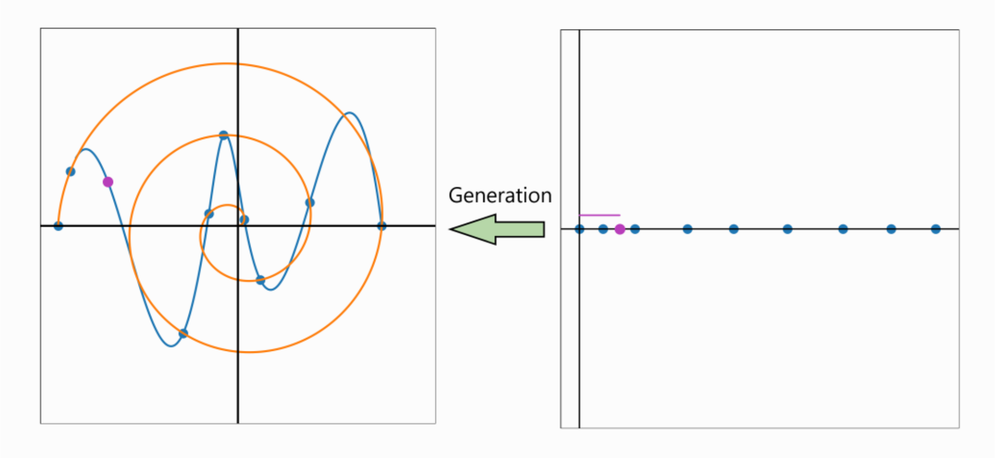
Actual Manifold



Hypothesized Manifold



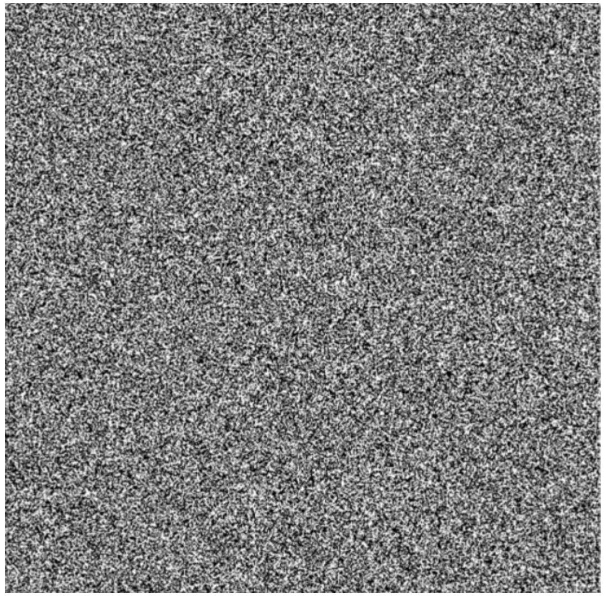
Encoder



Decoder

Data Generation from Noise

Random Noise Sampling



$\boldsymbol{Z}$

$P(\boldsymbol{X}|\boldsymbol{Z})$

Image Generation

$\boldsymbol{X}$

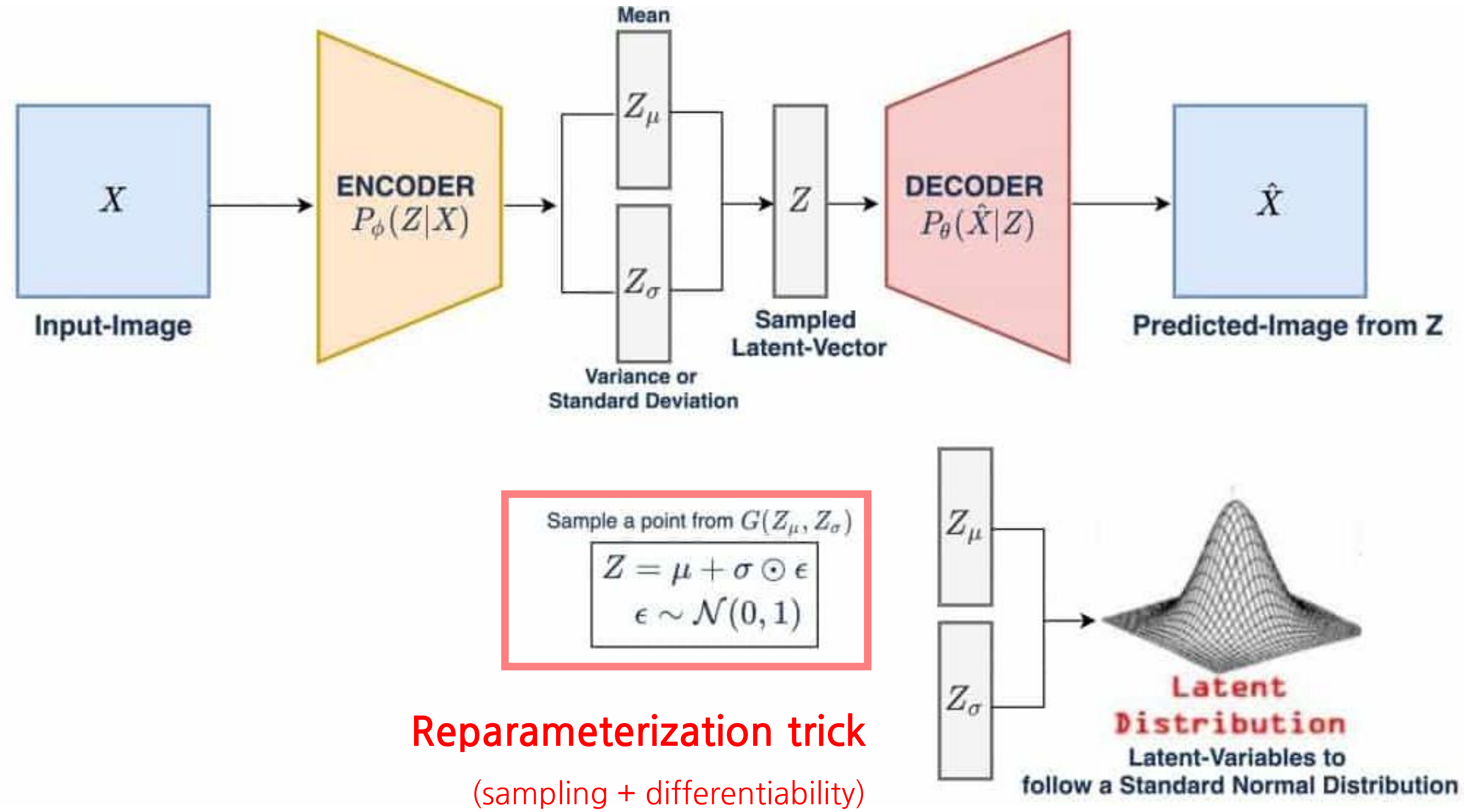
Data Generation from Noise

"How can we effectively encode the *variations* of generated data?"

	z	z	z
Smile	-0.21	0.03	0.56
Skin tone	0.72	0.72	0.72
Gender	-0.63	-0.63	-0.63
Glasses	-0.43	-0.43	-0.43
Hair color	0.44	0.44	0.44

Variational Autoencoder (VAE)

"VAEs encode inputs as distributions and sample from them to produce input variations for data generation."



Reparameterization trick

(sampling + differentiability)

Variational Autoencoder (VAE)

- **Objective function**

$$maximize \log p_\theta(\boldsymbol{x})$$

  - Explicit density modeling:

Prior Likelihood

$$\underbrace{p_\theta(\boldsymbol{x})}_{\text{Marginal likelihood}} = \frac{\overbrace{p_\theta(\boldsymbol{z})}^{\text{Prior}}\ \overbrace{p_\theta(\boldsymbol{x}|\boldsymbol{z})}^{\text{Likelihood}}}{\underbrace{p_\theta(\boldsymbol{z}|\boldsymbol{x})}_{\text{Posterior}}}$$

Marginal likelihood

Posterior → *Intractable!!*

- **Variational Inference**

  - Assume that $q(\boldsymbol{z}|\boldsymbol{x})$ follows a Gaussian distribution.
  - Variational inference approximates $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ using $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. $\rightarrow minimize\ \ D(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x}))$
  - $q^*(\boldsymbol{z}|\boldsymbol{x}) = \arg\min_{q\in Q}(D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})))$ where the Kullback-Leibler divergence $D_{KL}(q||p) = \int p(x)\log\frac{p(x)}{q(x)}dx$.

