Generative Model

2020. 11. 9

방준일

Korea AI Center for Drug Discovery and Development

Contents

1 AI Based Drug Development

2 개요

3 생성모델 기초

4 VAE

5 GAN

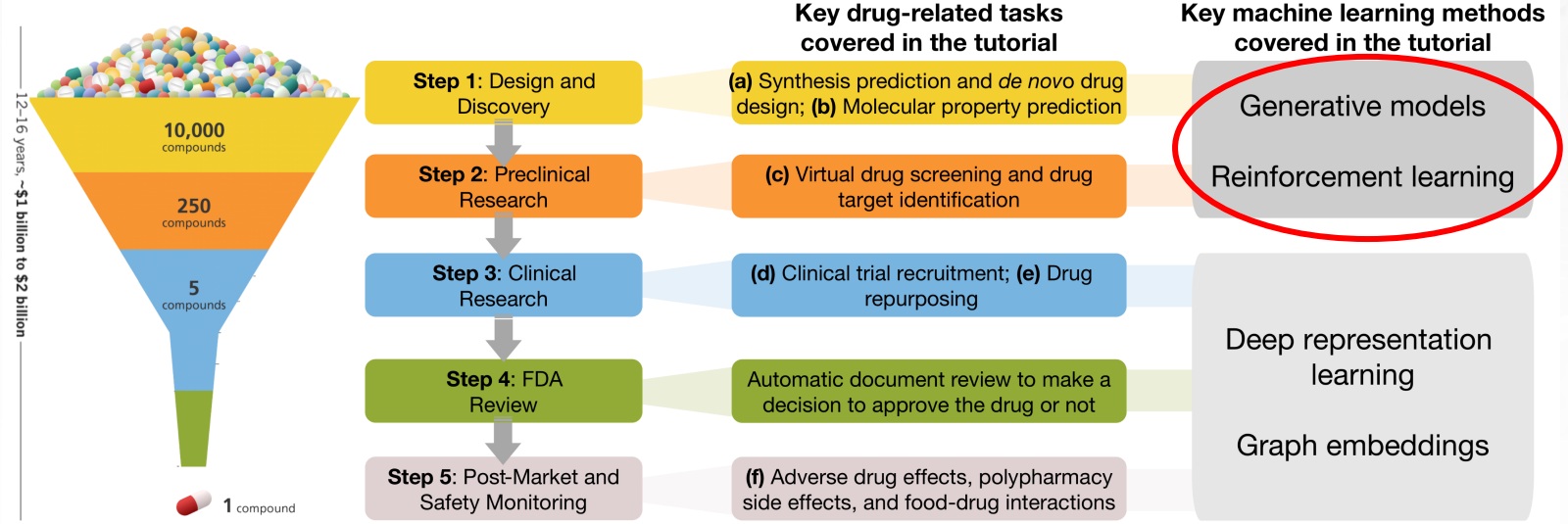
6 AAE

7 MOSES 실습

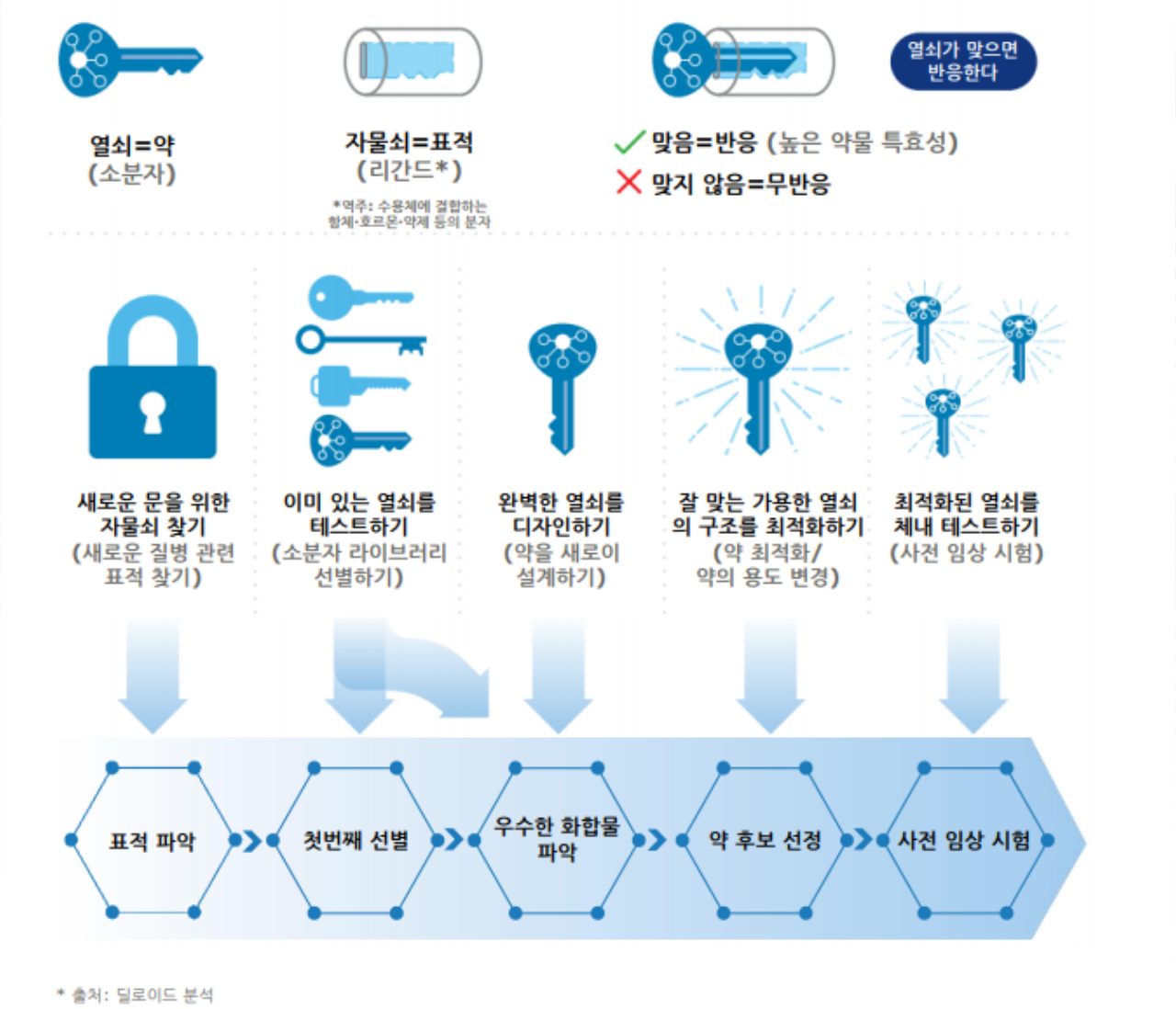
1. AI Based Drug Development

AI Based Drug Development

- a : 합성 예측 및 새로운 **약물 설계**, 복잡한 분자 합성을 지원하기 위해 화학적으로 올바른 구조를 생성
- b : 분자 **특성 예측**, 분자 데이터에서 효능, 생물 활성 및 독성과 같은 특성을 예측하여 분자의 치료 효과를 식별
- c : 가상 약물 스크리닝 및 **약물 표적 식별**, 약물이 표적 단백질에 결합하고 다운 스트림 활동에 영향을 줌으로써 약물이 인체에 미치는 영향을 예측



AI Based Drug Development



AI Based Drug Development

In Silico Drug Design & Discovery

functional genomics, structural genomics, proteomics and bioinformatics	in silico structure-based design (novel or known compound) HTS (in vitro & in vivo)	toxicity assays (in vitro & in vivo)

출처 : British Columbia

Drug Development Using AI

- Design & Discovery
  - *de novo* Drug Design
  - Molecular property prediction

- Preclinical Research
  - Virtual drug screening, drug target identification

AutoEncoder(VAE, AAE 등), GAN – **Generative**

DTI(QSAR, GraphDTI, DeepDTA …등) – **Discriminative**

2. 개요

Classification

- 분류 문제에서 사용되는 모델은 크게 2가지

- **Discriminative** Model – **판별** 모델

- **Generative** Model – **생성** 모델

Discriminative Classifiers

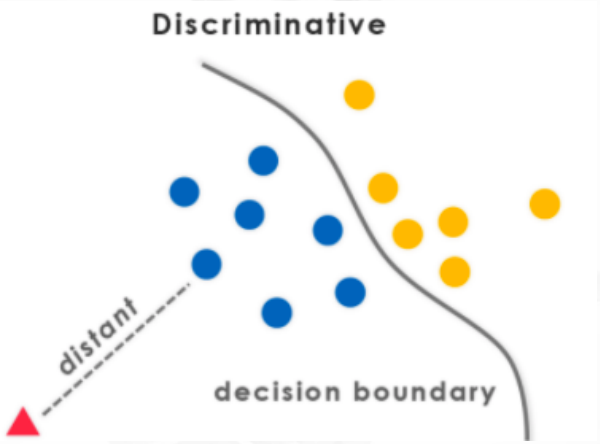
- Logistic regression
- Scalar Vector Machine
- Traditional neural networks
- Nearest neighbour
- Conditional Random Fields (CRF)s

Generative classifiers

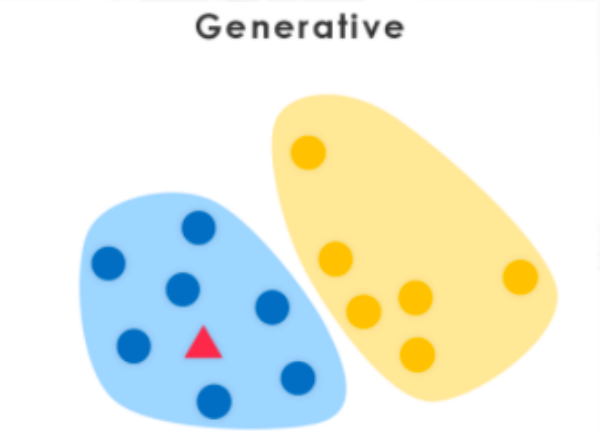
- Naïve Bayes
- Bayesian networks
- Markov random fields
- Hidden Markov Models (HMM)

