머신러닝응용 제14강

Deep Learning 1.

첨단공학부 김동하교수

제14강 Deep Learning 1.

1	딥러닝의 개념에 대해 학습한다.
2	(심층)인공신경망모형에 대해 학습한다.
3	(심층)인공신경망모형을 학습할 때 사용하는 목적함수와 최적화 알고리즘에 대해 학습한다.

핵심 단어

- (심층)인공신경망
- 활성함수
- 역전파 알고리즘
- Keras 모듈

14강. Deep Learning 1.

01. 딥러닝이란

1) 기계 학습

- 컴퓨터에 인공적인 학습 가능한 지능을 부여하는 것을 연구하는 분야.

  - 아서 사무엘 : *“기계학습이란 기계가 일일이 코드로 명시하지 않은 동작은 데이터로부터 학습하여 실행할 수 있도록 하는 알고리즘을 개발하는 연구 분야.”*

2) 인공 신경망

- 인공 신경망(Artificial Neural Network)
  - 기계학습 분야에서 연구되고 있는 학습 모델 중 하나.
  - 주로 패턴인식에 쓰이는 기술로, 인간의 뇌의 뉴런과 시냅스의 연결을 프로그램으로 재현하는 것.

- 딥러닝(Deep Learning)은 심층 인공 신경망 모형을 기초로 해서 발전하였음.
  - 심층 모형을 학습한다. -> 딥러닝

3) 딥러닝의 응용분야

- 이미지 분석
  - 이미지 인식, 압축, 복원, 생성 등
  - AlphaGo (Silver et al., 2016 & 2017)
  - DeepDream (그림을 그리는 인공지능), Artistic style transfer (Gatys et al., 2015)
  - 무인 자동차

