

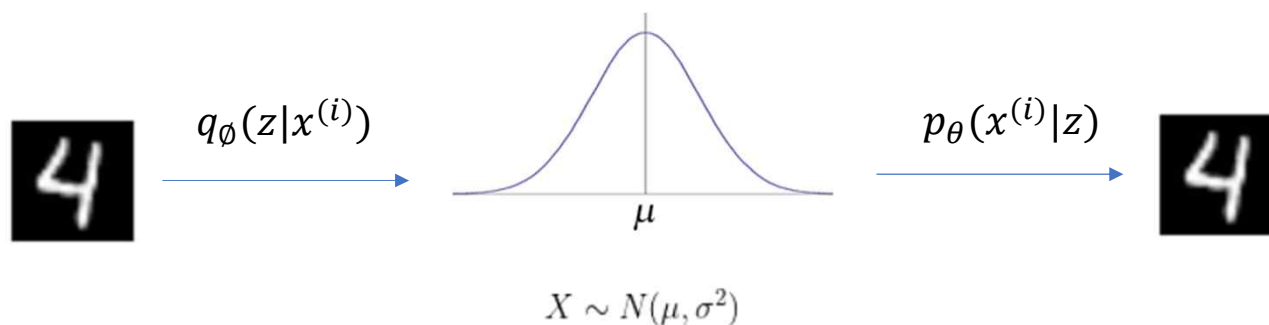
Variational Autoencoder

What is VAE (Variational Autoencoder) ?

- VAE는 통계학의 변분 추론 기법 + Deep Learning 의 Autoencoder
- VAE는 학습중인 인코딩 표현에 제약이 추가된 autoencoder
→ 입력 데이터에 대한 잠재 변수 모델을 학습하는 오토인코더
- 신경망이 임의의 함수를 학습하게하는 대신 데이터를 모델링하는 확률 분포의 매개 변수를 학습하고, 이 분포에서 샘플링하면 새로운 입력 데이터 샘플을 생성할 수 있다.
- VAE는 "확률적 생성 모델" 이고, Autoencoder는 "결정 모델"

생성 모델(Generative Model)이란 ?

- Training data가 주어졌을 때 이 training data가 가지는 real 분포와 같은 분포에서 sampling된 값으로 new data를 생성하는 model
- 입력 \rightarrow 잠재변수 분포를 생성 $q_{\phi}(z|x^{(i)})$
- 잠재변수로부터 sampling \rightarrow 입력에 가까운 출력 생성 $p_{\theta}(x^{(i)}|z)$