Bayes' Theorem

"Bayes' theorem gives a mathematical rule for inverting conditional probabilities."

- Bayes' rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Bayesian inference

$$P(\boldsymbol{\theta}|D) = \frac{P(D|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(D)} \propto P(D|\boldsymbol{\theta})P(\boldsymbol{\theta})$$

  - $P(\boldsymbol{\theta}|D)$: posterior
  - $P(D|\boldsymbol{\theta})$: likelihood
  - $P(\boldsymbol{\theta})$: prior

Variational Autoencoder (VAE)

$$maximize \ \log p_\theta(\boldsymbol{x}) = \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}) d\boldsymbol{z}$$

$$= \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{Z})p(z)}{p_\theta(\boldsymbol{z}|\boldsymbol{x})} d\boldsymbol{z}$$

$$= \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{Z})p(z)}{p_\theta(\boldsymbol{z}|\boldsymbol{x})} \frac{q_\theta(\boldsymbol{z}|\boldsymbol{x})}{q_\theta(\boldsymbol{z}|\boldsymbol{x})} d\boldsymbol{z}$$

$$= \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z} - D_{KL}\big(q_\phi(\boldsymbol{z}|\boldsymbol{x}) || p_\theta(\boldsymbol{z})\big) + D_{KL}\big(q_\phi(\boldsymbol{z}|\boldsymbol{x}) || p_\theta(\boldsymbol{z}|\boldsymbol{x})\big) \ \leftarrow \text{Intractable}$$

$$\geq \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z} - D_{KL}\big(q_\phi(\boldsymbol{z}|\boldsymbol{x}) || p_\theta(\boldsymbol{z})\big) \ \leftarrow \textit{Evidence lower bound (ELBO)}$$

Reconstruction error

Regularization

Generative Adversarial Networks (GANs)

Generative Adversarial Network (GAN)

"GAN produces data that is hard to distinguish from the real thing, but not real."