3) 딥러닝의 응용분야

◆ 이미지 분석

출처:
https://www.pinterest.co.kr/pin/421227371375202065/

출처:
http://bethgelab.org/research/machine_learning/style_transfer/

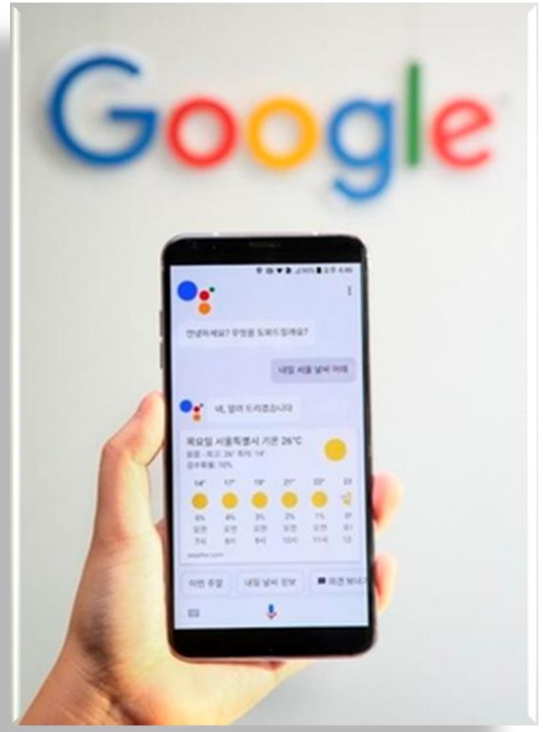
3) 딥러닝의 응용분야

- 텍스트 분석
  - 파파고, 구글 번역기
  - 챗봇

출처: https://papago.naver.com/

3) 딥러닝의 응용분야

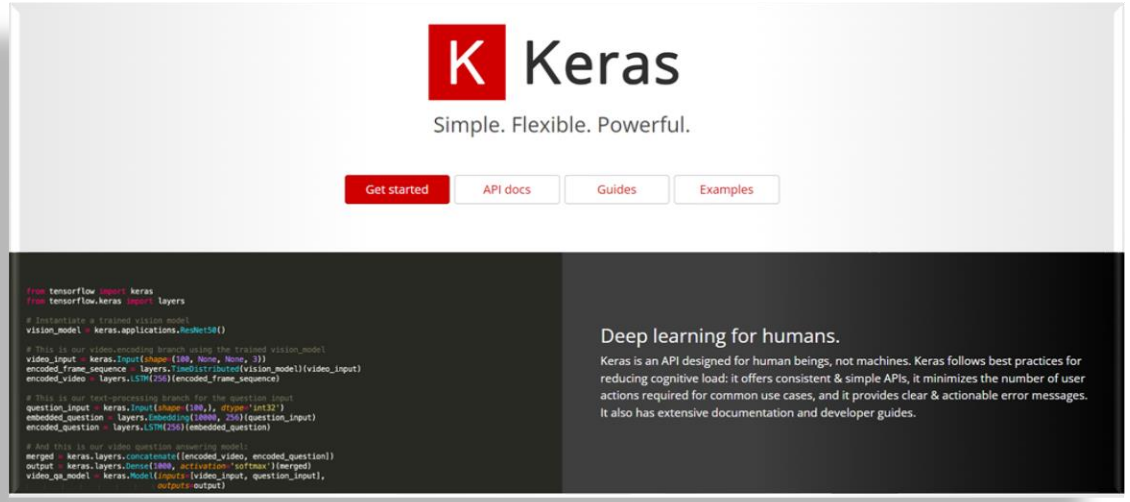
- 오디오 분석
  - 음성 인식, 복원, 생성, 분해 등
  - 인공지능 스피커, 챗봇, 비서



출처:
http://it.chosun.com/site/data/html_dir/2017/09/21/2017092185006.html

4) 딥러닝의 구현

- Python 기반으로 다양한 모듈을 제공하고 있음.
  - Tensorflow (Google)
  - Keras (Google) - tensorflow, CNTK 기반.
    - https://keras.io/
  - CNTK (Microsoft)
  - Pytorch (Facebook)



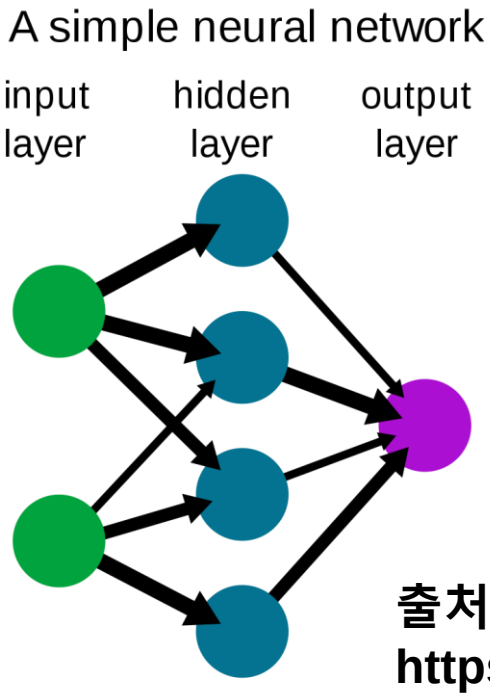
출처: https://keras.io/

14강. Deep Learning 1.

02. 인공신경망

1) 인공 신경망 모형

- 인공 신경망 모형 (Artificial neural networks)
  - 생물학의 뇌 구조 (신경망) 를 모방하여 만든 모형.
  - 인간의 뇌가 문제를 해결하는 방식과 유사하게 구현.



출처:
https://en.wikipedia.org/wiki/Neural_network

2) 인공 신경망 구조

- 1개의 중간층 혹은 은닉층 (hidden layer)을 갖는 (fully connected) neural networks $\mathbb{R}^p \rightarrow \mathbb{R}^k$모형을 생각하자.

$$z_m^{(0)} = b_m^{(0)} + x^T w_m^{(0)}, \qquad h_m = \sigma\left(z_m^{(0)}\right), \qquad m = 1, \ldots, M$$

