Autoencoder

한양대학교 인공지능연구실 인공지능세미나

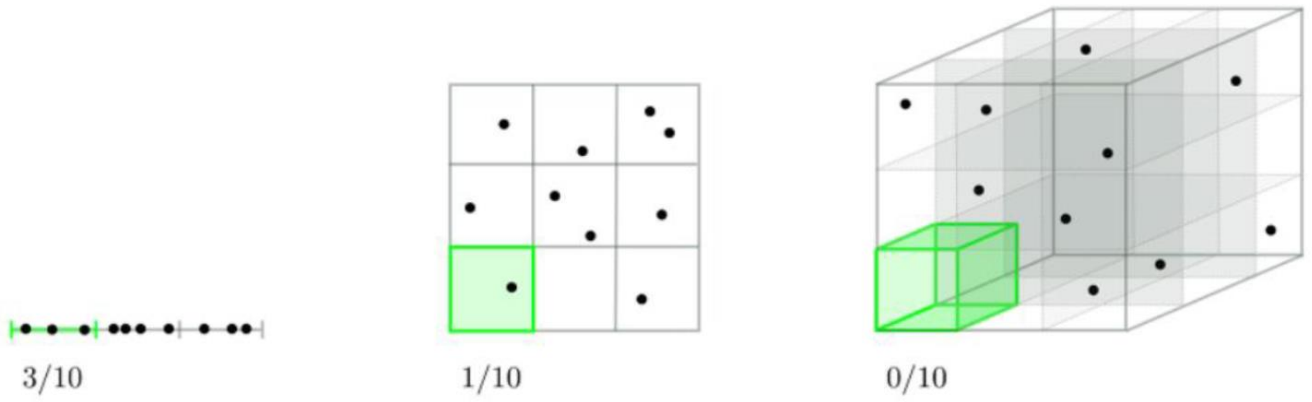
20220119
AILAB 조환희

Manifold Learning

Curse of Dimensionality

Deep learning 큰 데이터 차원으로 학습

1. 데이터 차원이 증가할수록 공간의 크기 기하급수적으로 증가
2. 동일한 개수의 데이터의 밀도는 차원이 증가할수록 급속히 희박



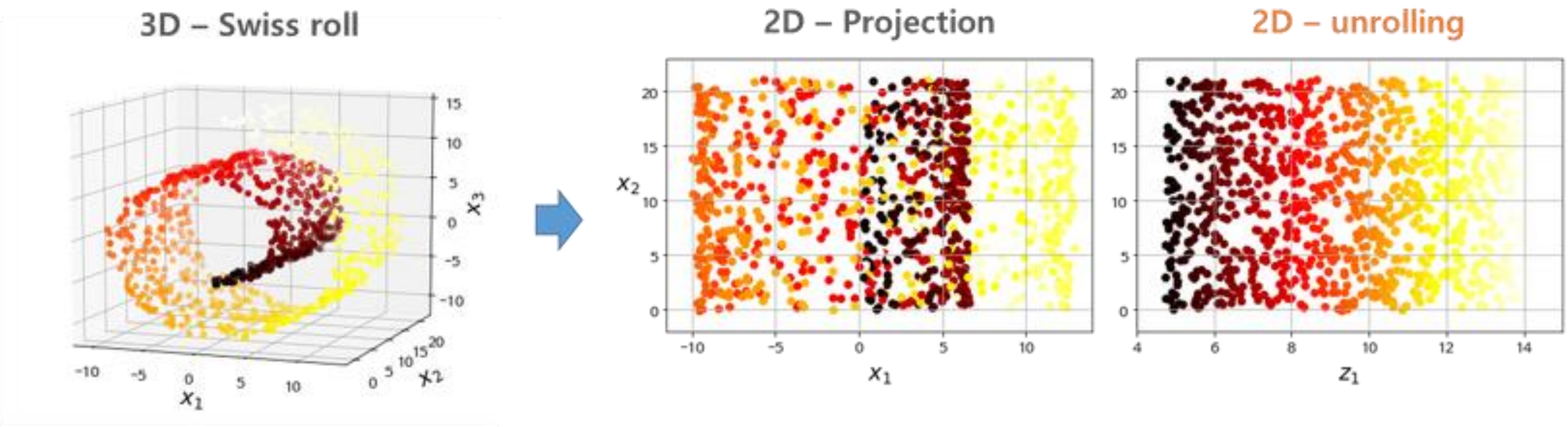
→ 차원이 증가할수록 모델 추정에 필요한 샘플 데이터가 기하급수적으로 필요!

