

딥러닝의 통계적 이해

9강. 오토인코더와 GAN(1)

1. 비지도 학습법
2. 주성분 분석
3. 오토 인코더

서울대학교 김용대 교수

오늘의 학습목표

1. 비지도 학습이 무엇인지 이해할 수 있다.
2. 주성분 분석 방법론을 이해할 수 있다.
3. 오토 인코더 방법론을 이해할 수 있다.

1. 비지도 학습법

1. 비지도 학습법

비지도 학습법

- ◆ Unsupervised learning
- ◆ 기계 학습의 일종.
- ◆ 데이터가 어떤 특징을 가지고 있는지를 알아내는 것이 목표.
- ◆ 차원축소, 밀도 추정, 군집 분석, 독립성 분석 등

1. 비지도 학습법

비지도 학습법

- ◆ 확률 모형 기반 X
 - 주성분 분석
 - 군집 분석 등
- ◆ 확률 모형 기반 O
 - 인자 분석
 - 밀도 추정 등

1. 비지도 학습법

심층 비지도 학습법

- ◆ Deep unsupervised learning
- ◆ 딥러닝 (deep learning)의 일종.
- ◆ 심층 신경망 모형에 기초한 비지도 학습 방법론을 총칭

1. 비지도 학습법

심층 비지도 학습법

◆ 확률 모형 기반 X

- Auto Encoder
- Deep clustering 등

◆ 확률 모형 기반 O

- Generative Adversarial Network,
- Variational Auto Encoder 등

1. 비지도 학습법

비지도 학습법

- ◆ 본 강연에서는 비지도 학습 방법 중 차원축소 방법인 주성분 분석과 오토 인코더를 다룰 예정.

변환 함수

확률 모형
기반 여부

	변환 함수	
	선형 함수	심층 신경망 함수
X	주성분 분석	오토 인코더
O	인자 분석	GAN

2. 주성분 분석

2. 주성분 분석

차원 축소 기법

- ◆ 분석 대상이 되는 변수의 수를 줄이는 일련의 탐색적 자료 분석의 과정.
- ◆ 고차원 자료 분석에서 자주 사용됨.
- ◆ 변수 변환 (feature transformation)
 - 기존 변수를 조합해 새로운 변수를 만드는 기법.
- ◆ 변수 선택 (feature selection)
 - 기존 변수 중 중요한 일부 변수만을 빼내는 기법.



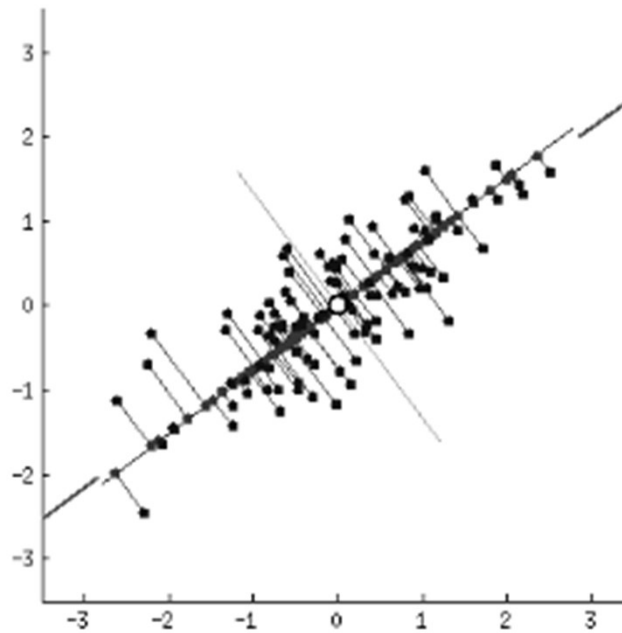
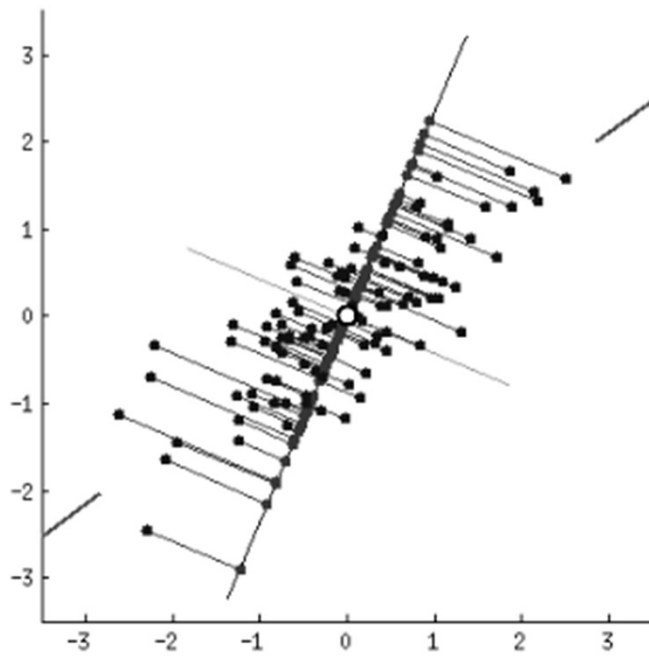
2. 주성분 분석