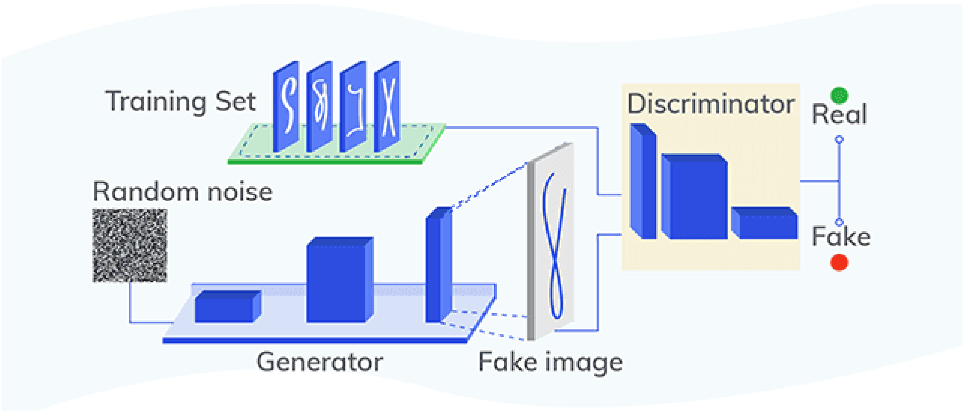


Image Source | xenonstack.com

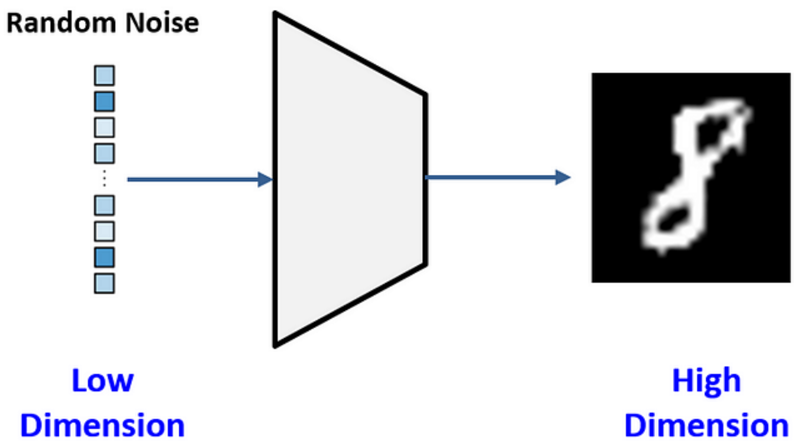
The standard GAN loss function, also known as the min-max loss,
was first described in a 2014 paper by Ian Goodfellow et al., titled "Generative Adversarial Networks":

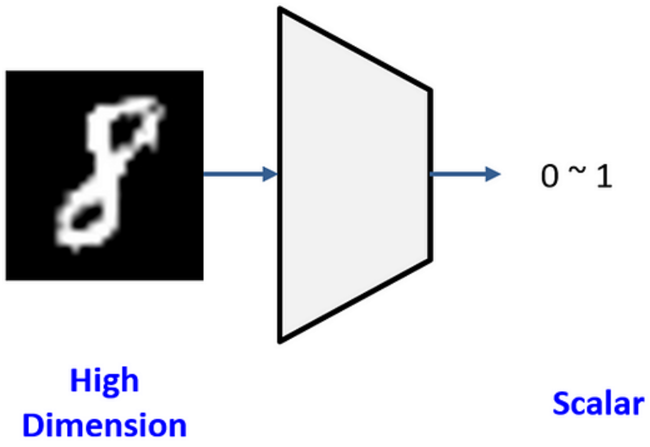
$$E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]$$

Goodfellow, Ian, et al. "Generative adversarial networks." *Communications of the ACM* 63.11 (2020): 139-144.

Generative Adversarial Network (GAN)