Classification

- Class의 **차이점**에 주목하여 확률적 계산을 통해 판별
  - 데이터가 충분할수록 좋은 성능
  - 구분만이 목적, 본질에 대한 이해가 어려움



- Class의 **분포**(거리, 밀도 등)에 주목
  - 데이터가 충분하지 않아도 성능을 보임
  - 본질 파악 가능
  - **유사 데이터 생성 가능**



Generative Model

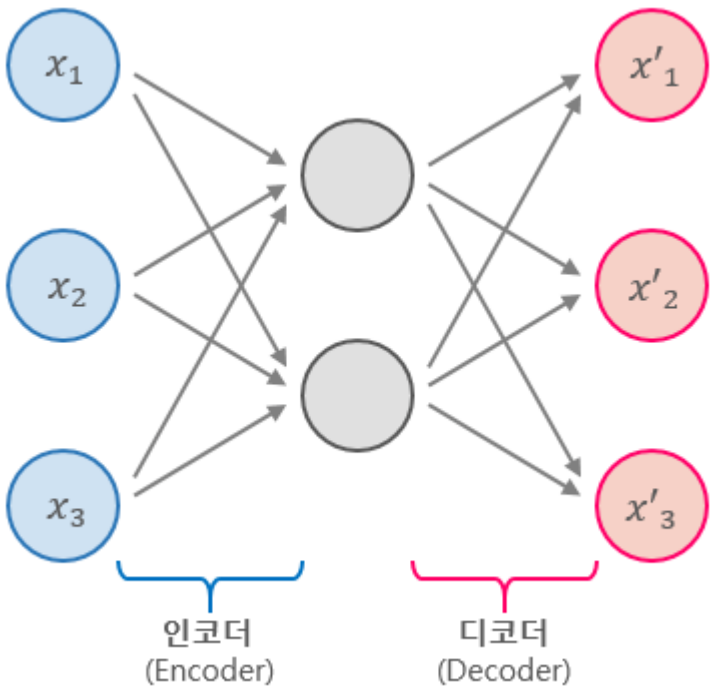
- Loss 값의 계산과 Back Propagation(역전파) 학습이 어려움
- 성능을 명확히 수치화하기 어려워 사용이 많지 않았음
- 여러 방식들의 활용 보고로 인하여 가능성이 조명을 받게 됨

3. 생성모델 기초

Auto Encoder

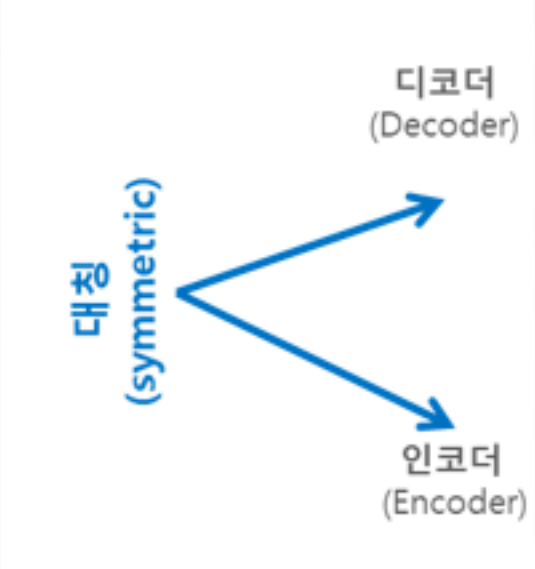
- 자기 자신을 **재생산**하는 모델

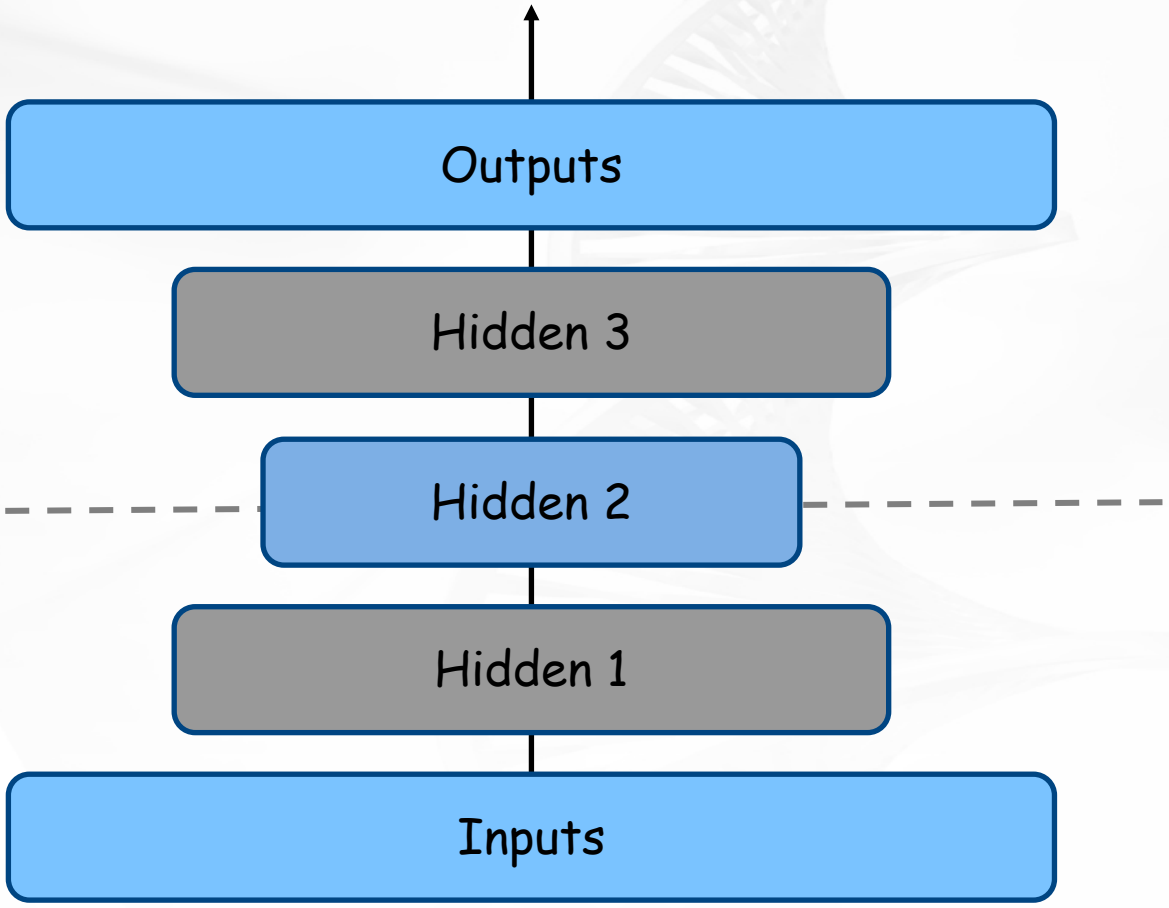
- Recognition network – encoder
- Generative network – decoder



Auto Encoder

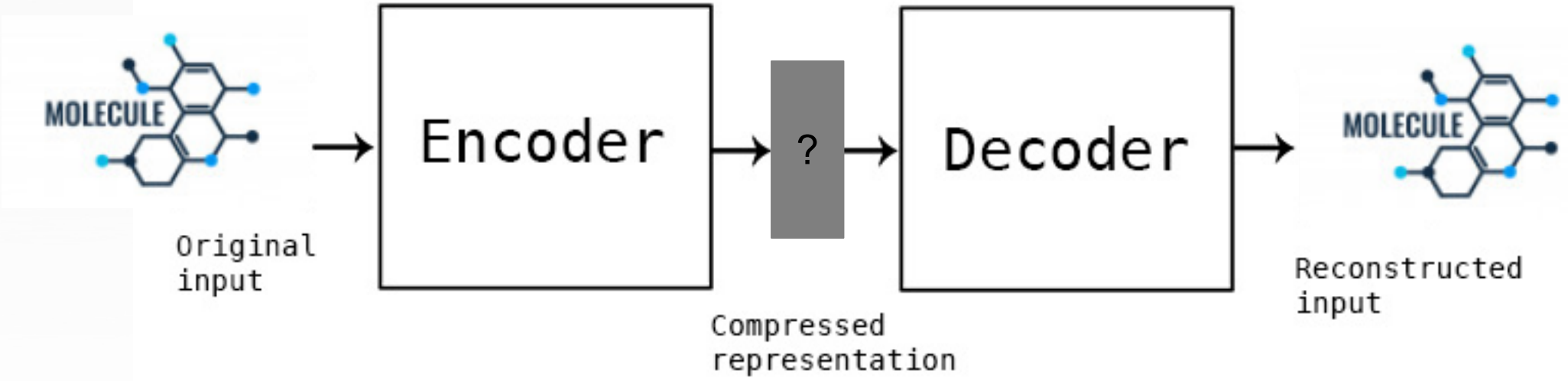
- Stacked Auto Encoder
- 여러 개의 히든 레이어를 가짐
- 보다 복잡한 학습 가능