주성분 분석

- ◆ Principal Component Analysis (PCA)
- ◆ 대표적인 차원 축소 기법 중 하나.
 - 변수 변환에 기초.
- ◆ 원래 변수들에 대한 선형 변환을 통해 변수들을 잘 설명하는 주성분들을 찾고 이를 이용하여 데이터의 차원을 축소.
- ◆ 데이터의 분산을 최대한 보존하면서 서로 직교하는 주성분을 찾는 것이 핵심.

2. 주성분 분석

주성분 분석



◆ 핑크색 사선 축이 제 1 주성분 (PC1) 임.

2. 주성분 분석

주성분 변수의 유도 과정

- ◆ 제 1 주성분 (PC1)
- ◆ $X \in \mathbb{R}^{n \times p}$: p 차원 자료 n 개를 모은 행렬.
- ◆ $x_i \in \mathbb{R}^p$: 자료 X 의 i 번째 자료. ($i = 1, \dots, n$)
- ◆ $Var(Xa)$ 를 최대화 하는 기저 $a \in \mathbb{R}^p$ 를 찾는 것이 목표.
 - 기저 a 의 크기를 1로 제한. (즉, $a^T a = 1$)

2. 주성분 분석

주성분 변수의 유도 과정

◆ 라그랑지안 방법을 이용.

$$v_1 = \operatorname{argmax}_a a^T S a - \rho(a^T a - 1)$$

◆ S 는 자료 X 의 표본 공분산 행렬,

ρ 는 라그랑지안 승수 (Lagrange multiplier).

$$\rightarrow S = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T.$$

$$\rightarrow \mu \text{ 는 자료 } X \text{ 의 평균 벡터. } (\mu = \frac{1}{n} \sum_{i=1}^n x_i)$$

2. 주성분 분석

주성분 변수의 유도 과정

- ◆ 제 1 주성분 v_1 은 표본 분산 행렬 S 의 가장 큰 고유값 (eigenvalue) 과 대응되는 고유 벡터 (eigenvector)임을 알 수 있음.
- ◆ 고유값과 고유 벡터
 - 정방 행렬 A 의 고유값 (eigenvalue) 과 고유 벡터 (eigenvector) 는 다음 성질을 만족시키는 숫자 λ 와 벡터 v 를 의미.

$$Av = \lambda v$$

2. 주성분 분석

주성분 변수의 유도 과정

- ◆ 제 2 주성분 (PC2)
- ◆ 제 1 주성분과 직교하면서 $Var(Xa)$ 를 최대화 하는 기저 $a \in \mathbb{R}^p$ 를 찾는 것이 목표.
- ◆ 제 1 주성분과 마찬가지로 라그랑지안 방법을 사용.

$$v_2 = \operatorname{argmax}_a a^T S a - \rho(a^T a - 1) - \phi a^T v_1$$

→ 여기서 ρ, ϕ 는 라그랑지안 승수.