Generator	Discriminator





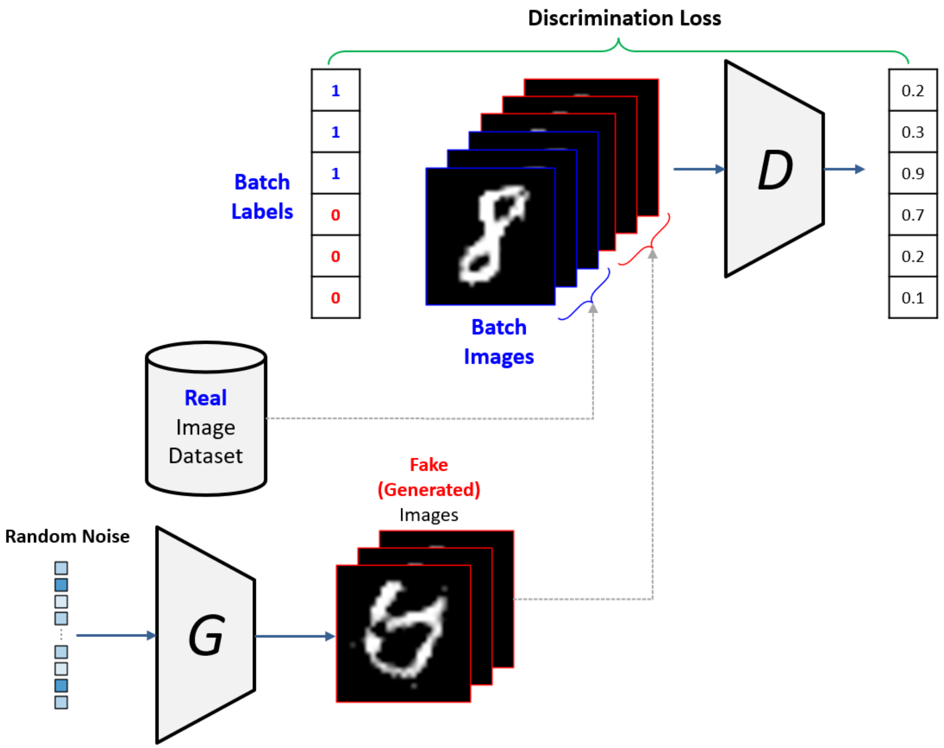
- It works like a 'decoder' in a VAE.
- GOAL: Generate "**REALISTIC**" data.

- It is a typical 'image classification'.
- GOAL: Discriminate between **real and fake** data.

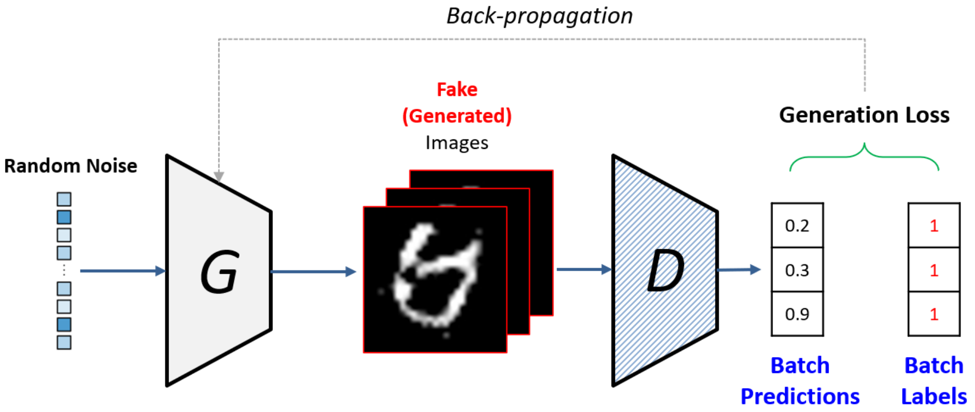
Image Source | https://medium.com/@hugmanskj/gan%EC%97%90-%EB%8C%80%ED%95%9C-%EC%9D%B4%ED%95%B4-a073a5425ef2

Generative Adversarial Network (GAN)

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{z\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(z)))]$$



Training of discriminator



Training of Generator

Image Source | https://medium.com/@hugmanskj/gan%EC%97%90-%EB%8C%80%ED%95%9C-%EC%9D%B4%ED%95%B4-a073a5425ef2