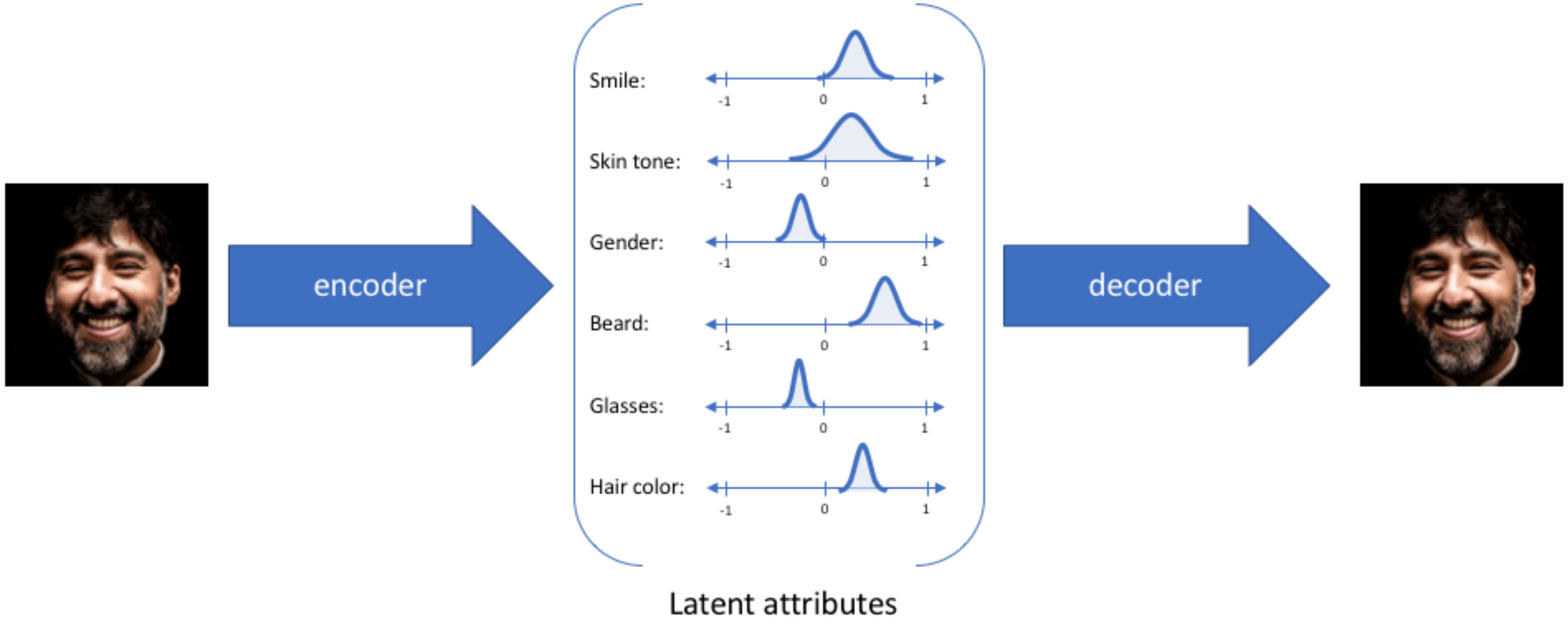
Auto Encoder

- 압축된 Latent Space는 input 데이터의 특징을 가지고 있음
- 해당 특징을 보존하여 output 생성



Auto Encoder

- 그림 예시



참조 : https://www.jeremyjordan.me/variational-autoencoders/

Auto Encoder

- Data- Specific :
  훈련된 데이터와 비슷한 데이터로만 압축 및 생성 가능

- 자동적 학습 :
  특정 입력값에 대하여 잘 작동하는 모델로 훈련

- 손실 값 존재 :
  압축 해제, 즉, 디코딩되어 생성된 값은 원본 보다 손실된 결과

Auto Encoder

- Auto Encoder는 타 알고리즘들의 활용 빈도수 보다 낮았으며, 실제 응용이 드물었음

- 다만, 비지도 학습의 문제를 풀어낼 **KEY**로 생각되었음

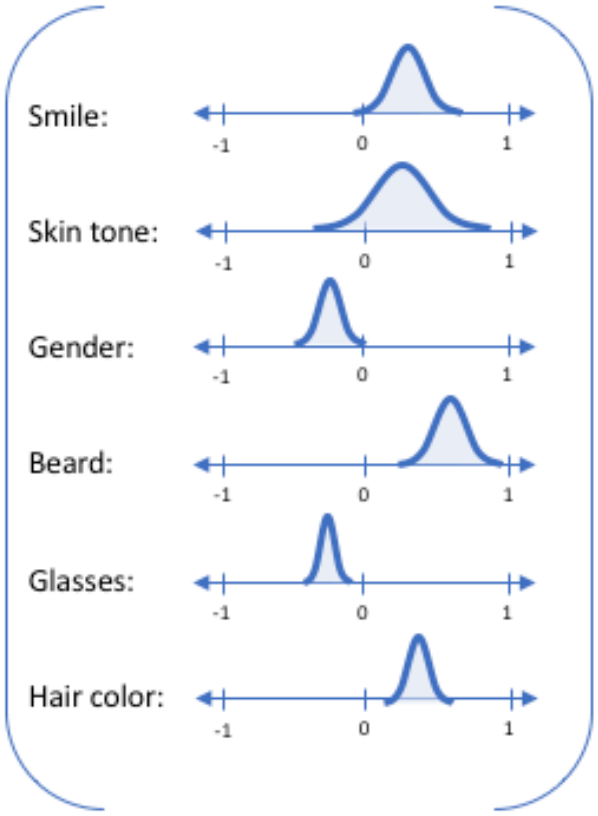
- Deep Auto Encoder, Convolutional Auto Encoder, Sequence-to-sequence Auto Encoder(used RNN) 등의 방향으로 발전

4. VAE

Variational Auto Encoder

- 제약 조건이 추가된 Auto Encoder
- 입력 데이터에 대한 Latent Variable Model을 학습
- 임의의 함수 학습 대신, 데이터를 모델링하는 확률 분포의 매개변수를 학습