Manifold Learning

Manifold

고차원 space에 있는 데이터를 나타낼 수 있는 저차원 subspace인 manifold가 존재함

Manifold learning



Manifold Learning

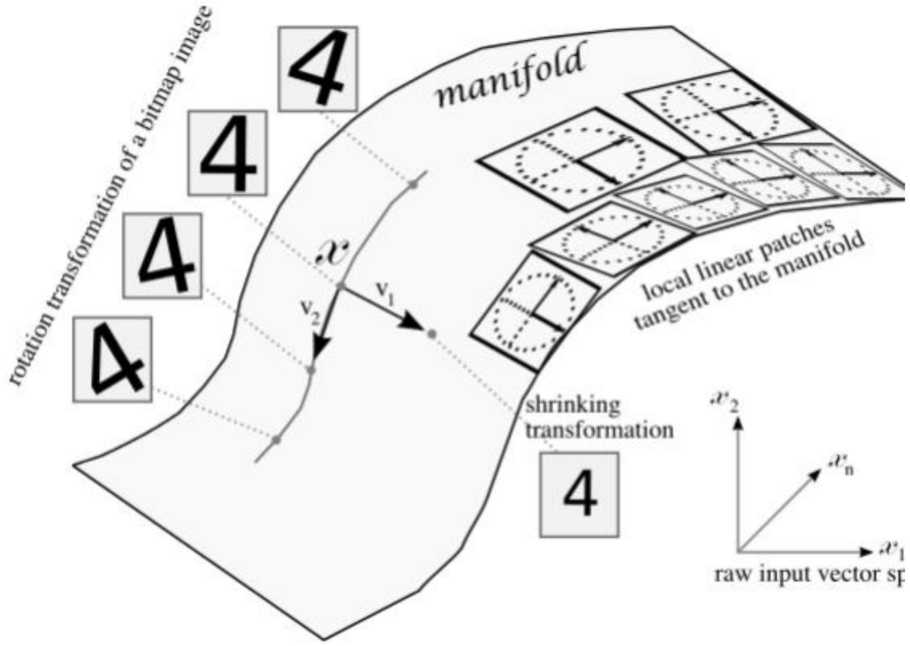
Learning을 잘했다

==압축을 잘했다==feature를 잘 찾았다.

MNIST dataset에서 manifold를 찾았을 때

그 manifold를 dataset의 feature라고 가정할 수 있다.