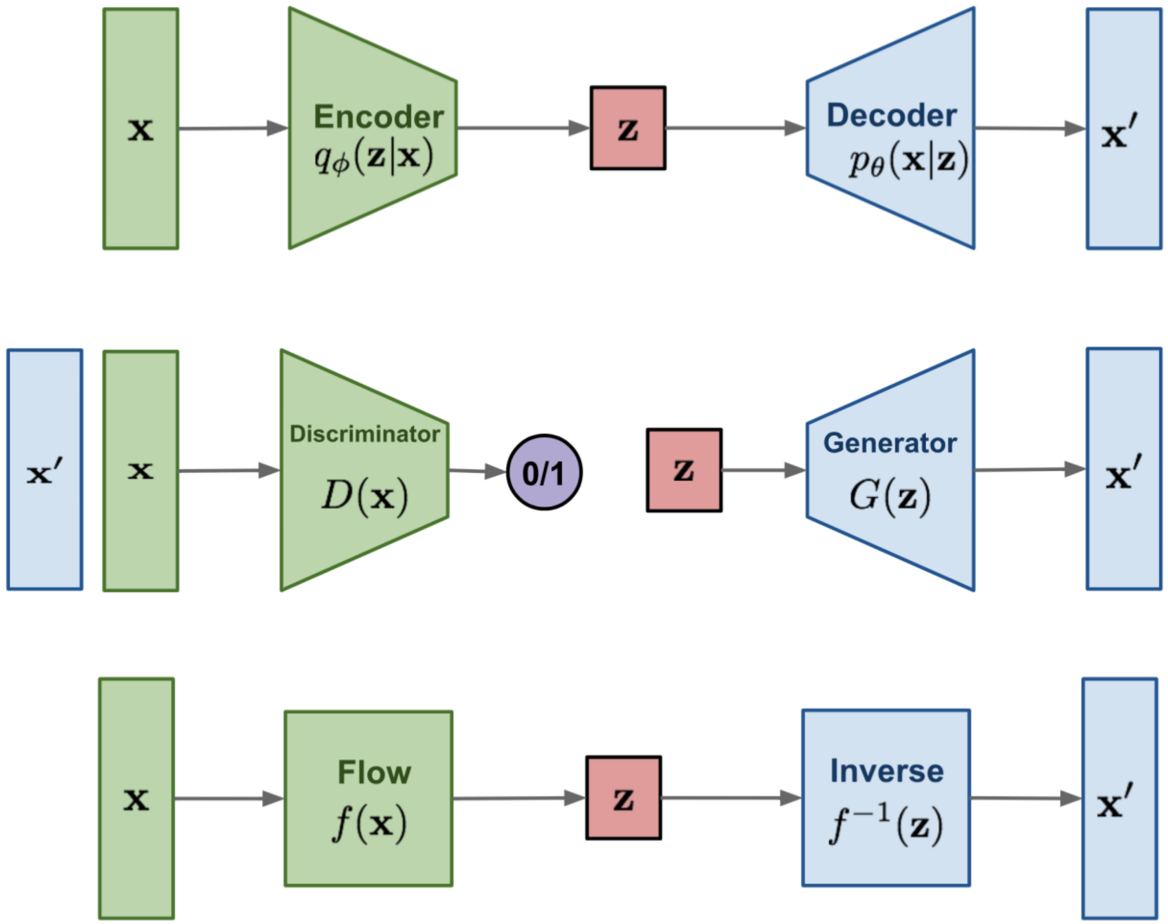
Flow-based Models

Comparison of Different Generative Model Types

VAE: maximize ELBO.

GAN: minimax the classification error loss.

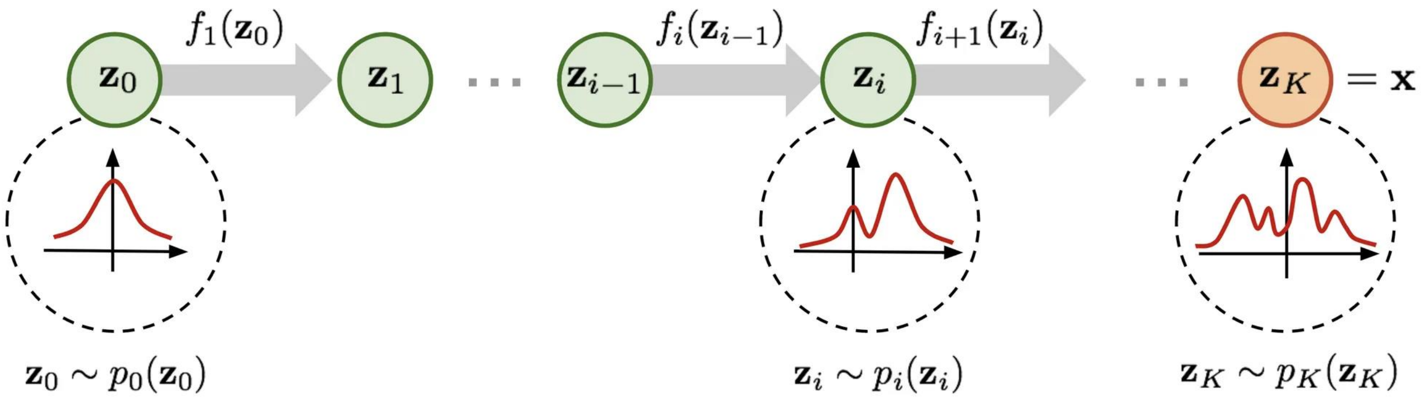
Flow-based generative models: minimize the negative log-likelihood



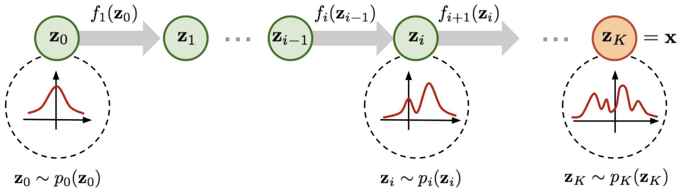
Normalizing Flows

"A normalizing flow transforms a simple distribution into a complex one by applying a sequence of *invertible* transformation functions."

$$\boldsymbol{X} \xrightarrow{\;f^{-1}\;} \boldsymbol{Z} \xrightarrow{\;f\;} \boldsymbol{X}$$



Normalizing Flows



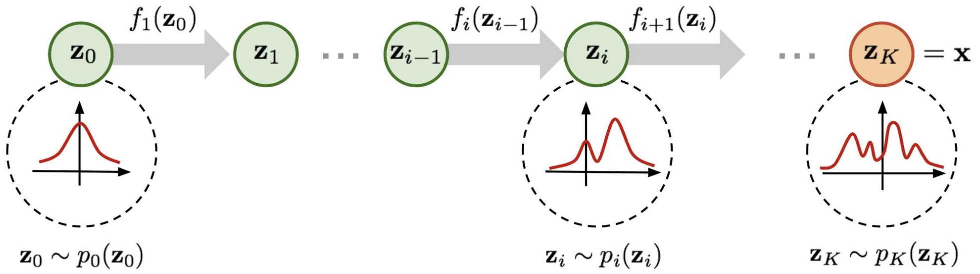
"A normalizing flow transforms a simple distribution into a complex one by applying a sequence of *invertible* transformation functions."