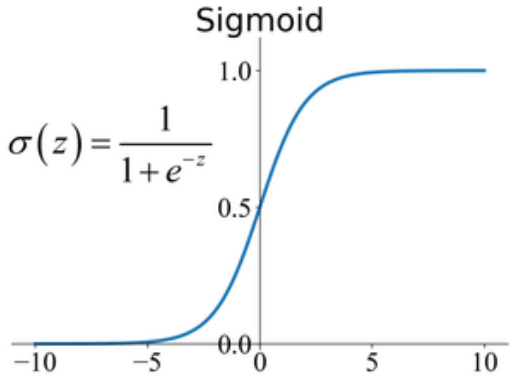
$$z_k^{(1)} = b_k^{(1)} + h^T w_k^{(1)}, \qquad f_k(x) = g_k\left(z^{(1)}\right), \qquad k = 1, \ldots, K$$

2) 인공 신경망 구조

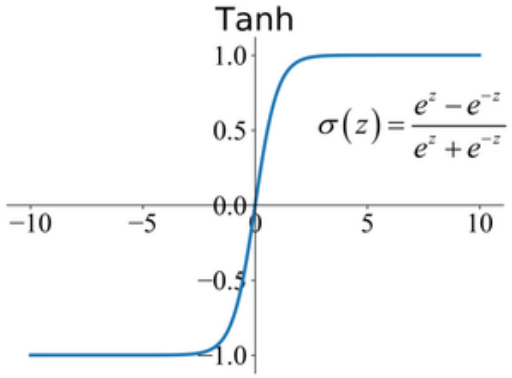
- $\sigma(\cdot)$: 활성함수 (activation function)
  - 비선형성을 갖도록 하는 함수
  - Sigmoid: $\sigma(v) = 1/(1 + \exp(-v))$
  - Tanh: $\sigma(v) = (e^{v} - e^{-v})/(e^{v} + e^{-v})$
- $g_k(\cdot)$: 출력 함수 (output function)
  - 회귀 분석 : $g_k(t) = t$
  - 분류 문제 : softmax 함수 사용!
    - $g_k(t) = e^{t_k}/(\sum_{l=1}^{K} e^{t_l})$

2) 인공 신경망 구조

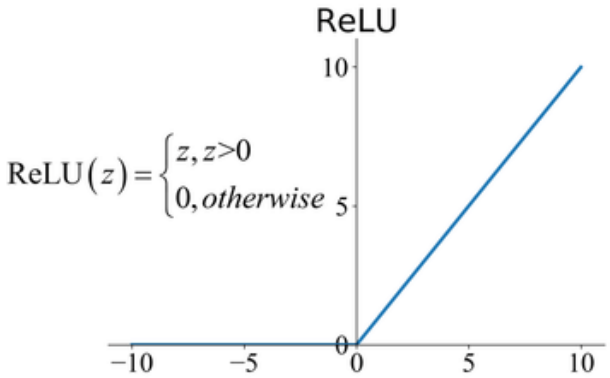
◆ 활성 함수의 형태



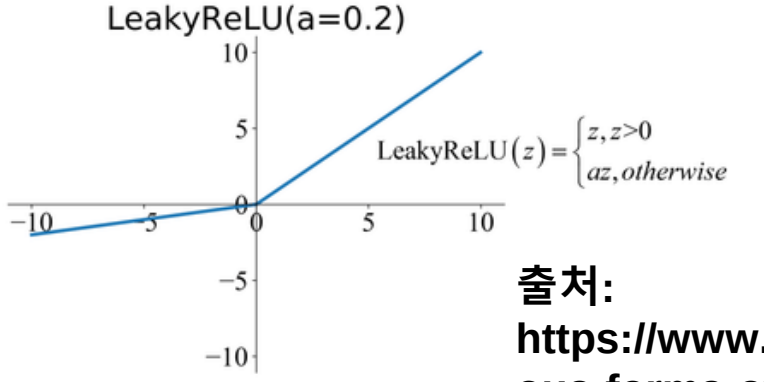
(a)



(b)



(c)



(d)

출처:
https://www.researchgate.net/figure/Various-forms-of-non-linear-activation-functions-Figure-adopted-from-Caffe-Tutorial_fig3_315667264