2. 주성분 분석

주성분 변수의 유도 과정

- ◆ 제 2 주성분 v_2 는 표본 분산 행렬 S 의 두 번째로 큰 고유값과 대응되는 고유 벡터임을 알 수 있음.
- ◆ 이와 같은 방법으로, 제 k 주성분은 표본 분산 행렬의 k 번째로 큰 고유값과 대응되는 고유 벡터임을 보일 수 있음.

2. 주성분 분석

차원 축소 데이터의 생성

◆ 자료 X 의 q 개의 주성분 v_1, \dots, v_q 를 구함 ($q < p$).

◆ $x_i \in \mathbb{R}^p$: 자료 X 의 i 번째 자료. ($i = 1, \dots, n$)

◆ 주성분 분석을 통해 q 차원으로 차원 축소된 x_i 의 값은 $(x_i^T v_1, \dots, x_i^T v_q)^T$.

◆ 차원 축소된 값을 이용해 자료 x_i 를 근사하면

$$z_i = \mu + \sum_{j=1}^q [(x_i - \mu)^T v_j] v_j.$$

2. 주성분 분석

경험 위험 최소화 관점

- ◆ 경험 위험 최소화 (Empirical Risk Minimization)
- ◆ 데이터에서 변수 간의 관계를 나타내는 함수를 추정하는 것이 목표.
- ◆ 함수의 적합성을 손실 함수 (loss function) 를 통해 수치화.
- ◆ 훈련 자료 (train data) 에서의 손실 함수의 합 (또는 평균) 을 최소화하는 함수를 선택.

2. 주성분 분석

경험 위험 최소화 관점

◆ 주성분 분석

- ✓ 데이터의 분산을 최대한 보존하면서 서로 직교하는 주성분을 탐색.
- ✓ 주성분들을 이용하여 기존 데이터를 축소 및 근사.

◆ 기존 변수와 차원 축소된 변수간의 정보의 손실을 최소화하는 관점으로도 이해해볼 수 있음.

- ✓ 주성분 분석의 결과 = 경험 위험 최소화 결과

2. 주성분 분석

경험 위험 최소화 관점

- ◆ $X \in \mathbb{R}^{n \times p} : p$ 차원 자료 n 개를 모은 행렬.
($p < n$ 이라 가정.)
- ◆ 표본 공분산 행렬을 다음과 같이 나타낼 수 있음.

$$S = \sum_{j=1}^p \lambda_j \mathbf{v}_j \mathbf{v}_j^T$$

- ✓ 여기서, λ_j, \mathbf{v}_j 는 각각 고유값과 고유 벡터를 의미.
- ✓ $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$ 를 가정.

2. 주성분 분석

경험 위험 최소화 관점

- ◆ $a_1, \dots, a_q : q$ 개의 임의의 직교 기저 ($q < p$ 를 가정)

$$\rightarrow a_j^T a_j = 1, j = 1, \dots, q$$

$$\rightarrow a_j^T a_k = 0, j \neq k$$

- ◆ $z_i : x_i$ 를 q 차원의 기저 공간으로 근사한 벡터

$$z_i = \mu + \sum_{j=1}^q [(x_i - \mu)^T a_j] a_j$$

2. 주성분 분석

경험 위험 최소화 관점

- ◆ 정리 : 위와 같은 식으로 나타내어지는 z_i 들 중에서 제곱 손실 함수의 합

$$\sum_{i=1}^n (x_i - z_i)^T (x_i - z_i)$$

를 최소로 하는 z_i 는 주성분 분석을 통해 얻은 근사 벡터이다.

- ◆ 즉,

$$z_i = \mu + \sum_{j=1}^q [(x_i - \mu)^T v_j] v_j$$

일 때 제곱 손실 함수의 합이 최소가 된다.