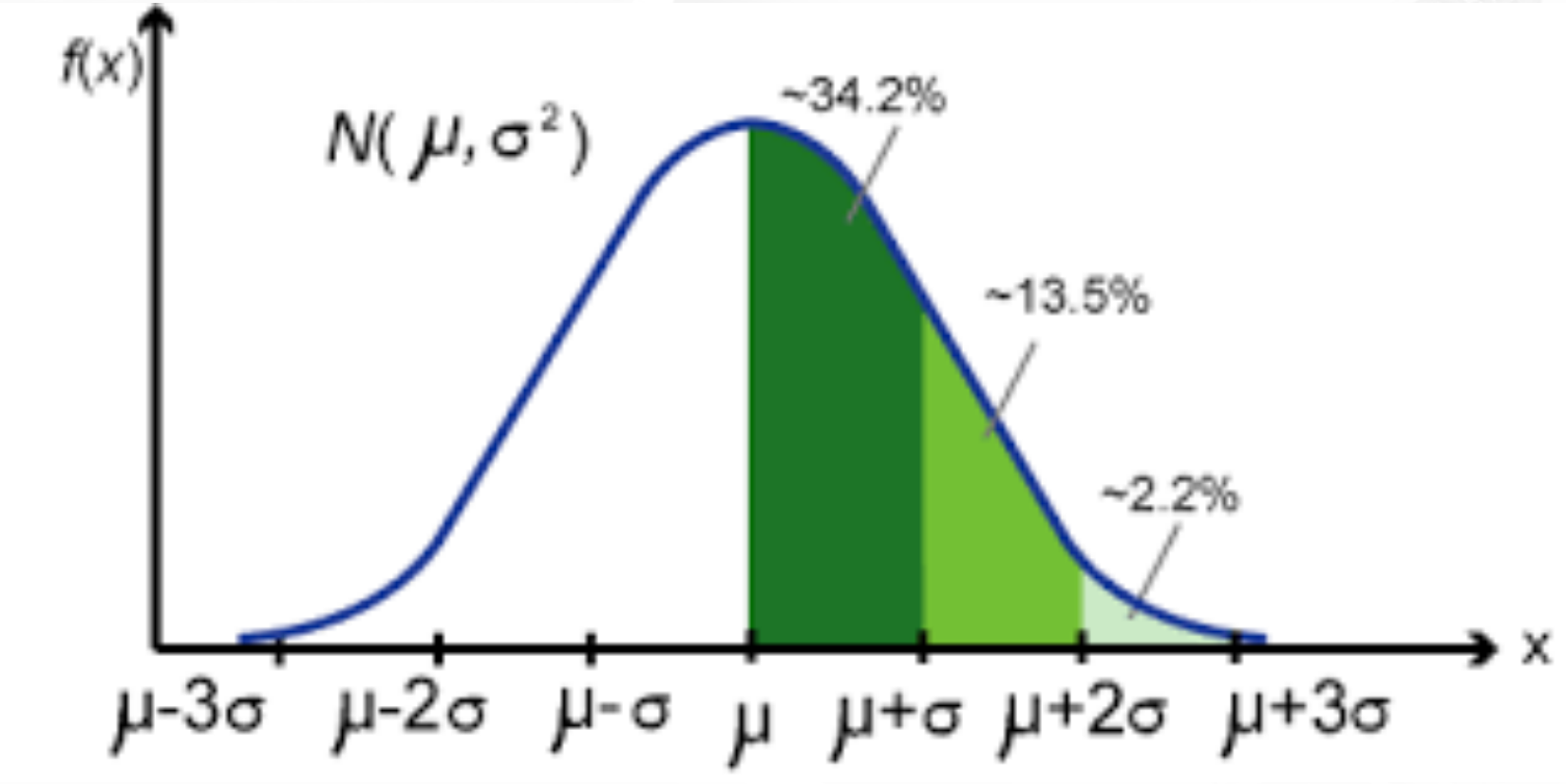
Variational Auto Encoder



Latent attributes

Variational Auto Encoder

$$z=\mu(x)+\sigma(x)\times\epsilon,\epsilon\sim N(0,1)$$

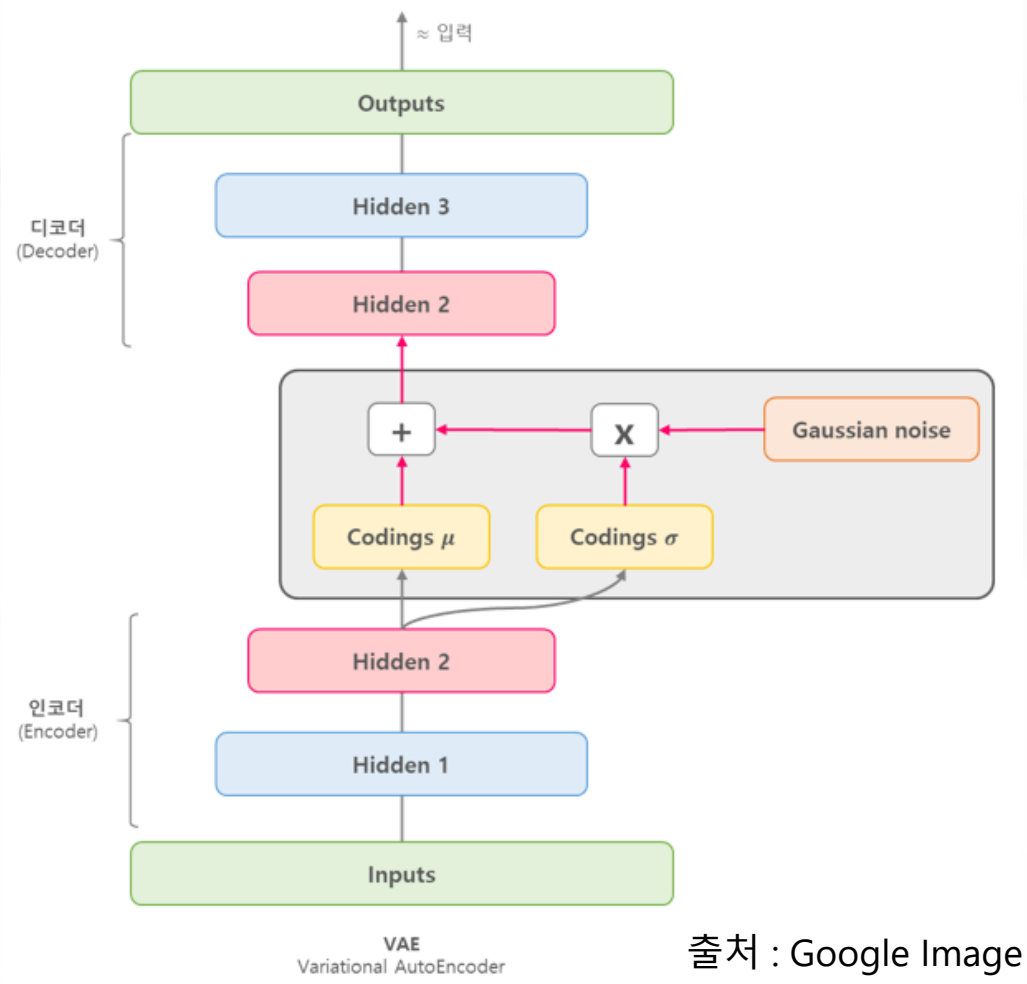


Variational Auto Encoder

- 확률적 오토인코더 :
  학습이 끝난 후에도 출력이 부분적으로
  우연에 의해 결정 – 가우시안 노이즈 영향

- 생성 오토인코더 :
  데이터셋에서 샘플링 된 것과 같은
  새로운 샘플을 생성 가능 – 가우시안 분포 영향

- 2014년 D.Kingma와 M.Welling
  Auto-Encoding Variational Bayes



출처 : Google Image

Variational Auto Encoder

- 보통의 Auto Encoder와 다르게, 주어진 입력에 추가적 연산

- 평균이 μ이고 표준편차가 σ 인 가우시안 분포에서 랜덤하게 샘플링 후, 디코더가 원본 입력으로 재구성
  -> 가우시안 분포(정규분포)에서 샘플링 된 것처럼 보이는 연산

  $z=\mu(x)+\sigma(x)\times\epsilon, \epsilon\sim N(0,1)$

- 학습하는 동안 손실함수가 가우시안 분포와 유사한 형태의 Latent Sapce(잠재변수 공간)을 따르도록 함

Loss Fucntion - Regularization

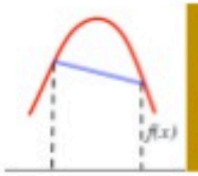
- KL Divergence - 쿨백-라이블러 발산(Kullback-Leibler divergence, KLD)

- 두 확률분포의 차이를 계산하는 데 사용하는 함수

$$D_{KL}(P||Q) = E_{X\sim P}\left[\log\frac{P(x)}{Q(x)}\right] = E_{X\sim P}\left[-\log\frac{Q(x)}{P(x)}\right]$$

Loss Fucntion

$$\log(p(x)) = \log\left(\int p(x|z)p(z)dz\right) \geq \int \log(p(x|z))p(z)dz \quad \leftarrow$$



[Jensen's Inequality]

For concave functions f(.)
f(E[x])≥E[f(x)]

f(.) = log(.) is concave

$$\log(p(x)) = \log\left(\int p(x|z)\frac{p(z)}{q_\phi(z|x)}q_\phi(z|x)dz\right) \geq \int \log\left(p(x|z)\frac{p(z)}{q_\phi(z|x)}\right)q_\phi(z|x)dz \leftarrow \text{Variational inference}$$

$$\log(p(x)) \geq \int \log(p(x|z))q_\phi(z|x)dz - \int \log\left(\frac{q_\phi(z|x)}{p(z)}\right)q_\phi(z|x)dz$$

$$= \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))] - KL(q_\phi(z|x)||p(z))}_{ELBO(\phi)}$$

Variational lower bound
Evidence lower bound (ELBO)

ELBO를 최대화하는 ϕ^*값을 찾으면 $\log(p(x)) = \mathbb{E}_{q_{\phi^*}(z|x)}[\log(p(x|z))] - KL(q_{\phi^*}(z|x)||p(z))$이다.

https://www.slideshare.net/NaverEngineering/ss-96581209

Loss Fucntion

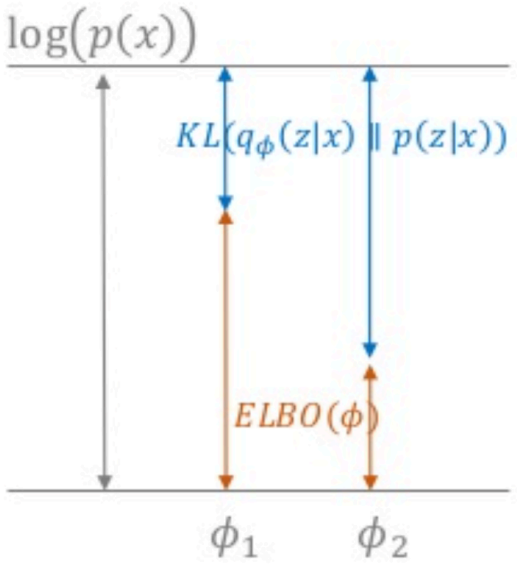
$$\log(p(x)) = \int \log(p(x)) q_\phi(z|x) dz \quad \leftarrow \int q_\phi(z|x)dz = 1$$

$$= \int \log\left(\frac{p(x,z)}{p(z|x)}\right) q_\phi(z|x) dz \quad \leftarrow p(x) = \frac{p(x,z)}{p(z|x)}$$

$$= \int \log\left(\frac{p(x,z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x) dz$$

$$= \underbrace{\int \log\left(\frac{p(x,z)}{q_\phi(z|x)}\right) q_\phi(z|x) dz}_{ELBO(\phi)} + \underbrace{\int \log\left(\frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x) dz}_{KL\left(q_\phi(z|x) \,\|\, p(z|x)\right)}$$

두 확률분포 간의 거리≥0



KL을 최소화하는 $q_\phi(z|x)$의 ϕ값을 찾으면 되는데 $p(z|x)$를 모르기 때문에,
KL최소화 대신에 ELBO를 최대화하는ϕ값을 찾는다.

https://www.slideshare.net/NaverEngineering/ss-96581209

Loss Fucntion

$$\log(p(x)) = ELBO(\phi) + KL(q_\phi(z|x)|p(z|x))$$

$$q_{\phi^*}(z|x) = \underset{\phi}{\operatorname{argmax}}\, ELBO(\phi)$$

$$\begin{aligned}
ELBO(\phi) &= \int \log\left(\frac{p(x,z)}{q_\phi(z|x)}\right) q_\phi(z|x)dz \\
&= \int \log\left(\frac{p(x|z)p(z)}{q_\phi(z|x)}\right) q_\phi(z|x)dz \\
&= \int \log(p(x|z))q_\phi(z|x)dz - \int \log\left(\frac{q_\phi(z|x)}{p(z)}\right) q_\phi(z|x)dz \\
&= \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))] - KL(q_\phi(z|x)||p(z))
\end{aligned}$$

https://www.slideshare.net/NaverEngineering/ss-96581209

Loss Fucntion

- Reconstruction Loss => cross entropy
  - 복원 값의 오차를 계산하여 줄이는 과정

- Regularization Loss => D_kl
  - 두 확률 분포의 차이를 줄이는 과정

원 데이터에 대한 likelihood 선택

Variational inference를 위한 approximation class 중 선택

다루기 쉬운 확률 분포 중 선택

$$L_i(\phi, \theta, x_i) = \underbrace{-\mathbb{E}_{q_\phi(z|x_i)}\left[\log\left(p_\theta(x_i|z)\right)\right]}_{\textbf{Reconstruction Error}} + \underbrace{KL\left(q_\phi(z|x_i)||p(z)\right)}_{\textbf{Regularization}}$$