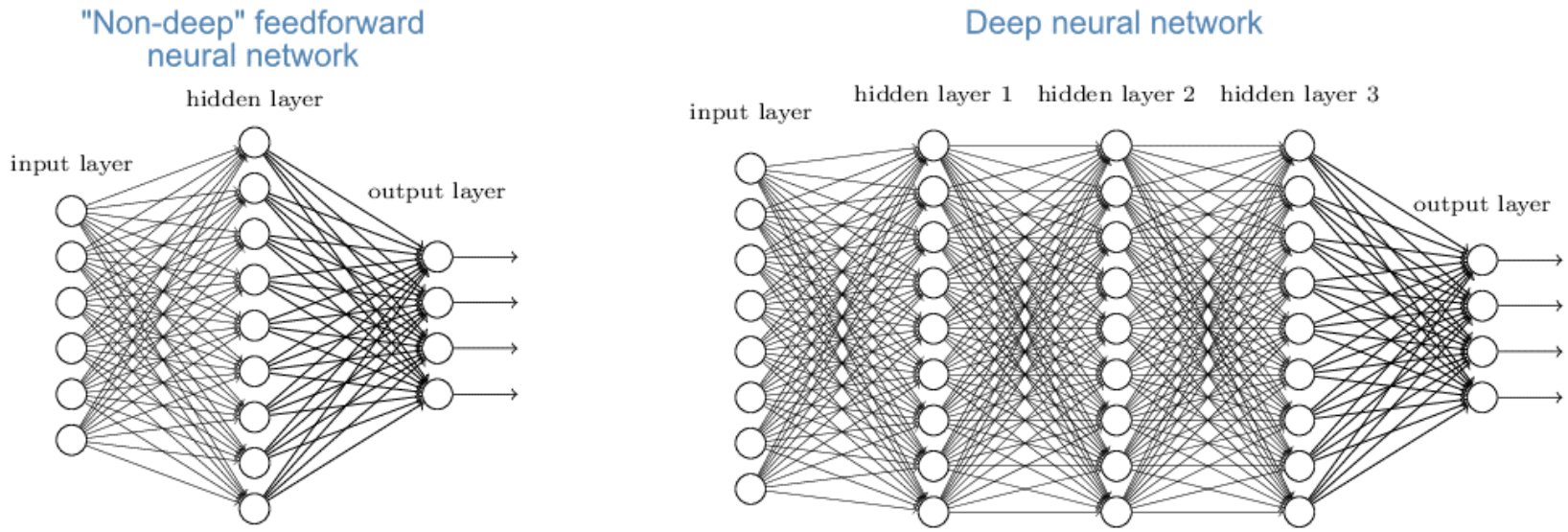
14강. Deep Learning 1.

03. 심층인공신경망

1) 심층인공 신경망 모형

심층 인공 신경망 모형 (Deep neural networks)

- 2개 이상의 중간층을 가지고 있는 신경망 모형



출처: https://www.i2tutorials.com/explain-deep-neural-network-and-shallow-neural-networks/

1) 심층인공 신경망 모형

- L개의 중간층 (hidden layer) 을 갖는 (fully connected) neural networks 모형을 생각하자.

- $h^{(l)} \in \mathbb{R}^{n_l}, l = 0, \ldots, (L+1)$: l 번째 중간층의 노드 벡터.
  - $h^{(0)} = x, h^{(L+1)} = f(x)$

1) 심층인공 신경망 모형

- 모형의 출력값은 다음과 같이 계산되어짐.

$$z^{(l+1)} = b^{(l)} + W^{(l)} h^{(l)}, \qquad l = 0, \dots, L$$

$$h^{(l+1)} = \sigma\left(z^{(l+1)}\right), \qquad l = 0, \dots, L$$

$$f(x) = h^{(L+1)} = g(z^{(L+1)})$$

- $\sigma(\cdot)$: 활성함수 (activation function), 비선형성을 갖도록 하는 함수.
  - 은닉층의 수가 많을 경우 Sigmoid보다는 ReLU가 많이 사용된다.