2. 주성분 분석

주성분 분석 예제

```
from keras.datasets import mnist
from sklearn.decomposition import PCA
```

MNIST 모듈 및 PCA 함수 불러오기

```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
X_train = X_train.reshape(X_train.shape[0], 28*28)
X_test = X_test.reshape(X_test.shape[0], 28*28)
```

MNIST 자료를 불러오고 형태 변환
(28,28)->(28*28)

```
pca = PCA(n_components = 10)
pca.fit(X_train)
```

훈련 자료를 이용해 주성분 찾기
(n_components는 찾을 주성분의 개수)

```
PCA(copy=True, iterated_power='auto', n_components=10, random_state=None,
     svd_solver='auto', tol=0.0, whiten=False)
```


2. 주성분 분석

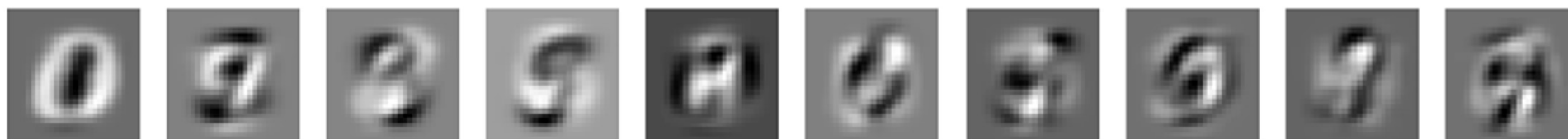
주성분 분석 예제

10개의 주성분을 그림으로 보기

```
import matplotlib.pyplot as plt

plt.figure(figsize=(15,3))

for i in range(10):
    plt.subplot(1, 10, i+1)
    plt.axis('off')
    plt.imshow(pca.components_[i].reshape(28,28), cmap='gray')
plt.show()
```



3. 오토 인코더

3. 오토 인코더

Auto Encoder

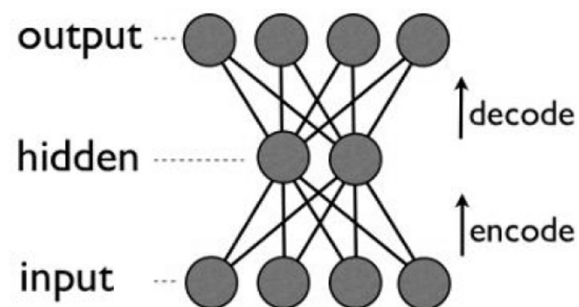
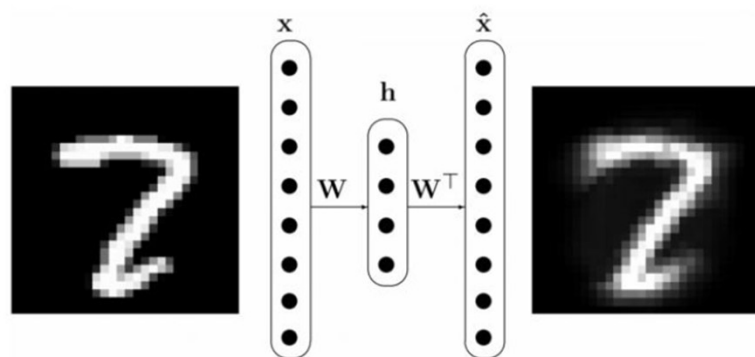
- ◆ 관측된 고차원 자료의 차원 축소를 목표로 하는 비지도 학습 방법론.
- ◆ 통계학의 주성분 분석과 유사하나, 관측치와 주성분 사이의 관계가 비선형임.
- ◆ 나아가 주성분들이 계층적 구조를 가짐.



3. 오토 인코더

Auto Encoder

- ◆ 출력값이 입력값을 근사하도록 신경망 함수를 학습.
- ◆ 인코더 (Encoder) 와 디코더 (Decoder) 로 나뉘어져 있음.
 - 인코더 : 입력 데이터에 대한 핵심 특징을 추출하는 함수.
 - 디코더 : 추출된 값을 이용하여 원본 데이터를 재구성해주는 함수.