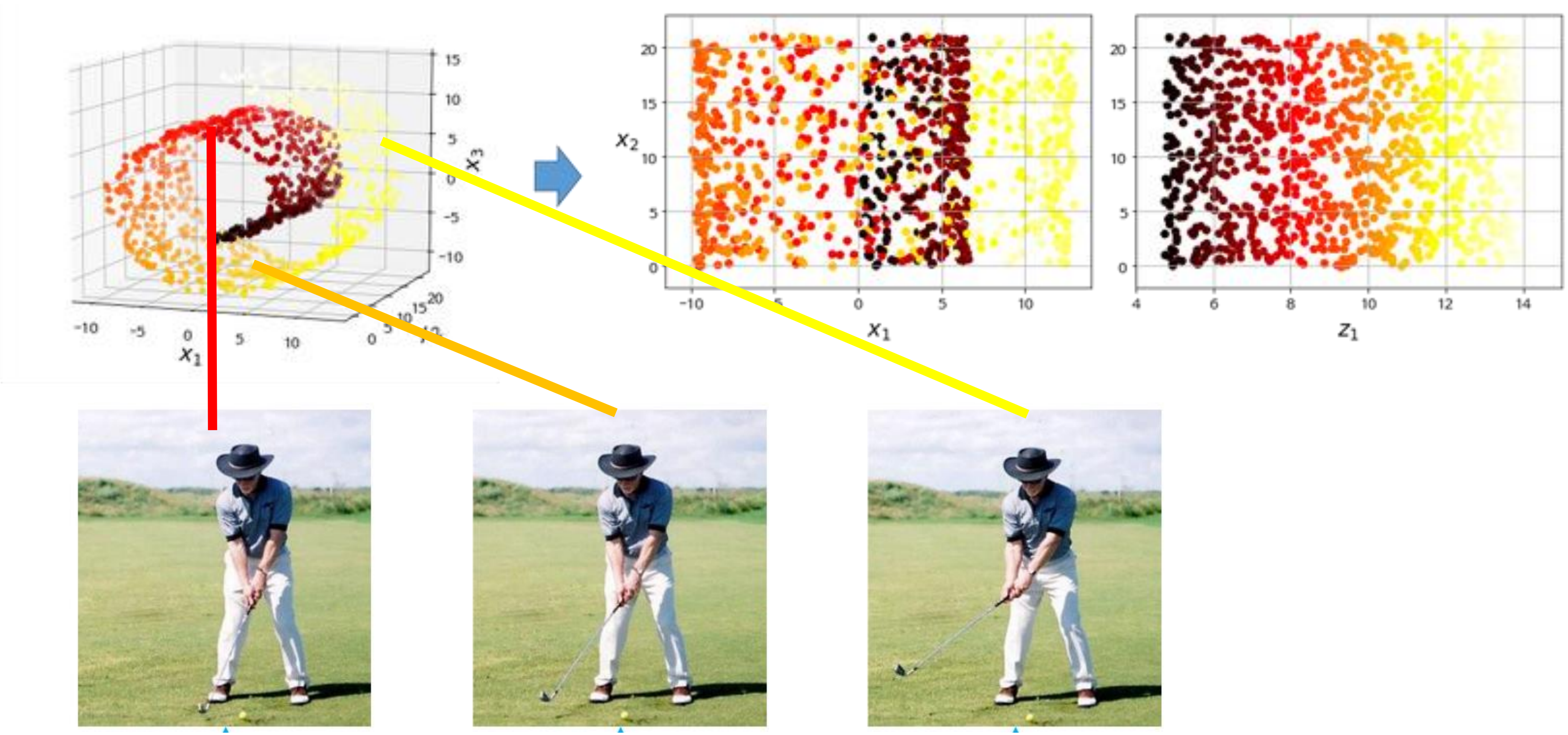
feature를 잘 찾았다면 하나의 축을 변경했더니 rotation이 된다든 지의 변화를 확인 할 수 있다.



Manifold Learning

Feature를 잘 찾았다.