- **Change of variable theorem**
  - Assume that $z \sim \pi(z)$, $x = f(z)$, $z = f^{-1}(x)$ for random variables x and z.
  - $\int p(x)dx = \int \pi(z)dz = 1$ (Definition of probability distribution)
  - $p(x) = \pi(z)\left|\frac{dz}{dx}\right| = \pi(f^{-1}(x))\left|\frac{df^{-1}}{dx}\right| \Rightarrow p(\mathbf{x}) = \pi(\mathbf{z})\left|\det\frac{d\mathbf{z}}{d\mathbf{x}}\right| = \pi\left(f^{-1}(\mathbf{x})\right)\left|\frac{df^{-1}}{d\mathbf{x}}\right|$

- **Inverse function theorem**
  - If $y = f(x)$ and $x = f^{-1}(y)$,

$$\frac{df^{-1}(y)}{dy} = \frac{dx}{dy} = \left(\frac{dy}{dx}\right)^{-1} = \left(\frac{df(x)}{dx}\right)^{-1}$$

Normalizing Flows



By the definition of the normalizing flow model,

$\mathbf{z}_{i-1} \sim p_{i-1}(\mathbf{z}_{i-1})$

$z_i = f_i(z_{i-1})$, thus $z_{i-1} = f_i^{-1}(z_i)$

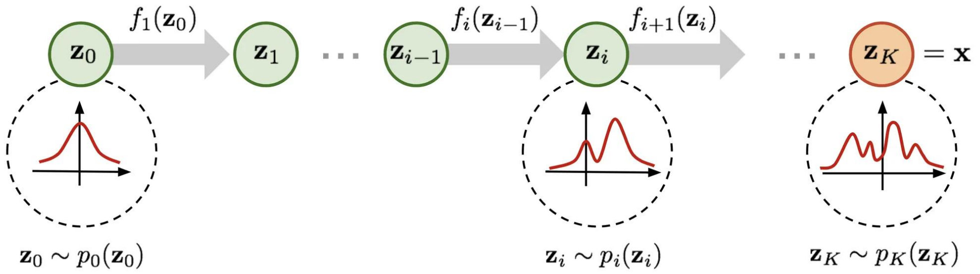
$p_i(z_i) = p_{i-1}\left(f_i^{-1}(z_i)\right)\left|\det \frac{df_i^{-1}}{dz_i}\right|$; according to the change of variable theorem.

$= p_{i-1}(z_{i-1})\left|\det \left(\frac{df_i}{dz_{i-1}}\right)^{-1}\right|$; according to the invers function theorem.

$= p_{i-1}(z_{i-1})\left|\det \frac{df_i}{dz_{i-1}}\right|^{-1}$; according to a property of Jacobians of invertible function.

$\Rightarrow \log p_i(z_i) = \log p_{i-1}(z_{i-1}) - \log\left|\det \frac{df_i}{dz_{i-1}}\right|.$