3. 오토 인코더

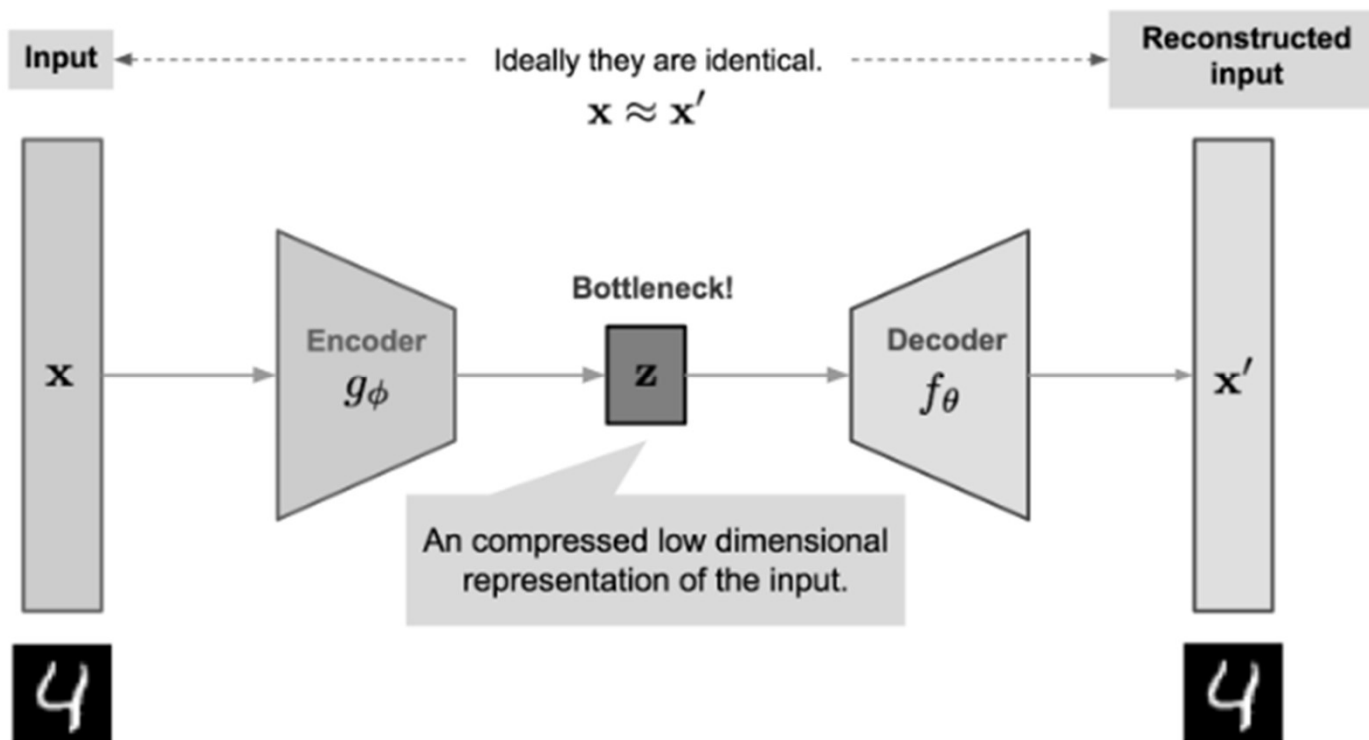
Auto Encoder

- ◆ 인코딩 과정에서 입력된 데이터의 핵심 정보만 hidden layer 에서 학습하고 나머지 정보는 손실시킴.
- ◆ 따라서, 디코딩 과정에서 hidden layer 의 출력값을 뽑았을 때 완벽한 복사가 아닌 입력값의 근사치가 됨.
- ◆ 출력값이 입력값과 최대한 비슷해지도록 학습.

3. 오토 인코더

Auto Encoder

◆ 오토 인코더의 학습 과정



3. 오토 인코더

Auto Encoder 수식 표현

◆ $x \in \mathbb{R}^p$: 입력값

◆ 인코더는 입력값 x 를 비선형 함수를 이용하여 q 차원 벡터 h 로 변환:

$$h = \sigma(Wx + b),$$

여기서 $W \in \mathbb{R}^{q \times p}, b \in \mathbb{R}^q$ 는 변환에 필요한 모두 행렬과 벡터,

$\sigma(\cdot)$ 는 주어진 활성화 함수를 뜻함.