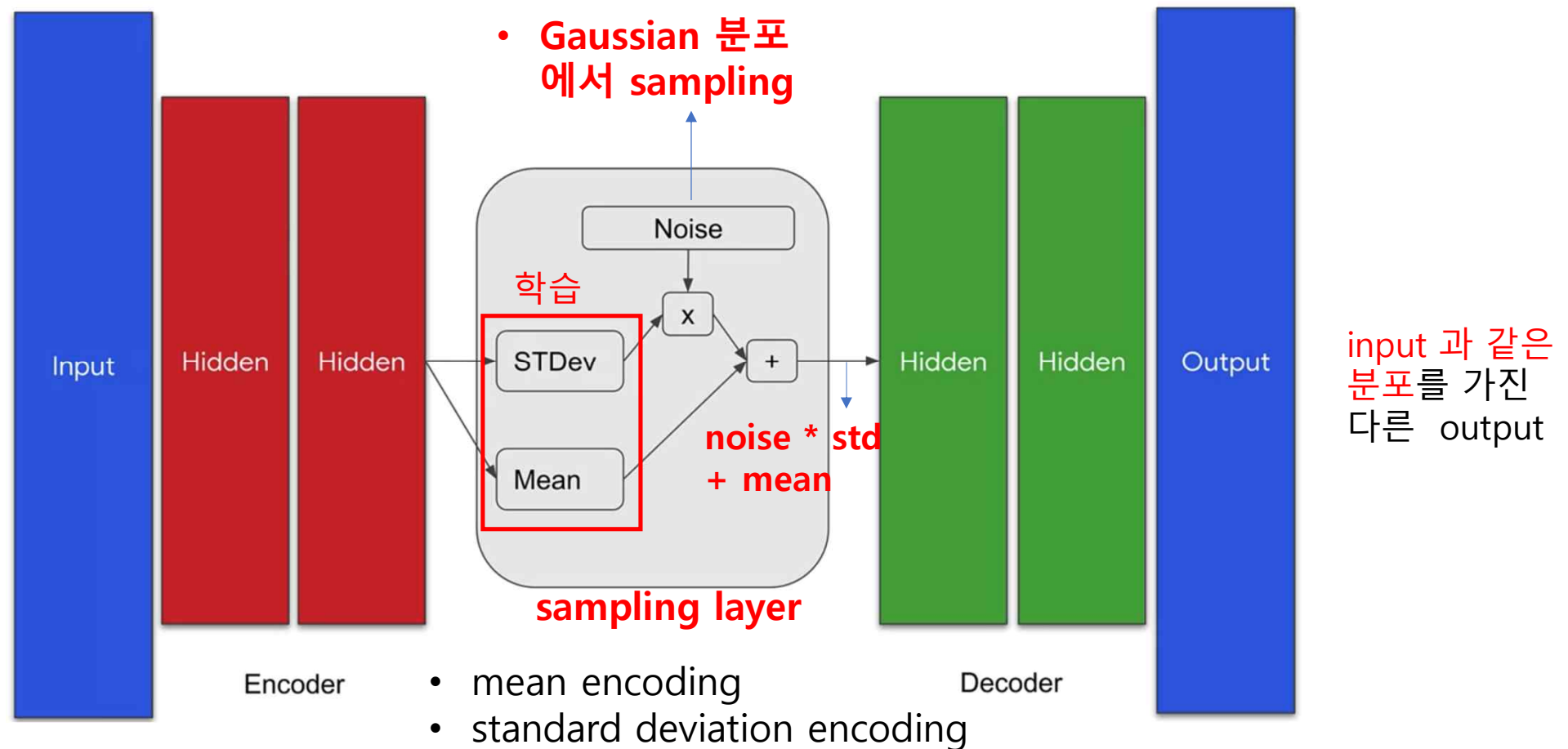


Autoencoder vs. VAE 비교 요약

	Autoencoder (AE)	Variational Autoencoder (VAE)
기본 아이디어	입력 데이터를 압축한 후 원래대로 복원하는 구조	AE의 아이디어 + 데이터의 확률 분포를 학습
인코더의 출력	Deterministic (결정론적)인 압축된 표현	평균과 로그 분산 (표준편차)의 두 가지 값 (확률 분포 파라미터)
목적함수	MSE (평균 제곱 오차) 등의 손실 함수 사용	복원 손실 + KL 발산
용도	차원 축소, 노이즈 제거, 특징 추출 등	생성 모델링, 데이터의 내재된 확률 분포 학습, 조건부 생성 등
특징	<ul style="list-style-type: none">일반적으로 데이터를 잘 복원하지만, 새로운 샘플 생성 능력은 제한적훈련 데이터에 과적합되는 경향이 강함	<ul style="list-style-type: none">데이터의 내재된 분포를 학습하므로, 새로운 샘플을 생성하는 데 더 유용과적합 완화

Variational Autoencoders

Backpropagation을 위한
reparameterization 기법



Bayes 정리 → VAE를 이해하기 위해 필요

$$\text{Posterior probability} = \frac{\text{Prior probability} \times \text{Likelihood}}{\text{Evidence}}$$