1) 심층인공 신경망 모형

- $\sigma(\cdot)$: 활성함수 (activation function), 비선형성을 갖도록 하는 함수.
  - 은닉층의 수가 많을 경우 Sigmoid보다는 ReLU가 많이 사용된다.
- $g(\cdot)$: 출력 함수 (output function)
  - 회귀 분석 : $g_k(t) = t$
  - 분류 문제 : softmax 함수 사용!
    - $g_k(t) = e^{t_k} / (\sum_{l=1}^{K} e^{t_l})$

2) DNN vs. SVM

공통점

- Hidden feature를 사용.
- Hidden space 위에서 linear decision boundary를 이용하여 분류.

2) DNN vs. SVM

차이점

- SVM : 정해진 feature mapping을 사용하고, linear decision boundary만 학습.
- NN : linear decision boundary 뿐만 아니라 feature mapping도 함께 학습.

14강. Deep Learning 1.

04. 인공신경망모형의 학습

1) 목적함수

- 학습해야 하는 모수: $\left(b^{(l)}, W^{(l)}\right), l = 0, \ldots, L$
  - $\theta = \{b^{(l)}, W^{(l)}\}_{l=0}^{L}$
- 훈련 데이터: $(x_i, y_i), i = 1, \ldots, n$
- 본 강의에서는 지도 학습만을 다룰 예정.

1) 목적함수

분류 문제의 경우 : Cross entropy function

$$L(\theta) = -\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \log f_k(x_i)$$

- $y_i \in \mathbb{R}^K$: one hot vector

회귀 문제의 경우: Squared loss function

$$L(\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K} (y_{ik} - f_k(x_i))^2$$

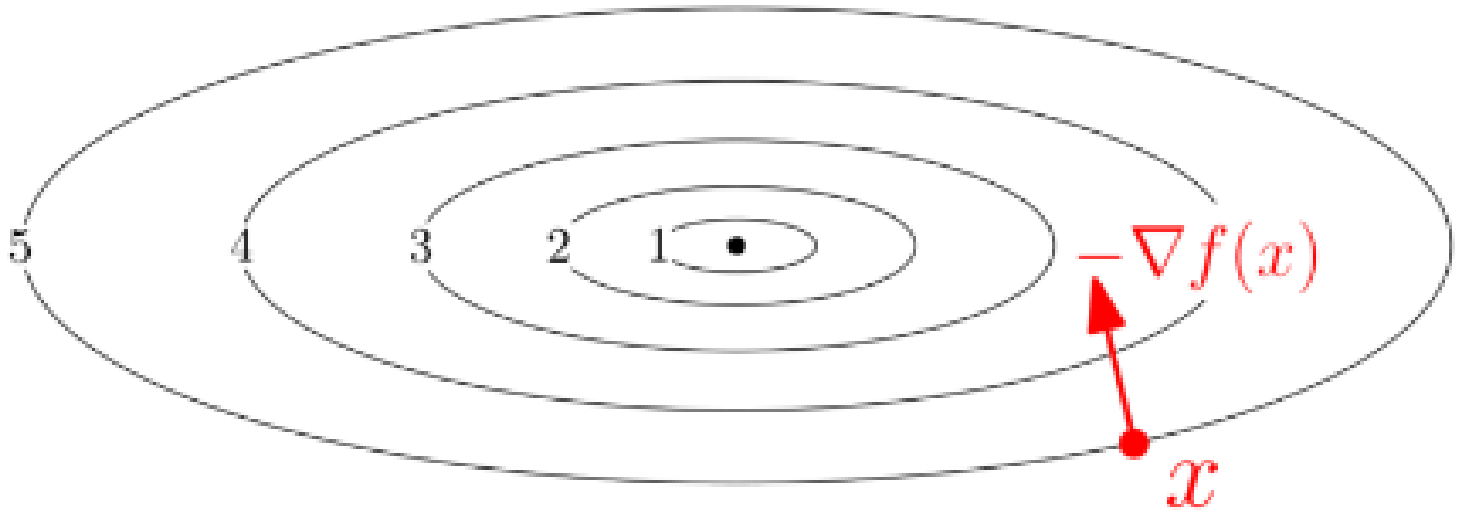
2) 목적함수의 최적화: 기울기 강하 알고리즘

- Gradient descent algorithm

- 특정 목적 함수 $L(\theta)$를 최소화하는 θ을 한 번에 찾기 힘든 경우에 사용하는 대표적인 반복 알고리즘.