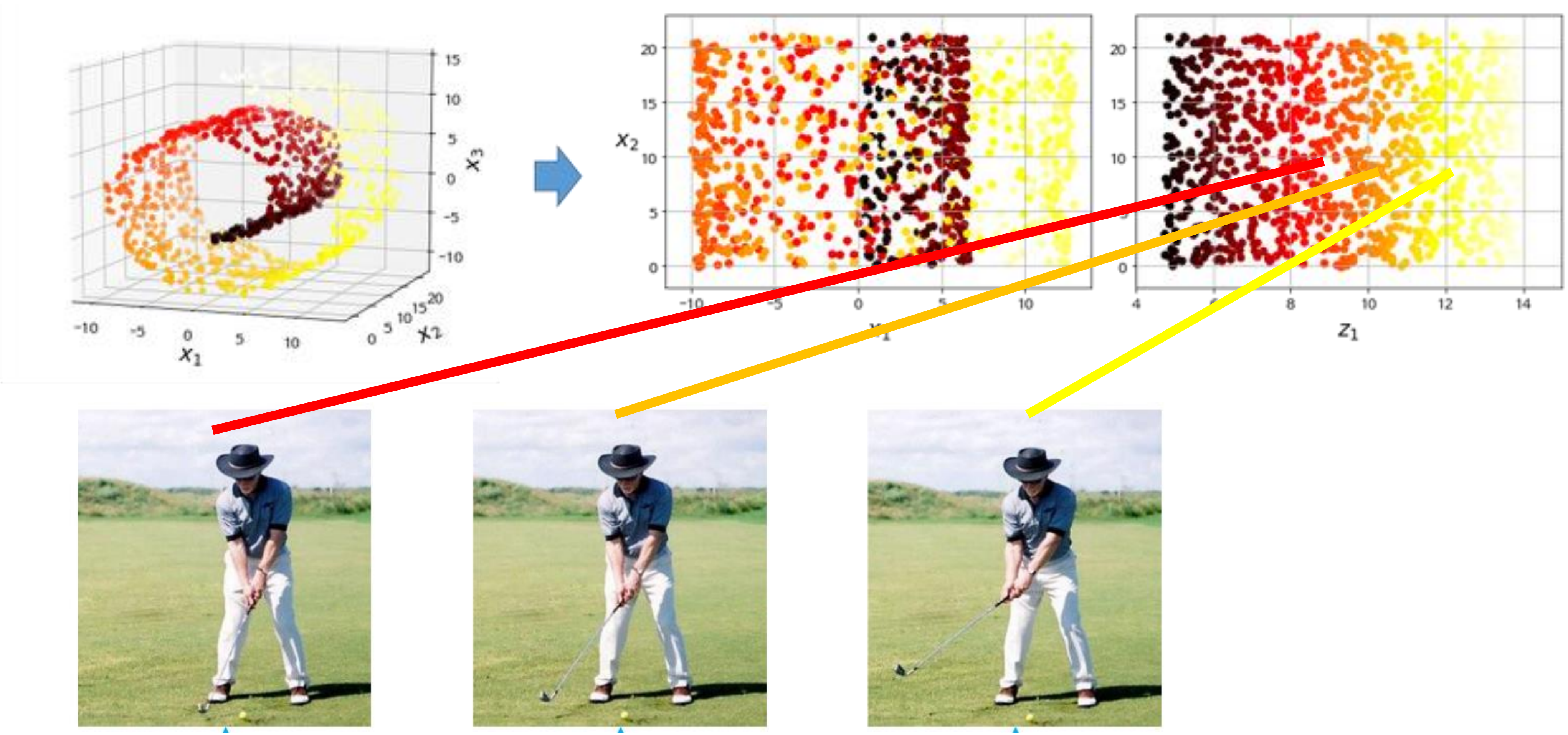
==manifold에서 의미적으로 비슷한 데이터가 가깝다



Manifold Learning

Feature를 잘 찾았다.

==manifold에서 의미적으로 비슷한 데이터가 가깝다



Autoencoder

An **autoencoder** is a type of artificial neural network used to **learn efficient codings** of **unlabeled data**.