→ 주로, $q \leq p$ 를 가정.

3. 오토 인코더

Auto Encoder 수식 표현

- ◆ 디코더는 인코딩된 벡터 h 를 다시 p 차원 벡터로 복원 :

$$\tilde{x} = \sigma(\tilde{W}h(x) + \tilde{b}),$$

여기서 $\tilde{W} \in \mathbb{R}^{p \times q}, \tilde{b} \in \mathbb{R}^p$ 는 변환에 필요한 모두 행렬과 벡터.

3. 오토 인코더

Auto Encoder의 학습

- ◆ 관찰된 자료 : x_1, \dots, x_n
- ◆ 추정해야 할 모수 : $\theta = (W, b, \tilde{W}, \tilde{b})$
- ◆ 입력값과 출력값이 최대한 비슷할 수 있도록 모수를 학습.

3. 오토 인코더

Auto Encoder의 학습

- ◆ 주로, 제곱합 함수를 사용.

$$l(\theta) = \sum_{i=1}^n \|\tilde{x}_i - x_i\|_2^2$$

- ◆ 제곱합 함수를 최소화 하는 모수를 추정.
- ◆ 역전파 알고리즘 사용.

3. 오토 인코더

Stacked Auto Encoder (SAE)

- ◆ 오토 인코더를 이용하여 만든 딥러닝 모형 중 하나.
- ◆ 오토 인코더와 같은 목적을 가지고 있는 모형이지만
오토 인코더와는 달리 여러 개의 중간층을 가지고 있음.
 - 심층 신경망 함수를 사용.

3. 오토 인코더

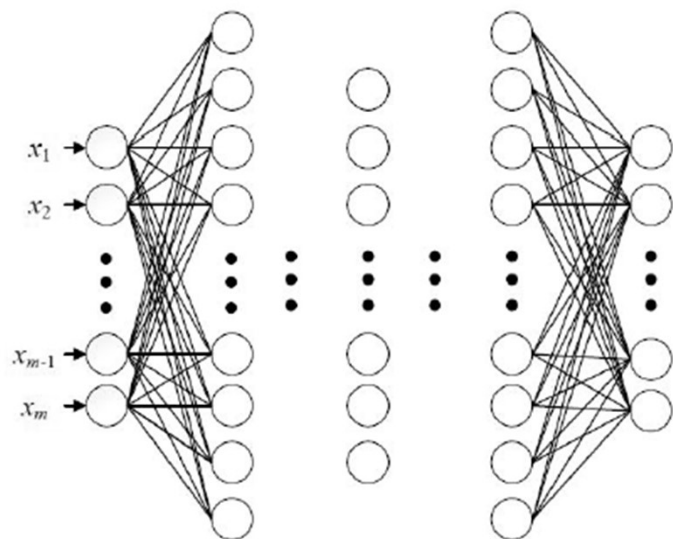
Stacked Auto Encoder (SAE)

- ◆ $2l-1$ 개의 중간층을 가지는 SAE 모형 :

$$h^{(1)} = \sigma(W^{(1)}x + b^{(1)})$$

$$h^{(k)} = \sigma(W^{(k)}h^{(k-1)} + b^{(k)}), 2 \leq k \leq 2l - 1$$

$$\tilde{x} = \sigma(W^{(2l)}h^{(2l-1)} + b^{(2l)})$$



3. 오토 인코더

SAE 의 학습

- ◆ 관찰된 자료 : x_1, \dots, x_n
- ◆ 추정해야 할 모수 : $\theta = (W^{(l)}, b^{(l)}, l = 1, \dots, 2l)$
- ◆ 입력값과 출력값이 최대한 비슷해지도록 모수를 학습.

$$l(\theta) = \sum_{i=1}^n \|\tilde{x}_i - x_i\|_2^2$$

- ◆ 역전파 알고리즘 사용.