2) 목적함수의 최적화: 기울기 강하 알고리즘

- $-\frac{\partial L(\theta)}{\partial \theta}|_{\theta=\theta_0}$은 $\theta = \theta_0$일 때 $L(\theta)$을 가장 빠르게 감소시키는 방향이라는 점에서 착안.



출처: https://francisbach.com/gradient-flows/

2) 목적함수의 최적화: 기울기 강하 알고리즘

알고리즘

- 초기값 설정: $\theta = \theta_0$
- 현재의 값을 $\theta^{(t)}$라 할 때, 학습률 (learning rate) ϵ_t을 이용하여

  $\theta^{(t+1)} \leftarrow \theta^{(t)} - \epsilon_t \frac{\partial L(\theta)}{\partial \theta}\Big|_{\theta=\theta^{(t)}}$으로 업데이트
- 수렴할 때까지의 ②의 과정을 반복.

3) 목적함수의 최적화: 역전파 알고리즘

- Back propagation 알고리즘
- Deep neural network의 모수들의 gradient를 구하는 알고리즘

3) 목적함수의 최적화: 역전파 알고리즘

- 인공신경망의 목적 함수 $L(\theta)$을 최소화하기 위해 기울기 강하 알고리즘을 사용하면, 인공 신경망 모형의 특수한 형태 때문에 $\frac{\partial L(\theta)}{\partial \theta^{(l+1)}}$ 의 계산에 필요한 값을 알고 있으면, $\frac{\partial L(\theta)}{\partial \theta^{(l)}}$가 자동적으로 계산됨.
  - $\theta^{(l)}$은 l 층에서의 모수

- 미분값이 위에서 아래로 계산되어짐.
  - Back propagation

14강. Deep Learning 1.

06. Python을 이용한 실습

1) 데이터 설명

- Fashion MNIST
  - 의류가 그려져 있는 이미지 데이터
  - 훈련 자료: 60,000개, 시험 자료: 10,000개
  - 각 이미지는 10가지의 의류 중 한가지가 그려져 있음.

- Fashion MNIST 자료를 이용하여 의류의 종류를 예측하는 심층인공신경망 모형을 학습해보자.
  - Keras 모듈 사용!

2) 환경설정

◆필요한 패키지 불러오기

```
from tensorflow.keras.datasets import fashion_mnist
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten

from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

3) 데이터 불러오기

◆ 데이터 불러오기

```
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
```

3) 데이터 불러오기

◆ 라벨 이름 저장

```
# label 이름을 저장
item = {
        0: 'T-shirt/top'
      , 1: 'Trouser'
      , 2: 'Pullover'
      , 3: 'Dress'
      , 4: 'Coat'
      , 5: 'Sandal'
      , 6: 'Shirt'
      , 7: 'Sneaker'
      , 8: 'Bag'
      , 9: 'Ankle boot'
}
```

3) 데이터 불러오기

◆ 데이터 확인하기

```
# train 이미지 확인
plt.figure(figsize=(8, 8))
for i in range(16):
    plt.subplot(4, 4, i+1)
    plt.suptitle('Train Images', fontsize=20)
    plt.title(item[y_train[i]])
    plt.imshow(x_train[i], cmap=plt.cm.gray)
    plt.axis("off")

plt.show()
```

3) 데이터 불러오기

◆ 데이터 확인하기

4) 데이터 전처리

- 입력값을 32 비트 형태로 변환
- 픽셀값을 0과 1 사이로 재조정

```
x_train = x_train.astype('float32')
x_train = x_train/255


x_test = x_test.astype('float32')
x_test = x_test/255
```

4) 데이터 전처리

◆분류값을 one hot vector 형태로 변환