유도과정

$$\text{Pr}(y | X) = \frac{\text{Pr}(y) \times \text{Pr}(X | y)}{\text{Pr}(X)}$$

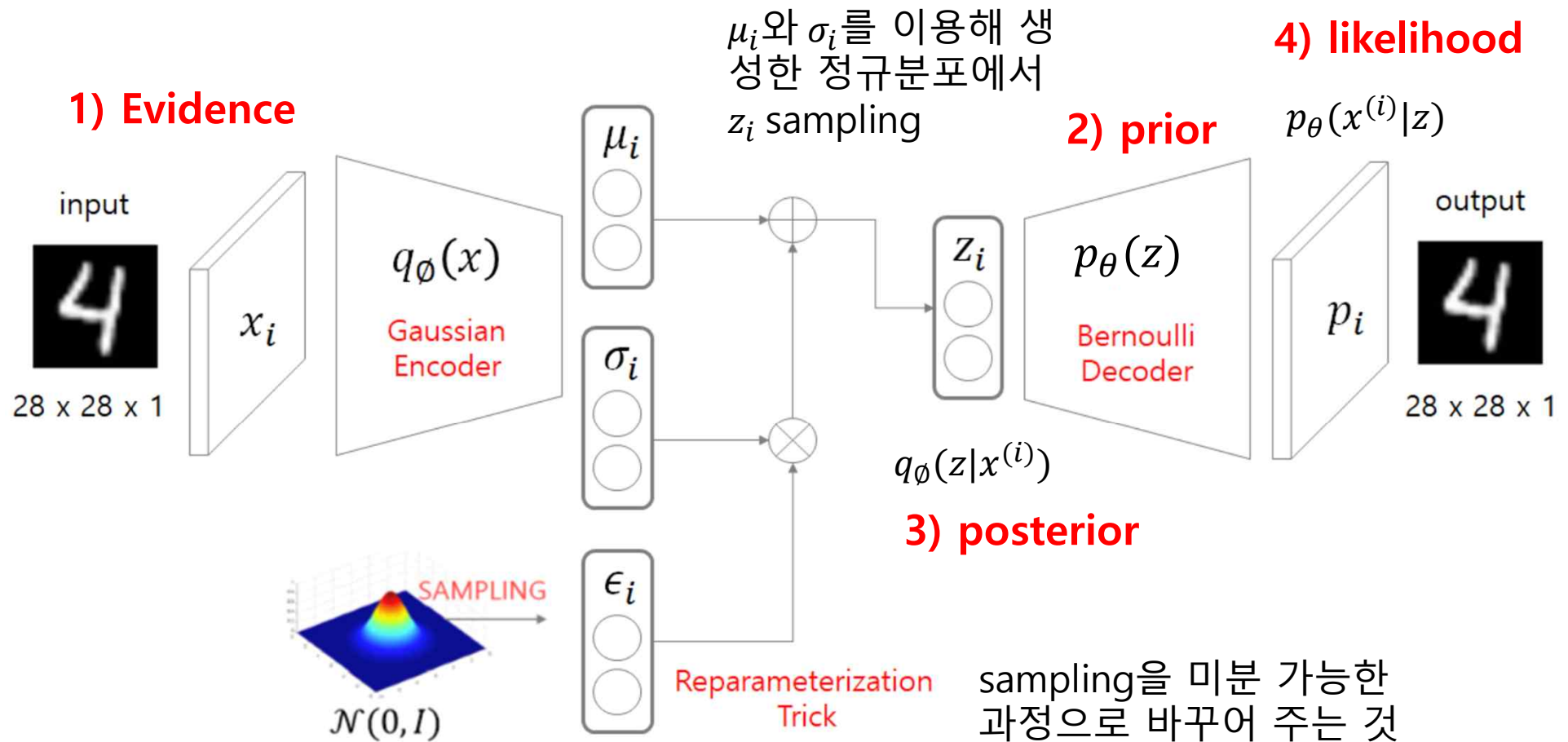
X는 관측하여 알아낼 수 있는 대상. (X : feature)
y는 확률을 계산해야 하는 불확실한 대상 (y : class label)

사전 확률로부터 사후 확률을 구할 수 있음.

- $\text{Pr}(y)$: y의 **사전 확률**. 아직 X에 관한 어떤 정보도 알지 못하는 것을 의미.
- $\text{Pr}(y|X)$: X의 값이 주어진 경우에 대한 y의 **사후 확률**
- $\text{Pr}(X|y)$: y가 주어졌을 때 X의 조건부 확률
- $\text{Pr}(X)$: X의 사전 확률이며 정규화 상수 역할. $\int_y \text{Pr}(X|y)$ 로 구할 수 있다.