Reconstruction Error

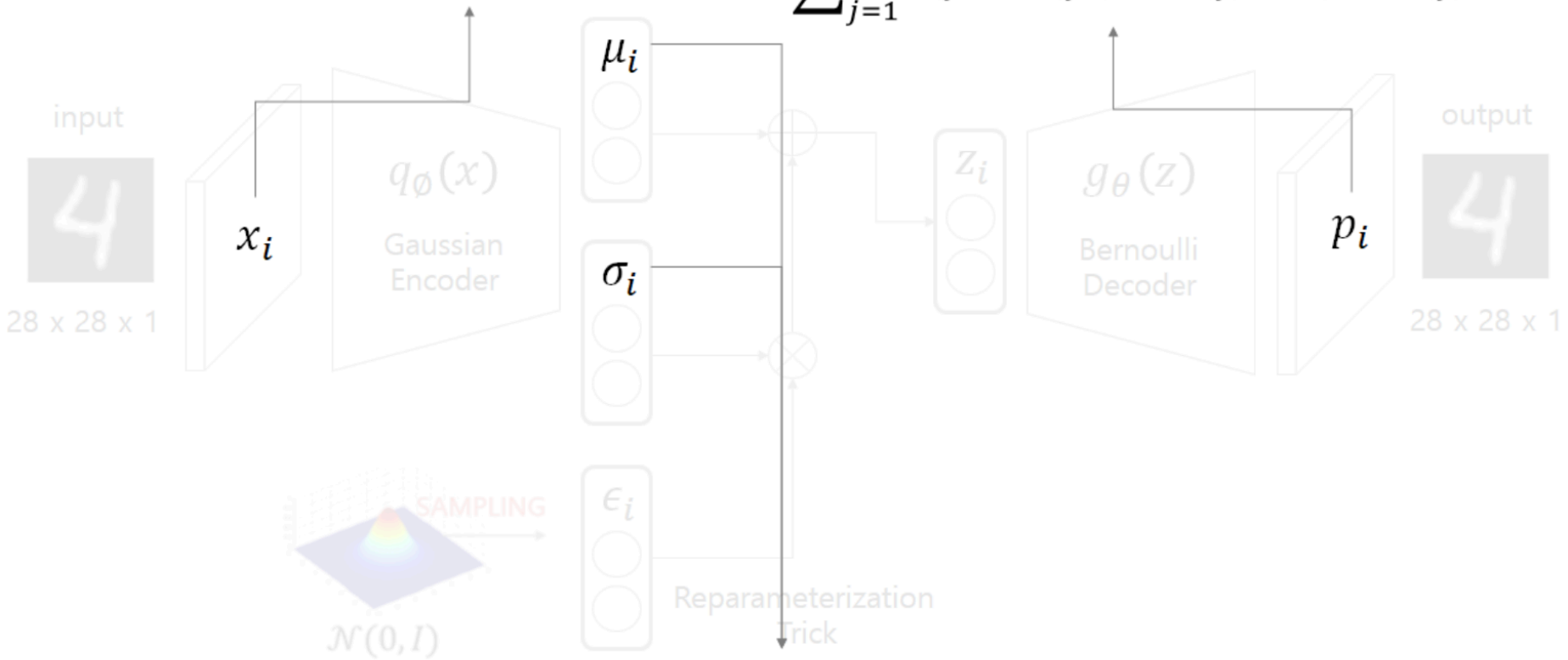
- 현재 샘플된 z에 대한 negative log likelihood
- x_i에 대한 복원 오차 (AutoEncoder 관점)

Regularization

- 현재 샘플된 z에 대한 대한 추가 조건
- 샘플링되는 z들에 대한 통제성을 prior를 통해 부여, Variational distribution q(z|x)가 p(z)와 유사해야 한다는 조건을 부여

Loss Fucntion

Reconstruction Error: $-\sum_{j=1}^{D} x_{i,j} \log p_{i,j} \, (1 - x_{i,j}) \log(1 - P_{i,j})$

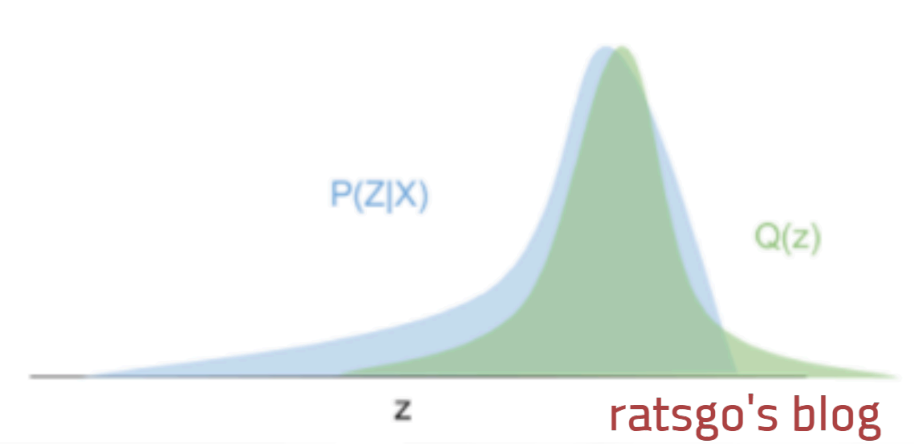


Regularization : $\frac{1}{2} \Sigma_{j=1}^{\bar{J}} (\mu_{i,j}^2 + \sigma_{i,j}^2 - \ln(\sigma_{i,j}^2) - 1)$