```
y_onehot_train = to_categorical(y_train, num_classes=10)
y_onehot_test = to_categorical(y_test, num_classes=10)
```

5) 심층인공신경망 생성

Hyper parameter 설정

```
OUTPUT_SHAPE = 10
BATCH_SIZE = 128
EPOCHS = 10
VERBOSE = 1
```

5) 심층인공신경망 생성

모형 생성

```
model = Sequential([
  Flatten(),
  Dense(128, activation='relu'),
  Dense(64, activation='relu'),
  Dense(10, activation='softmax')
])
```

5) 심층인공신경망 생성

◆ 목적함수 및 최적화 알고리즘 설정

```
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

6) 심층인공신경망 학습

인공신경망 모형 학습

```
history = model.fit(
    x_train, y_onehot_train,
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    verbose=VERBOSE,
    validation_split=0.3
)
```

```
Epoch 1/10
329/329 [==============================] - 1s 4ms/step - loss: 0.5964 -
accuracy: 0.7938 - val_loss: 0.4447 - val_accuracy: 0.8439
Epoch 2/10
329/329 [==============================] - 1s 3ms/step - loss: 0.4124 -
accuracy: 0.8538 - val_loss: 0.4134 - val_accuracy: 0.8531
Epoch 3/10
329/329 [==============================] - 1s 3ms/step - loss: 0.3728 -
accuracy: 0.8658 - val_loss: 0.3737 - val_accuracy: 0.8656
```

7) 학습된 모형 평가

◆ 시험 자료의 목적함수 값과 정분류율 확인

```
model.evaluate(x_test, y_onehot_test)    # loss와 accuracy 반환
```

```
313/313 [==============================] - 0s 1ms/step - loss: 0.3568 -
accuracy: 0.8759
[0.3567776679992676, 0.8758999705314636]
```

7) 학습된 모형 평가

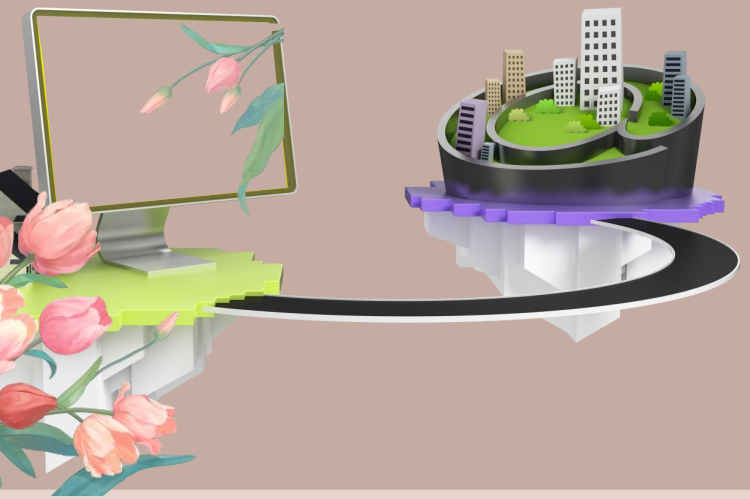
◆ 오차행렬 확인하기

```
y_pred_enc = model.predict(x_test)
y_pred = [np.argmax(i) for i in y_pred_enc]

matrix = confusion_matrix(y_test, y_pred)
df = pd.DataFrame(matrix)
df.columns = item.values()
df.index = item.values()

df
```

7) 학습된 모형 평가

◆ 오차행렬 확인하기

	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
T-shirt/top	835	10	20	30	2	3	95	0	5	0
Trouser	0	971	1	23	3	0	2	0	0	0
Pullover	11	0	797	8	100	1	81	0	2	0
Dress	21	10	18	872	42	1	33	0	3	0
Coat	0	1	104	22	798	1	74	0	0	0
Sandal	0	0	0	1	0	966	0	21	1	11
Shirt	143	3	92	26	58	0	669	0	9	0
Sneaker	0	0	0	0	0	24	0	957	1	18
Bag	7	1	9	4	4	5	10	4	956	0
Ankle boot	0	0	0	0	0	15	1	46	0	938



다음시간안내

제15강

Deep learning 2.