- 사전 확률 (Prior Probability)
 - 현재 가지고 있는 정보를 기초로 하여 정한 초기 확률.
확률 시행 전에 이미 가지고 있는 지식을 통해 부여한 확률
ex) 초기 정보(정상적인 동전일 것)로 추정한 앞면이 나올 확률 $\rightarrow 0.5$
- 사후 확률 (Posterior Probability)
 - 사건 발생 후에 추가된 정보로부터 사전 정보를 새롭게 수정한 확률.
ex) 10번 던져 9번 앞면이 나오면 동전이 정상적일 것이라는 초기 믿음
업데이트 필요
 - 조건부 확률을 통해 사후 확률을 표현
 - 사후 확률은 Bayes 정리로 부터 구할 수 있다.

VAE 의 구조



1. Evidence (증거) – $\Pr(X)$

데이터 x 가 관찰될 전체 확률을 의미. VAE에서는 이 값을 직접 계산하는 대신, 변분 하한(Evidence Lower Bound, ELBO)을 최적화함으로써 간접적으로 다룹니다.

2. Prior (사전 확률) – $\Pr(z)$

VAE에서 이는 잠재 변수 z 의 분포가 어떠할 것이라는 모델의 초기 가정. 보통 표준 정규 분포 사용.

3. Posterior (사후 확률) – $\Pr(z|X)$

관측 데이터 x 가 주어졌을 때 잠재 변수 z 의 분포 – Encoder 로 모델링

4. Likelihood (가능도) – $\Pr(X|y)$

잠재 변수 z 가 주어졌을 때 실제 데이터 x 가 나타날 확률 – Decoder 로 모델링

변분추론(Variational Inference)

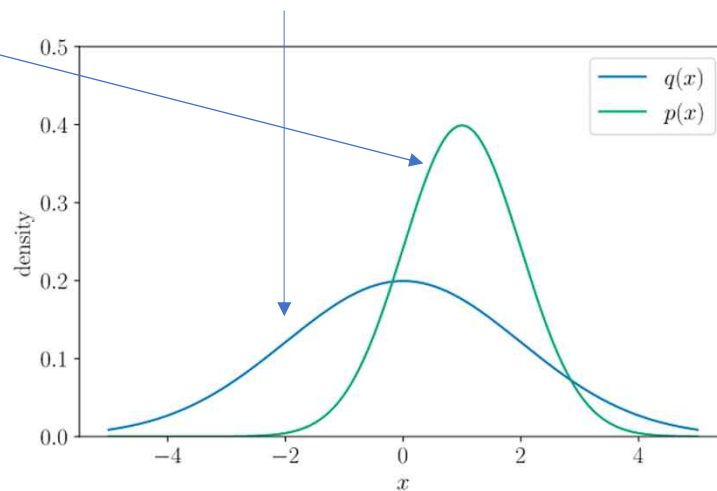
- **베이지안 모델**링에서 간단한 확률 분포(**정규 분포**)로 부터 **사후 확률 분포**를 추정하는 방법
- 변분추론의 핵심 아이디어
 1. 근사 분포 선택 – 대개 정규 분포
 2. **KL 발산**을 최소화 하는 매개 변수 찾기
 3. KL 발산을 직접 최소화하는 것은 계산이 어려우므로 **ELBO(Evidence Lower Bound)**를 최대화 하는 방식으로 진행
 4. ELBO는 원래의 **로그우도(log likelihood)**에서 KL 발산을 뺀 값이며, 이 값을 최대화 하는 것은 KL 발산을 최소화 하는 것과 동일

VAE에서 변분추론의 역할

- 인코더는 데이터에 대해 가정한 확률 분포의 파라미터(예: 평균과 분산)를 출력
- AE에서 인코더가 출력하는 이러한 파라미터는 잠재 변수에 대한 사후 확률 분포를 근사하는 데 사용됩니다.
- 변분추론은 이 근사 사후 확률 분포가 실제 데이터에서 얻어진 잠재 변수의 분포와 가까워지도록 합니다. 이를 위해 KL 발산을 최소화하는 방향으로 인코더의 파라미터를 조정합니다.