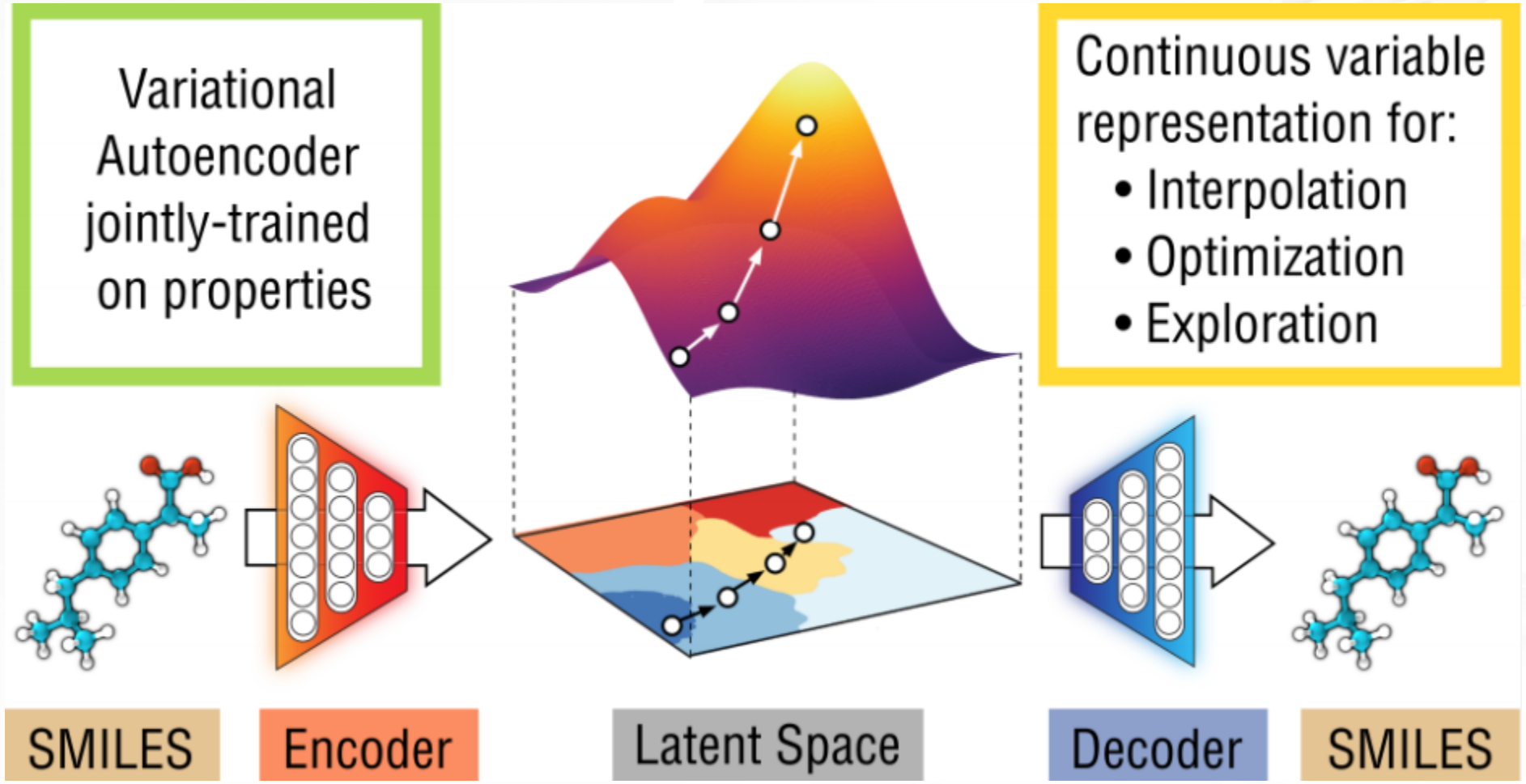
그림 출처 : taeu.github

Loss Fucntion - Regularization

- 두 확률 분포의 차이를 줄이는 것으로 학습



Variational Auto Encoder



참조 : Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules

Variational Auto Encoder

- 학습이 쉬움

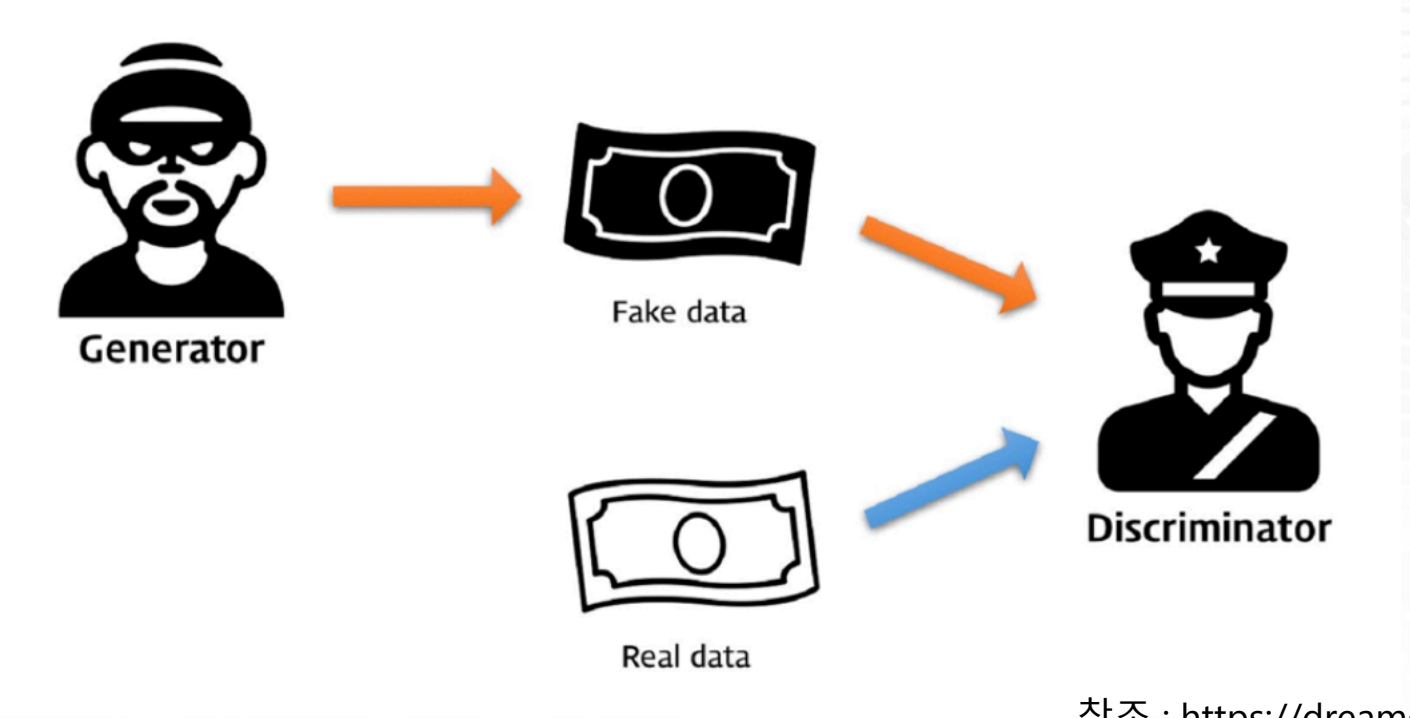
- 복잡한 데이터 생성 모델을 설계하고 대규모 set에 적응 할 수 있게 해줌

- VAE는 data 분포가 잘 학습되기만 하면 sampling (=data generation)이 자연스럽게 따라옴

5. GAN

Generative Adversarial Networks

- Generative Adversarial Networks (GAN) 의 특성 :
  - Generator Model의 목적 자체가 임의의 정해진 분포를 학습하는 것이 아님
  - 진짜 같은 Sample을 Generate하는 것이 목적



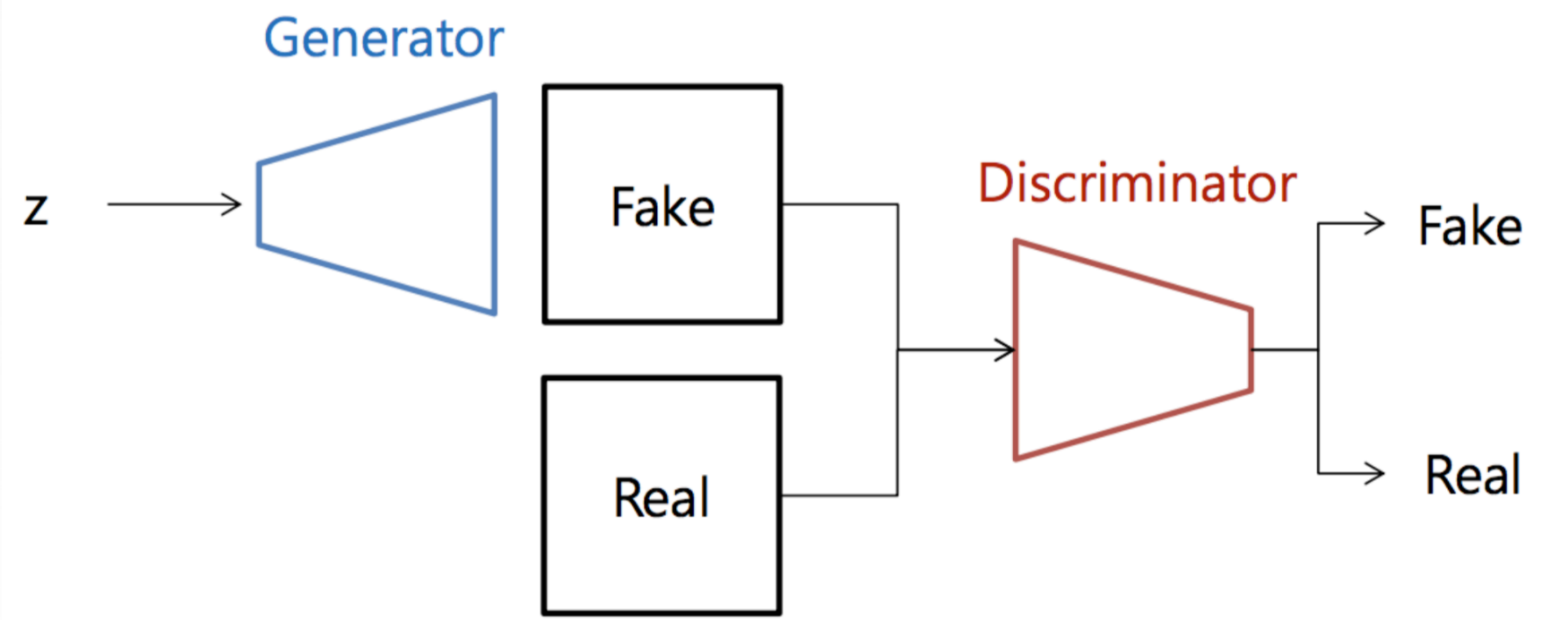
참조 : https://dreamgonfly.github.io/blog/gan-explained/

Generative Advesarial Networks

- 두 네트워크를 적대적으로 학습
- 비지도 학습 기반 생성 모델 - unsupervised generative model
- 생성자 ***G***는 *Zero-Mean Gaussian*으로 생성된 ***z*** 를 기반
- GAN의 궁극적인 목적은 <u>실제 데이터의 분포</u>에 가까운 데이터를 생성

Generative Advesarial Networks

- 생성자(generator, ***G***)
- 구분자(discirimiator, ***D***)



참조 : ratsgo's blog

Loss Function

$$\min_{G} \max_{D} V(D, G) = E_{x \sim p_{data}(x)} \left[\log D(x) \right] + E_{z \sim p_z(z)} \left[\log \{ 1 - D(G(z)) \} \right]$$

- Discirimiator – D
  - 실제 데이터 x 를 입력하면 D(x) 증가 – 좌항 증가
  - 가짜 데이터 G(z) 를 입력하면 D(G(z)) 증가 – 우항 감소

- Generator – G
  - 가짜 데이터 G(z)가 실제 데이터의 확률처럼 높게 나오도록 D(G(z)) 감소 – 우항 증가

- 두 네트워크를 동시에 학습시키지 않고 따로따로 업데이트(D와 G 고정)

Loss Function - D

$$\max_{D} V(D) = E_{x \sim p_{data}(x)} \left[\log D(x)\right] + E_{z \sim p_z(z)} \left[\log \{1 - D(z)\}\right]$$

$$= \frac{1}{m} \sum_{i=1}^{m} \log D\left(x^i\right) + \frac{1}{m} \sum_{i=1}^{m} \log \left\{1 - D\left(G\left(z^i\right)\right)\right\}$$

Loss Function - G

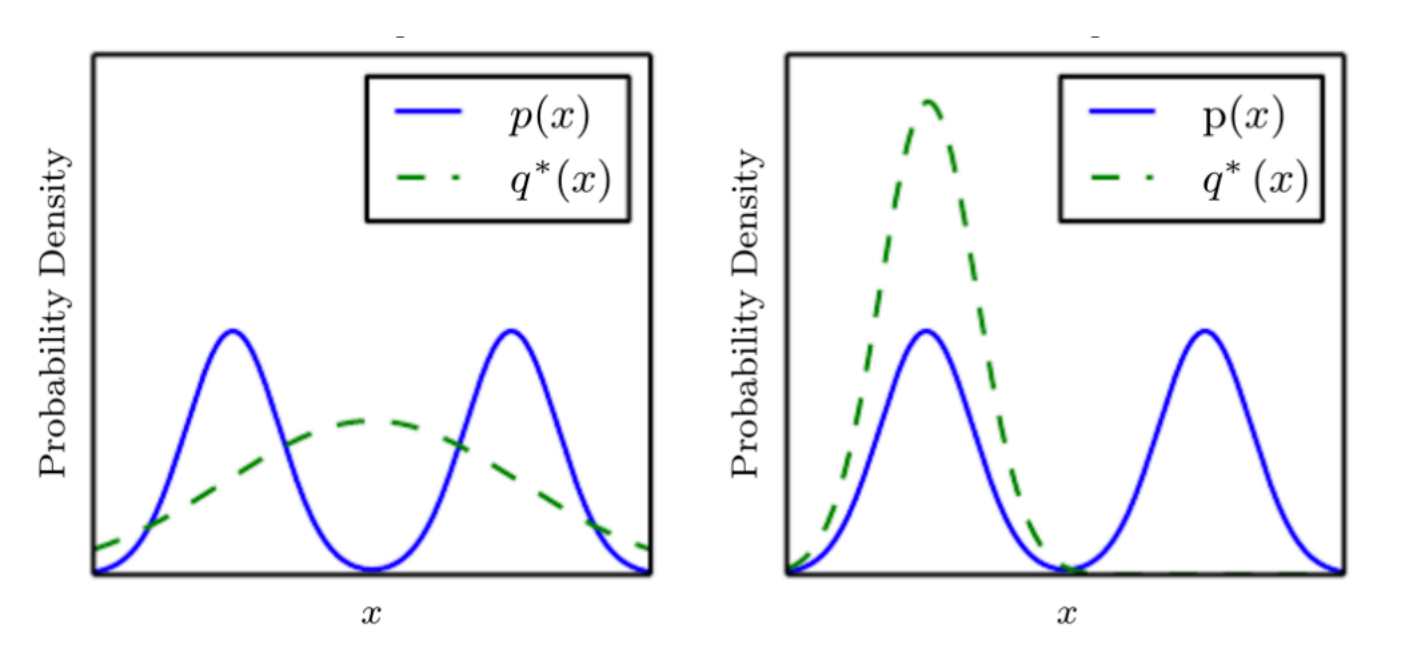
$$\min_{G} V(G) = E_{z \sim p_z(z)} \left[\log \{ 1 - D(G(z)) \} \right]$$

$$= \frac{1}{m} \sum_{j=1}^{m} \log \left\{ 1 - D\left(G\left(z^j\right)\right) \right\}$$

$$\Rightarrow -E_{z \sim p_z(z)} \left[\log D(G(z)) \right]$$

Generative Advesarial Networks

- **mode collapsing**
  - 실제 데이터의 분포를 모두 커버하지 못하고 다양성을 잃어버리는 현상
- Loss값의 감소 만을 위한 학습으로 인해 G가 전체 데이터 분포를 찾지 못함 (하나의 *mode*에만 강하게 몰림)