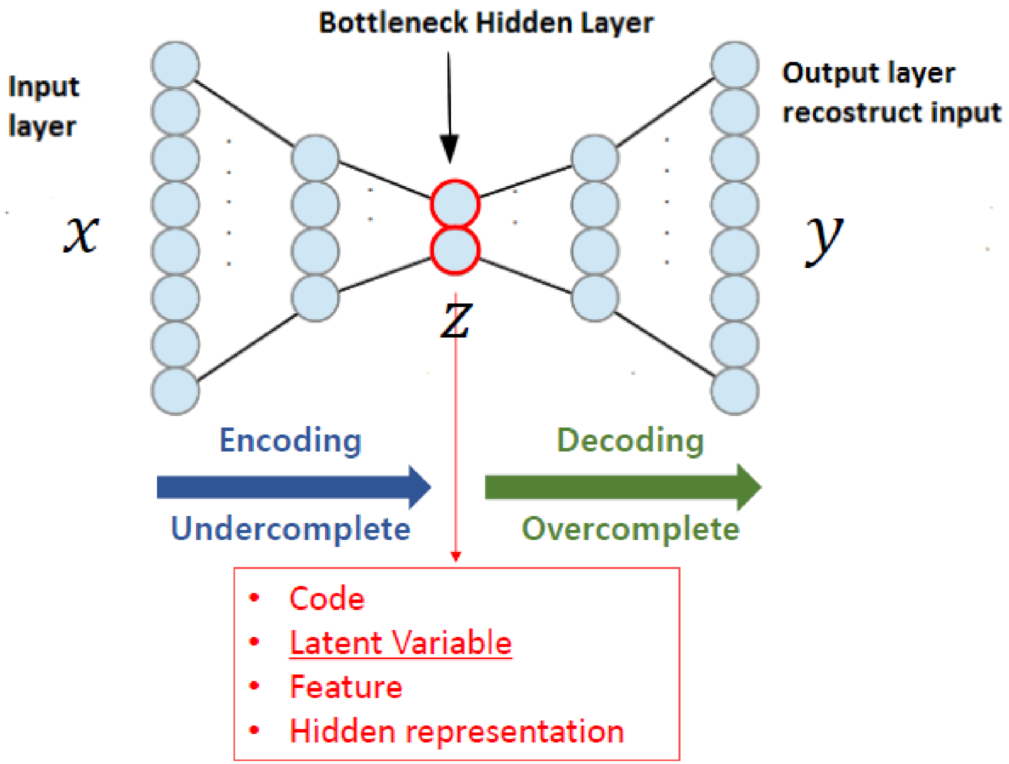
Autoencoder

ENCODER

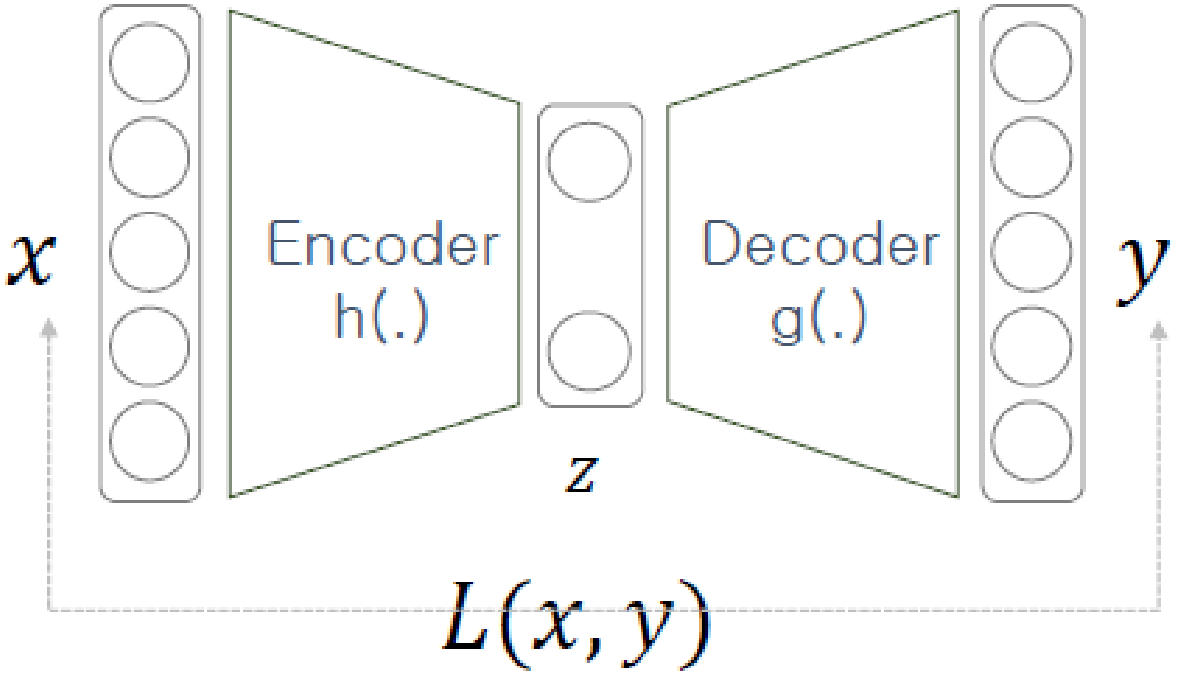
maps input into code

DECODER

maps code to reconstruction of the input



Autoencoder



$$z = h(x) \in \mathbb{R}^{d_z}$$

$$y = g(z) = g(h(x))$$

$$L_{AE} = \sum_{x \in D} L(x, y)$$

reconstruction error $L(x, y)$ ← $\|x - y\|^2$ or cross-entropy

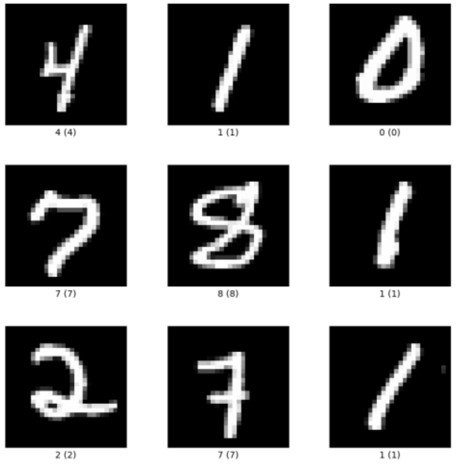
Minimize $L_{AE} = \sum_{x \in D} L(x, g(h(x)))$

Hidden layer 1개이고 레이어 간 fully-connected로 연결된 구조

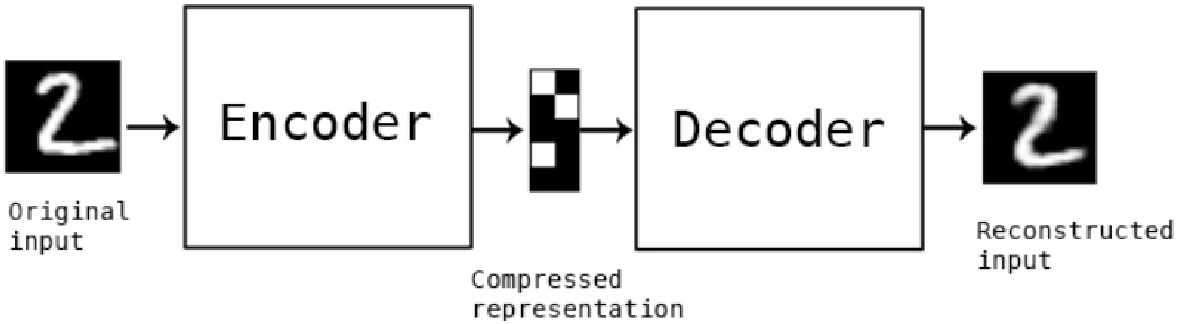
Autoencoder

MNIST dataset 학습 코드

https://colab.research.google.com/drive/13mloqpMcN_23cAZc1zullpFcgC9b-C94?usp=sharing



MNIST dataset



MNIST data를 잘 reconstruction 하는지 확인

Autoencoder

MNIST dataset 학습 코드

1. 모델 정의

Fully connected 4-layer encoder, decoder

```
class autoencoder(nn.Module):
    def __init__(self):
        super(autoencoder, self).__init__()
        self.encoder = nn.Sequential(
            nn.Linear(28 * 28, 128),
            nn.ReLU(True),
            nn.Linear(128, 64),
            nn.ReLU(True), nn.Linear(64, 12), nn.ReLU(True), nn.Linear(12, 3))
        self.decoder = nn.Sequential(
            nn.Linear(3, 12),
            nn.ReLU(True),
            nn.Linear(12, 64),
            nn.ReLU(True),
            nn.Linear(64, 128),
            nn.ReLU(True), nn.Linear(128, 28 * 28), nn.Tanh())

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)
        return x
```