KL Divergence (쿨백-라이블러 발산)

- VAE 손실 함수의 일부
- 두 확률 분포의 차이 값
 - 모델링 오류에서 발생한 오차(추가 비용)를 의미
- $p(x) \rightarrow$ true 확률 분포, $q(x) \rightarrow$ 모델이 추정한 확률 분포



VAE의 손실 함수 - ELBO

- KL 항과 재구성 손실로 구성된 ELBO(Evidence Lower bound) 최소화
- $p_\theta(x|z)$ 가 multivariate-Bernoulli를 따른다고 가정하면 cross entropy를 사용.
normal distribution을 따른다고 가정하면 MSE를 사용.

$$\mathcal{L} = \underbrace{-\mathbb{E}_{z \sim q_\phi(z|x^{[i]})} [\log p_\theta(x^{[i]}|z)]}_{\text{재구성 항}} + \underbrace{KL(q_\phi(z|x^{[i]}) || p_\theta(z))}_{\text{정규화(regularization) 항}}$$

ELBO

Encoder 분포에 대한
negative log likelihood 기대값 항
→ maximize

KL 발산 항 → minimize

[유도과정](#)

3 개의 확률 분포를 modeling 해야 한다.

prior $p(z)$

variational encoder $q(z|x)$

decoder $p(x|z)$

손실함수 구현

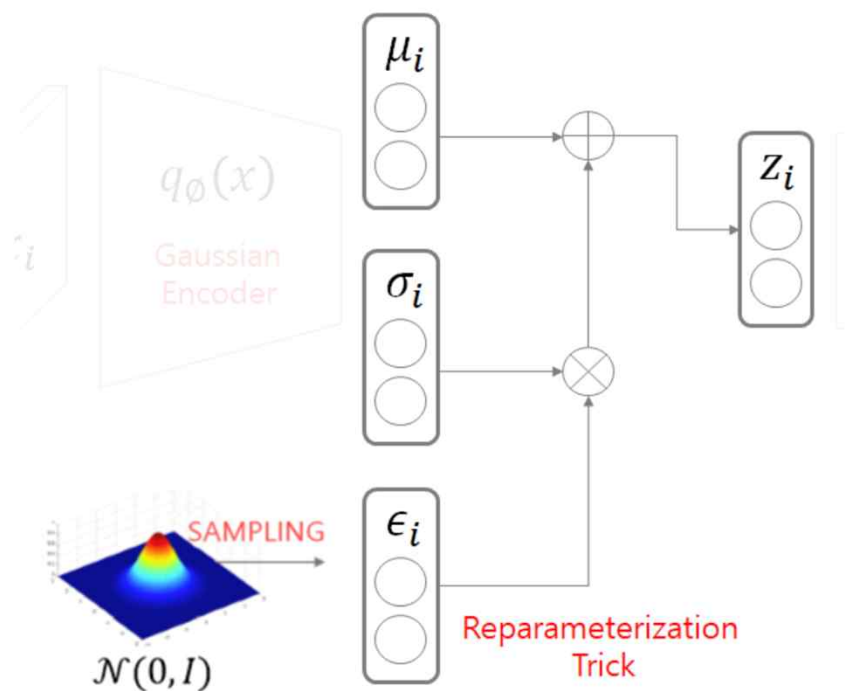
$$\mathcal{L} = \underbrace{-\mathbb{E}_{z \sim q_{\phi}(z|x^{[i]})} [\log p_{\theta}(x^{[i]}|z)]}_{\text{재구성 항}} + \underbrace{KL(q_{\phi}(z|x^{[i]}) || p_{\theta}(z))}_{\text{정규화(regularization) 항}}$$

ELBO

이진 교차 엔트로피 손실 함수를 사용합니다.
`bce_loss = tf.keras.losses.BinaryCrossentropy()`

KLD 계산 공식 적용
`kl_loss = 1 + sigma - tf.square(mu) - tf.math.exp(sigma)`
배치 내 평균을 계산하고 -0.5를 곱하여 최종 KLD 손실을 계산
`kl_loss = tf.reduce_mean(kl_loss) * -0.5`

Reparameterization Trick (Sampling)



평균 - 0
표준편차 - 1

인코더의 출력을 분해합니다.

```
mu, sigma = inputs
```

배치의 크기와 차원을 얻습니다.

```
batch = tf.shape(mu)[0]
```

```
dim = tf.shape(mu)[1]
```