참조 : ratsgo's blog

Generative Advesarial Networks

- 정량적인 평가가 어려움

 - **정성평가** – 사람이 직접 평가

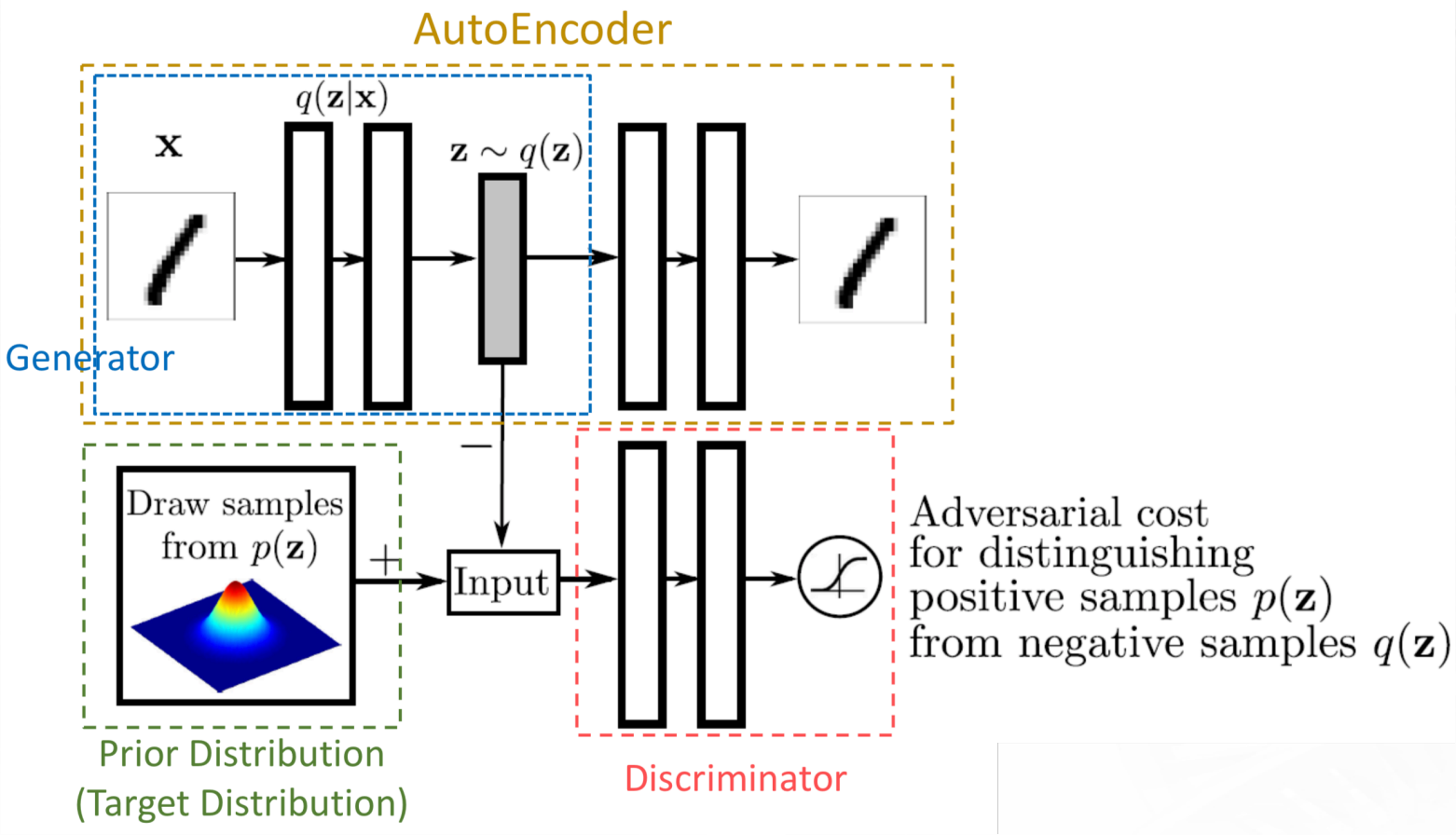
 - **학습된 분류기를 이용** – 지도학습한 다른 판별모델로 평가

- 학습이 어려움

6. AAE

Adversarial Auto Encoder

- VAE에 GAN을 융합



참조 : ratsgo's blog

Adversarial Auto Encoder

- VAE는 Sampling이 가능하고 Numerical하게 계산이 되어야 함
- 따라서 사전 확률 분포를 정규분포로 가정하고 KL-divergence를 사용
- AAE는 두 분포를 유사하게 만들기 위하여 KL-divergence 대신, GAN이 이 역할을 수행
- 특정 확률 분포를 전제로 할 필요가 없는 GAN의 장점을 이용
  (실제 데이터가 어떤 확률 분포 형태이던 따르도록 설계됨)

Loss Function

- Autoencoder **- Reconstruction Loss**
- Generator Loss, Discriminator Loss **- Regularization Loss**

Training Step 1 : Update AE
update ϕ, θ according to reconstruction error

$$L_i(\phi, \theta, x_i) = -\mathbb{E}_{q_\phi(z|x_i)}\left[\log\left(p_\theta(x_i|z)\right)\right]$$

Training Step 2 : Update Discriminator
update λ according to loss for discriminator

$$-V_i(\phi, \lambda, x_i, z_i) = -\log d_\lambda(z_i) - \log\left(1 - d_\lambda\left(q_\phi(x_i)\right)\right)$$

Training Step 3 : Update Generator
update ϕ according to loss for discriminator

$$-V_i(\phi, \lambda, x_i, z_i) = -\log\left(d_\lambda\left(q_\phi(x_i)\right)\right)$$

참조 : ratsgo's blog

Adversarial Auto Encoder

- Regularization 역할의 두 Loss가 없다면 기본 AutoEncoder 모델과 동일

- 샘플링을 담당하는 재건 부분이 없는 모델은,
  단순히 Input을 재현하는 기본적인 autoencoder의 형태를 취하기 때문

7. MOSES 실습

Property

- log P : 물-옥탄올 분배 계수(water-octanal partition coefficient)
  - 농도 비율에 log를 취한 값을 보통 사용
  - 이온화 되지 않은 화합물의 농도 비율을 logP
  - 유기 용매의 극성이 높을 수록 P 값이 낮아지며,
  - 따라서 독성이 더 큼을 의미
  - 해당 코드에서는 객체 사이의 유사성을 판별하기 위하여 사용

Property

- QED : 약물 유사성의 정량적 추정(Quantitative Estimation of Drug-likeness)

~~- 논문 "Quantifying the chemical beauty of drugs"에 기반~~

- 0~1 사이의 값

- 분자량, logP, 토폴로지 극성 표면적, 수소 결합 공여체 및 수용체 수, 방향족 고리 및 회전 가능한 결합 수, 원치 않는 화학적 기능의 존재를 포함한 분자 특성의 기본 분포를 반영
  - molecular weight, logP, topological polar surface area, number of hydrogen bond donors and acceptors, the numbe r of aromatic rings and rotatable bonds, and the presence of unwanted chemical functionalities
- 화합물의 품질을 정량화하기 위해 선호도 개념을 적용
- 표적 기능과 관련된 장점에 따라 화학 구조의 순위를 매기는 수단
- 참조: https://www.rdkit.org/docs/source/rdkit.Chem.QED.html

Property

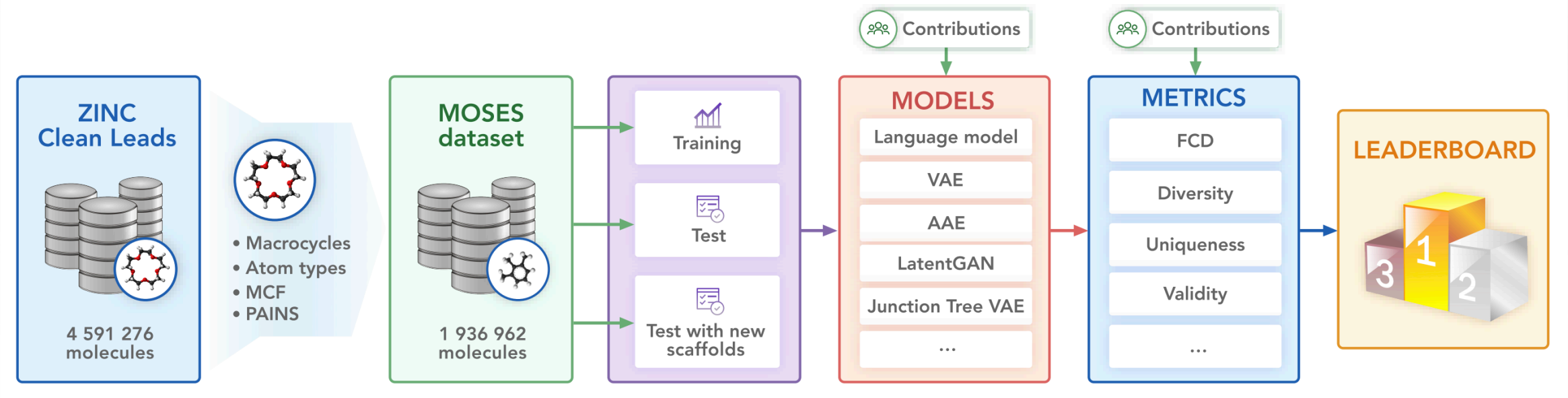
- SAS : 합성(종합) 접근성 점수(synthetic accessibility score)