Autoencoder

MNIST dataset 학습 코드

2. 모델 선언 및 학습

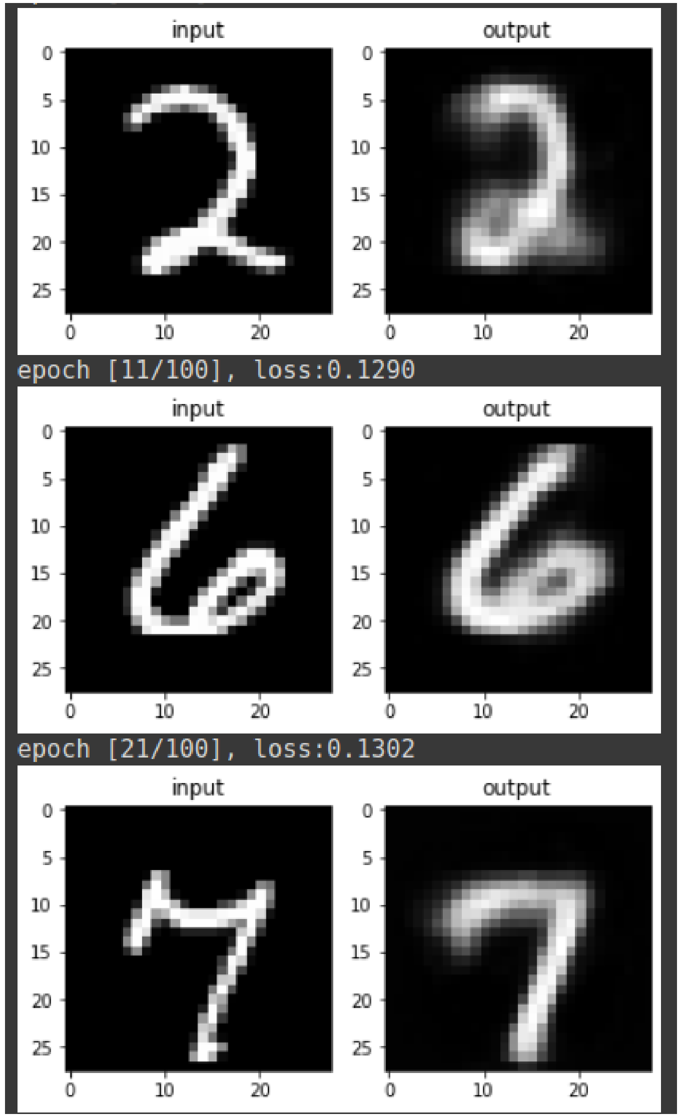
```
model = autoencoder().cuda()
criterion = nn.MSELoss()
optimizer = torch.optim.Adam(
    model.parameters(), lr=learning_rate, weight_decay=1e-5)
```

```
for epoch in range(num_epochs):
    for data in dataloader:
        img_, _ = data
        img = img_.view(img_.size(0), -1)
        img = Variable(img).cuda()
        # ===================forward=====================
        output = model(img)
        loss = criterion(output, img)