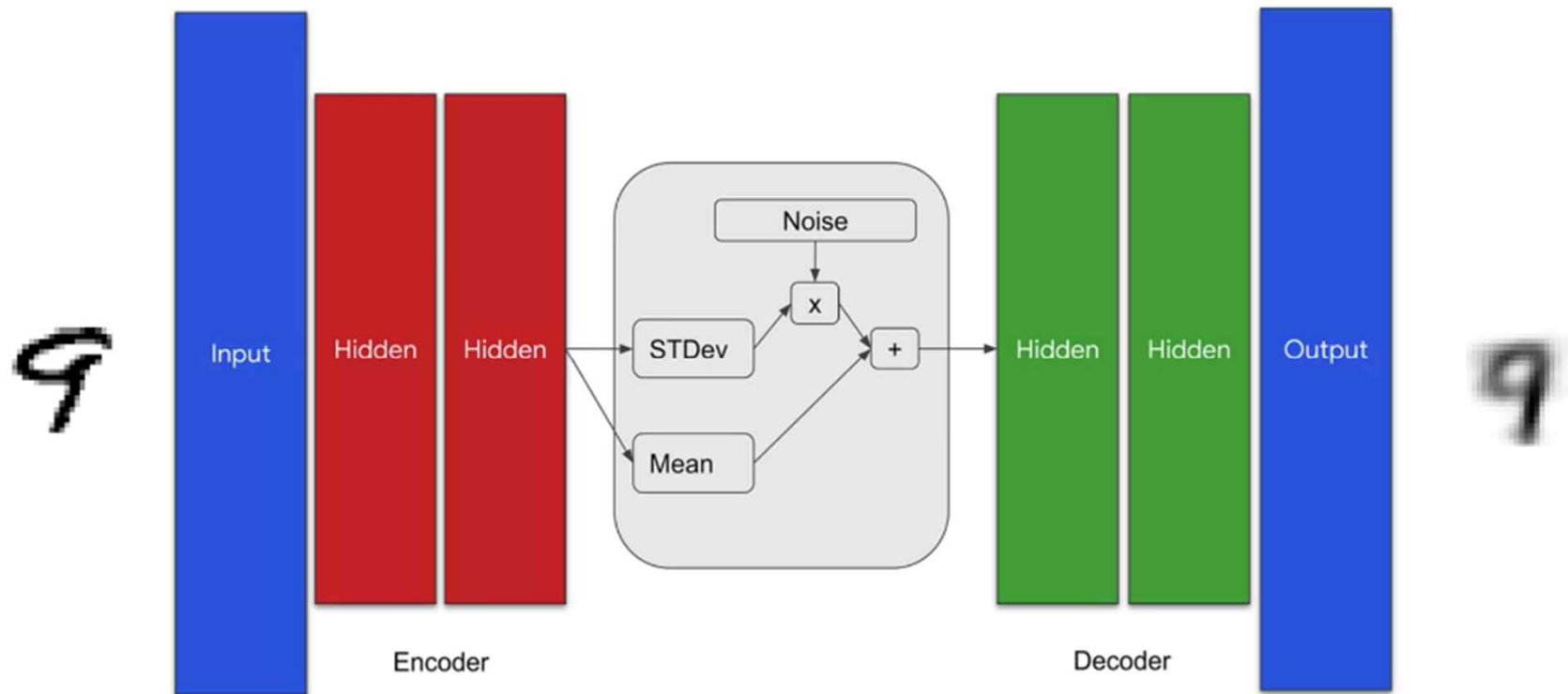
무작위 텐서를 생성합니다.

```
epsilon = tf.keras.backend.random_normal(shape=(batch, dim))
```

재매개변수화 기법을 적용합니다.

```
return mu + tf.math.exp(0.5 * sigma) * epsilon
```

실습 : 225. VAE mnist



실습: 255. VAE를 이용한 Fraud Detection

1. Variational Autoencoder 훈련
2. 잠재 표현 시각화 (Visualize Latent Representations)
3. VAE model 훈련
4. 재구성 오류 계산
5. 테스트 세트에 대한 평가