Normalizing Flows



$$x = z_K = f_K \circ f_{K-1} \circ \cdots \circ f_1(z_0)$$

$$\log p\,(x) = \log \pi_K(z_K)$$

$$= \log \pi_{K-1}(z_{K-1}) - \log |\det \frac{f_K}{dz_{K-1}}|$$

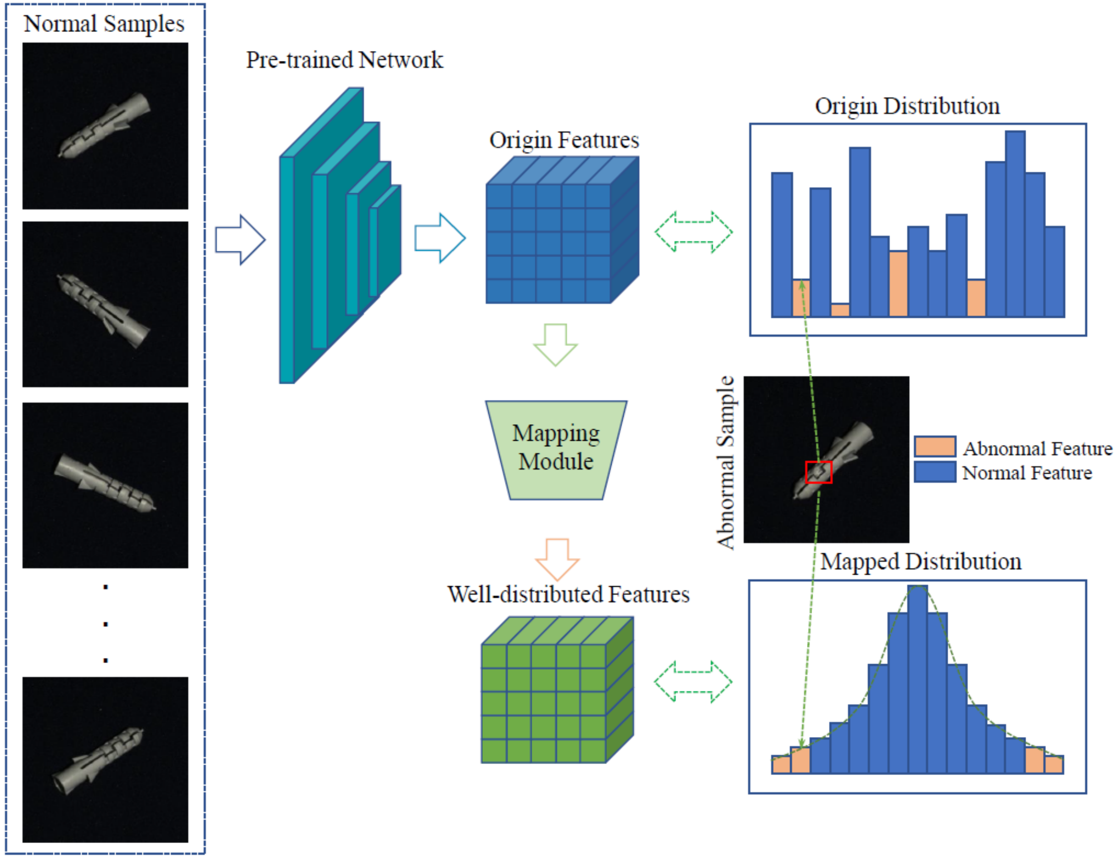
$$= \log \pi_{K-2}(z_{K-2}) - \log \left|\det \frac{f_{K-1}}{dz_{K-2}}\right| - \log |\det \frac{f_K}{dz_{K-1}}|$$

$$= \cdots$$

$$= \log \pi_0(z_0) - \sum_{i=1}^{K} \log |\det \frac{df_i}{dz_{i-1}}|$$

Anomaly Detection Using Normalizing Flows

“Normalizing flows can learn the distribution of normal data to detect anomalies lie out of the distribution.”



Diffusion Models

Diffusion

"A diffusion model gradually converts a well-known distribution like Gaussian noise into a complex data distribution by iteratively removing a bit of noise."

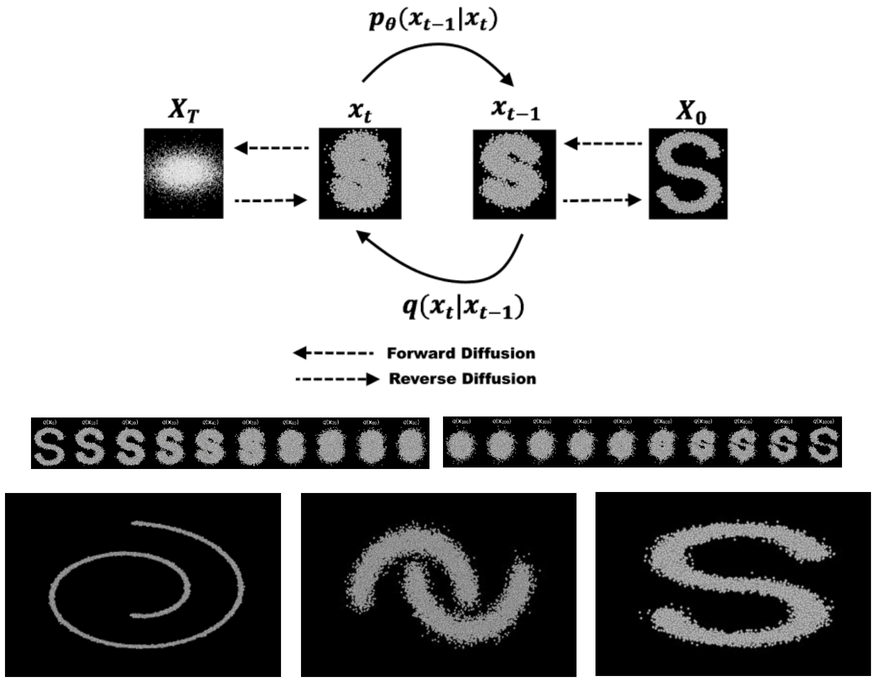


Image Source | Towards Data Science (link)