        # ===================backward====================
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

Autoencoder

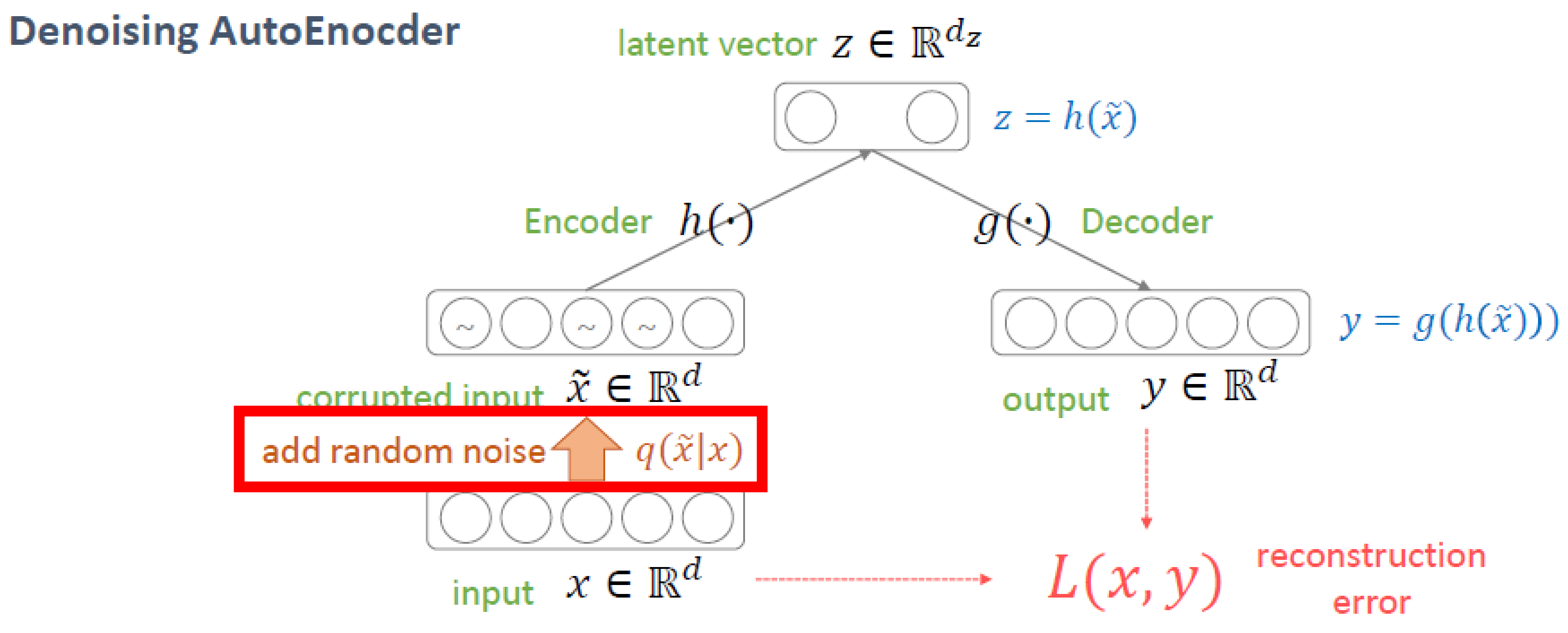
MNIST dataset 학습 코드

학습 과정



Denoising Autoencoder

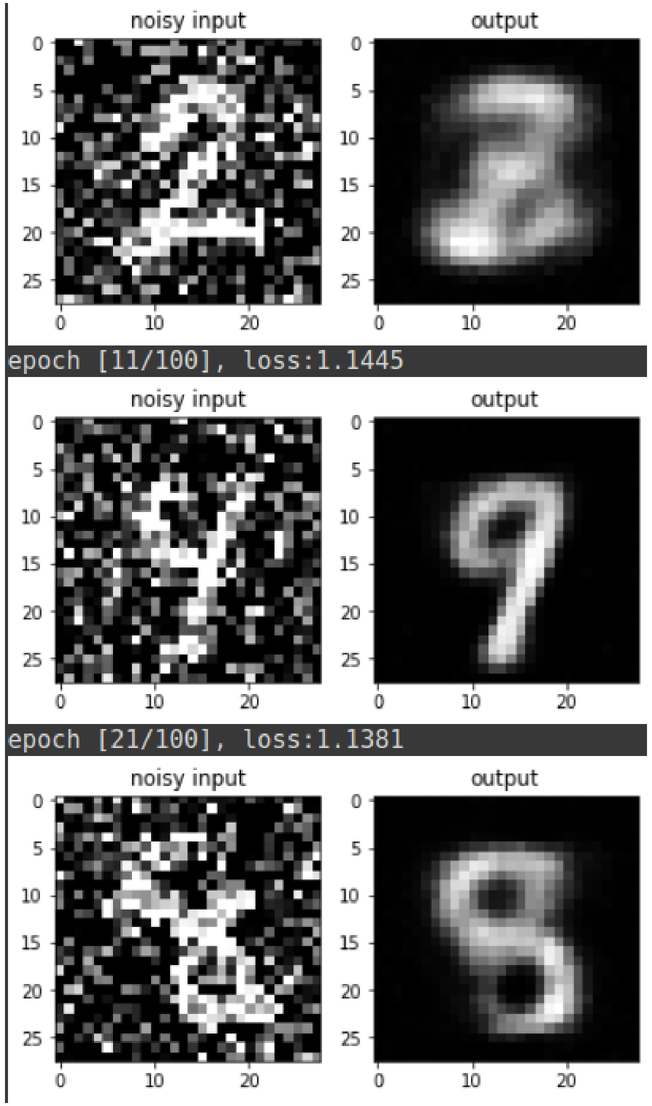
노이즈를 추가해도 의미적으로 원본 이미지와 같으므로 manifold 학습가능



Minimize $L_{DAE} = \sum_{x \in D} E_{q(\tilde{x}|x)} \left[L(x, g(h(\tilde{x}))) \right]$

Denoising Autoencoder

MNIST dataset 학습 코드



Summary

Manifold learning

Autoencoder

Denoising Autoencoder

Question?

Thank You!