3. 오토 인코더

오토 인코더 예제

```
from keras.datasets import mnist
```

```
from keras.layers import Input, Dense  
from keras.models import Sequential
```

```
(X_train, y_train), (X_test, y_test) = mnist.load_data()  
X_train = X_train.reshape(X_train.shape[0], 28*28)  
X_test = X_test.reshape(X_test.shape[0], 28*28)  
X_train = X_train.astype('float32')/255  
X_test = X_test.astype('float32')/255
```

```
model = Sequential()
```

```
model.add(Dense(10, activation='relu')) # encoder
```

```
model.add(Dense(784, activation='sigmoid')) # decoder
```

```
model.compile(loss='mean_squared_error', optimizer='adadelta')
```

MNIST 모듈 및 신경망 구성 요소 불러오기

MNIST 자료를 불러오고 형태 변환 (28,28)->(28*28),
각 픽셀 값을 [0,1]로 정규화

모형 생성

- * 인코더, 디코더 모두 하나의 fully connected layer로 구성
- * 디코더에 sigmoid 활성화함수를 적용해 결과 역시 [0,1]의 값을 취하도록 함
- * 최적화 방법은 adadelta, 손실함수는 제곱오차를 이용

3. 오토 인코더

오토 인코더 예제

모형 학습

```
hist = model.fit(X_train, X_train, batch_size = 200, epochs = 50, verbose = 1, validation_split = .05)
```

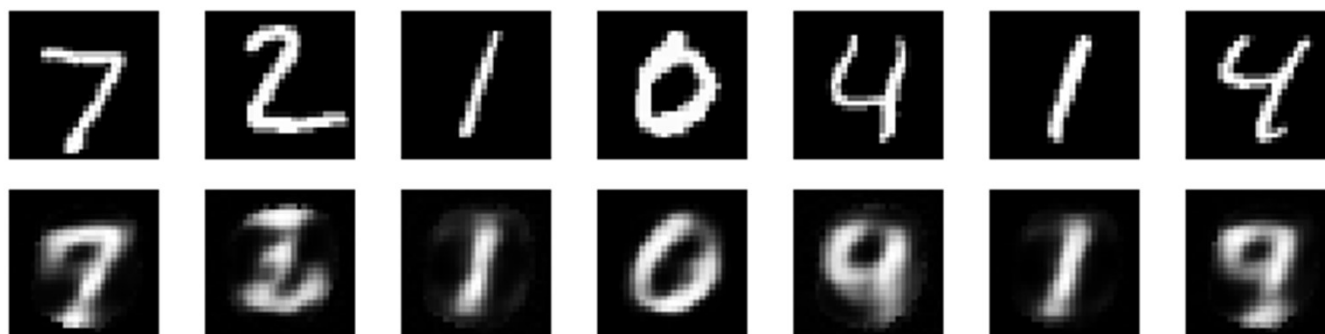
```
X_test_pred = model.predict(X_test)
```

모형에 테스트 자료 넣은 결과 얻기

```
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(12,3))
for i in range(7):
    plt.subplot(2, 7, i+1)
    plt.axis('off')
    plt.imshow(X_test[i].reshape(28,28), cmap='gray')
    plt.subplot(2, 7, i+8)
    plt.axis('off')
    plt.imshow(X_test_pred[i].reshape(28,28), cmap='gray')
plt.show()
```

테스트 자료 중 첫 7개의 원래 그림 및
복원된 그림 보여주기



학습정리

- ✓ 비지도 학습은 데이터의 특징을 추출하는 방법론을 총칭한다.
- ✓ 주성분 분석은 데이터의 분산을 최대한 보존하면서 직교하는 주성분을 찾는 방법이다.
- ✓ 주성분 분석은 경험 위험 최소화의 관점으로도 해석할 수 있다.
- ✓ 오토 인코더는 신경망 함수를 통해 데이터의 핵심 정보를 추출하는 방법론 (차원축소)이며, 역전파 알고리즘을 통해 모수를 추정한다.

딥러닝의 통계적이해

다음시간안내

10강. 오토인코더와 GAN(2)