  ~~- 논문 "Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions"와 "Prediction of Physicochemical Parameters by Atomic Contributions"에 기반~~

  - **1-10 사이의 값**으로 계산 가능
  - SAscore = fragmentscore - complexitypenalty
  - fragmentscore (fragment의 유사도를 측정)
  - complexitypenalty (ringComplexityScore, stereoComplexityScore, macrocyclePenalty, sizePenalty 의 조합)
  - PubChem 데이터베이스에서 934,046 개의 대표 분자를 fragment화 하여 계산
  - 참조 : https://depth-first.com/articles/2010/10/28/predicting-synthetic-accessibility/

DataSet & Parameter Setting

- **ZINC Dataset** - 캘리포니아 대학의 제약 화학과 Shoichet Laboratory의 John Irwin에 의해 개발
  - 가상 스크리닝을 위해 상업적으로 이용 가능한 화합물의 무료 데이터베이스
- http://zinc15.docking.org/
- ZINC 데이터베이스에서 벤치마크된 데이터셋을 사용
- ZINC Clean Leads 컬렉션을 기반 - 1,936,962 개



Code

Scripts – 모델 호출을 위한 기본 구성

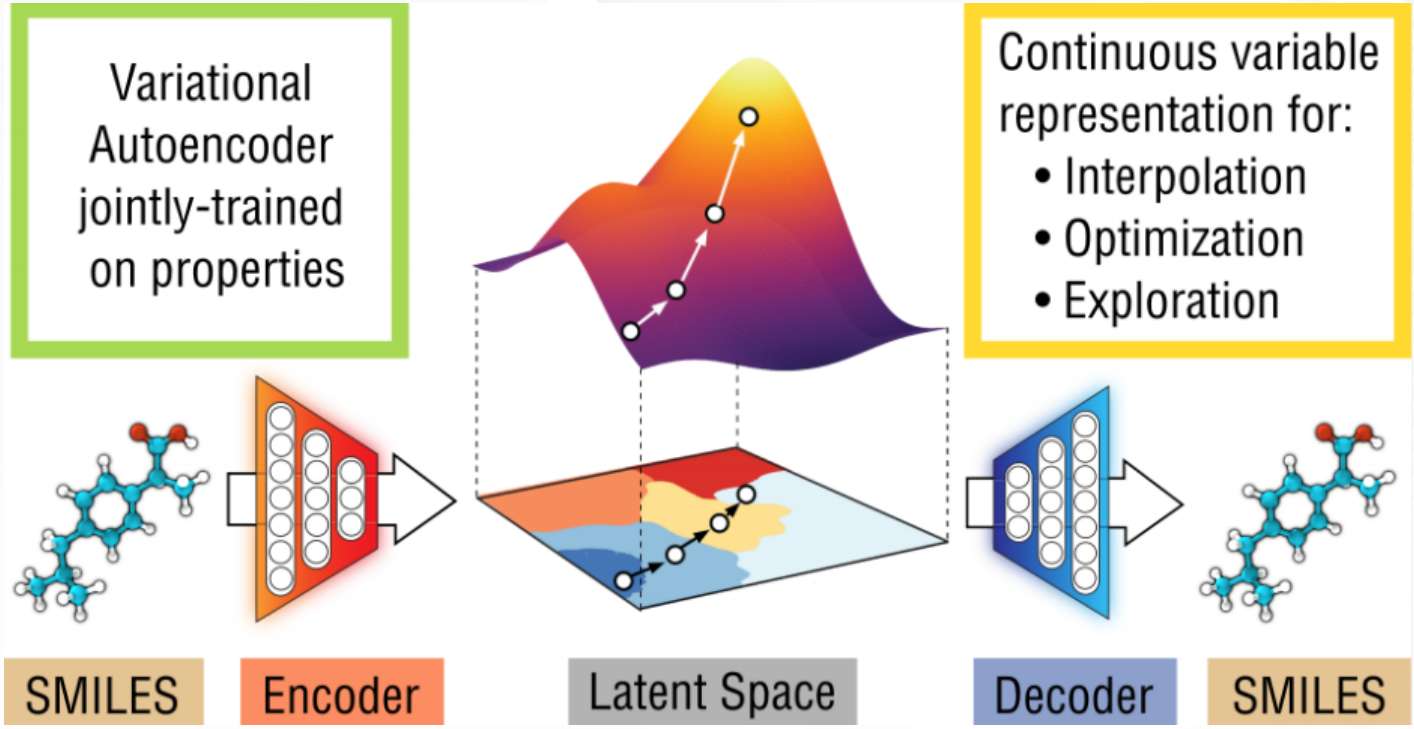
- Scripts/run.py – 데이터 실행을 위한 train.py 호출 및 configuration 명령 Parse
- Scripts/ train.py – 학습할 모델(VAE, AAE 등)의 호출, 모델 구성을 위해 default로 선언된 configuration 호출
- Scripts/ eval.py – 평가를 위한 함수 호출 및 configuration 명령 Parse

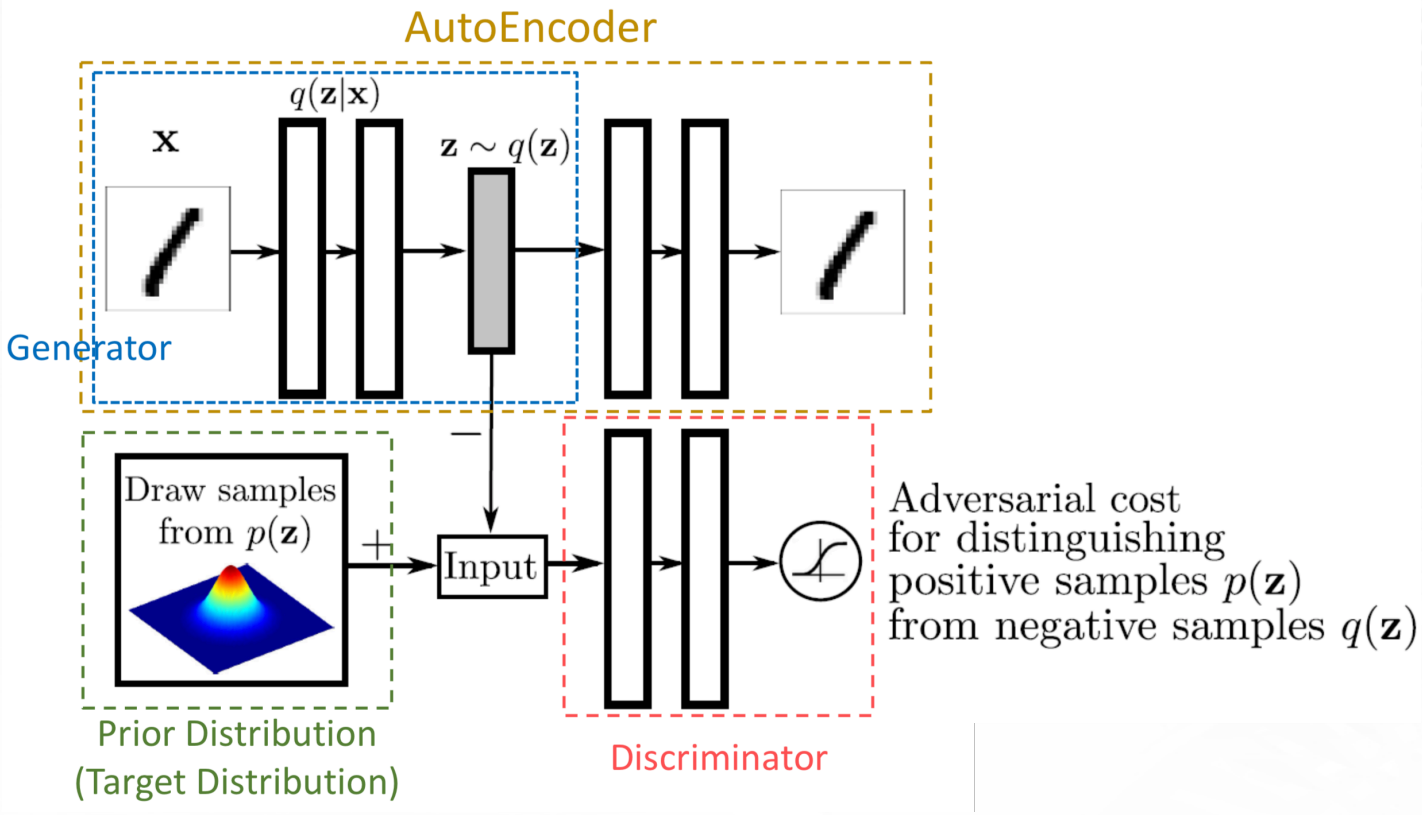
moses 디렉토리 안에 모델(VAE, AAE 등)을 구성하는 python file 존재

- moses/models_storage.py – 각 모델을 호출하기 위한 징검다리
- moses/utils.py – vocabulary(smiles변환) 정의

Code - VAE

- moses/vae/config.py. – 모델 호출 시 같이 호출해야 하는 default configuration
- moses/vae/trainer.py – 실질적인 모델의 학습 동작부
- moses/vae/model.py. – VAE 모델 선언, VAE



Code - AAE

- moses/vae/config.py. – 모델 호출 시 같이 호출해야 하는 default configuration
- moses/vae/trainer.py – 실질적인 모델의 학습 동작부
- moses/vae/model.py. – VAE 모델 선언, VAE