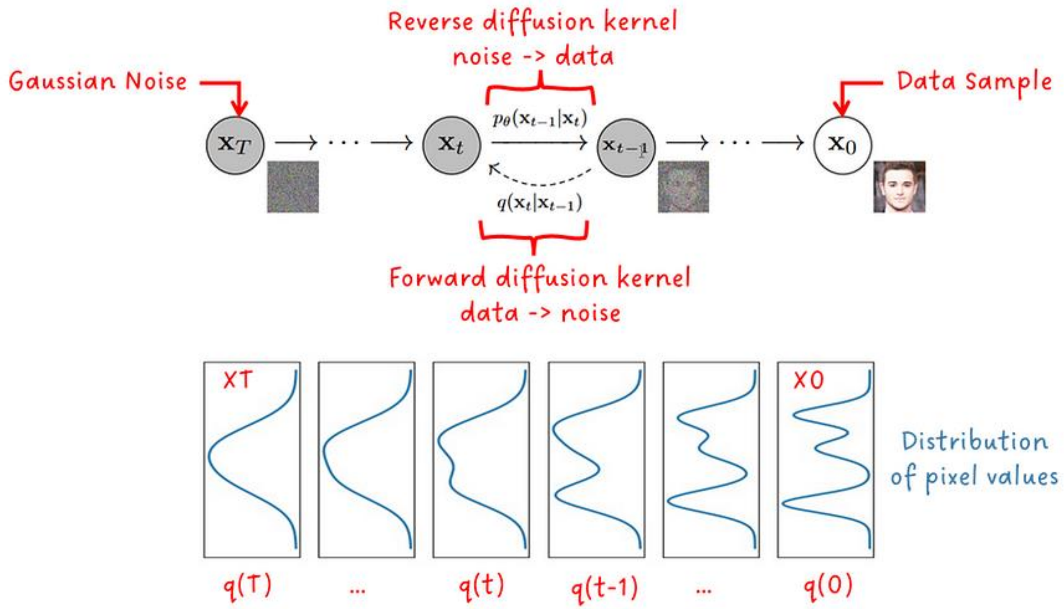


Image Source | Towards Data Science (link)

DDPM: Denoising Diffusion Probabilistic Model

"DDPM is diffusion probabilistic model inspired by considerations from nonequilibrium thermodynamics."

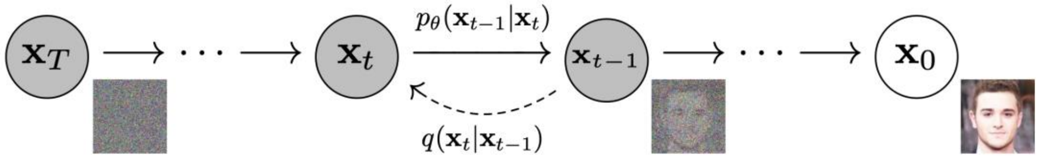


Figure 2: The directed graphical model considered in this work.

- **DDPM at a glance**
  - [Forward diffusion process] We know $q(x_t|x_{t-1})$, but not $q(x_{t-1}|x_t)$.

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

  - [Reverse diffusion process] We approximate $q(x_{t-1}|x_t)$ by modelling $p_\theta(x_{t-1}|x_t)$.
  - Finally, we can model $p_\theta(x_T|x_0)$ through $p_\theta(x_{t-1}|x_t)$ by the Markov chain based on Gaussian transitions.

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)), \qquad p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T, ; 0, \mathbf{I})$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

DDPM: Denoising Diffusion Probabilistic Model

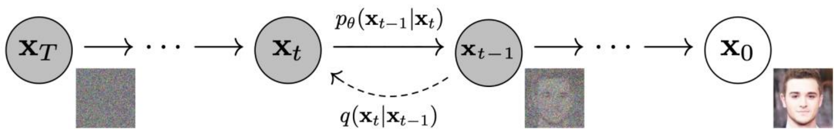


Figure 2: The directed graphical model considered in this work.

- **Objective function**

Minimize the negative log-likelihood:

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L$$

- **Loss function**

$$\mathbb{E}_q\Bigg[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\,\|\,p(\mathbf{x}_T))}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\,\|\,p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Bigg]$$

  - L_T: the difference in distribution between noise of $\mathbf{x}_T$ generated by p and by q given x_0.
  - L_{t-1}: the difference between the reverse (p) and the forward (q) processes.
  - L_0: the likelihood of $\mathbf{x}_0$ given $\mathbf{x}_1$.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

Takeaways

A Probabilistic View of Deep Learning: Discriminative vs. Generative Models

"Discriminative and generative models are two fundamental approaches in deep learning, each with distinct characteristics and applications."

- **Discriminative models**
  - Learn the *decision boundary (e.g.*, classification and regression)
  - Focus on the inference of the output, Y, given the input, X.

  $\Rightarrow$ Model the *conditional probability* of the output given the input, $P(Y|X)$.

- **Generative models**
  - Learn the underlying distribution of data.
  - Focus on the generation of new data instances.

  $\Rightarrow$ Model the *joint probability* $P(X,Y)$. Depending on the problem, $P(X|Y)$ or $P(X)$ can be of interest.

Examples of Deep Generative Models

- Variational Autoencoders (VAEs)
- Generative Adversarial Networks (GANs)
- Autoregressive models (e.g., Transformers like GPT)
- Flow-based models
- Diffusion models
- …

Thank you! 😀