

TensorFlow-Keras 딥러닝

CNN : Fashion-Mnist

- TensorFlow의 합성곱 신경망(convolutional neural network, CNN)을 활용한 Deep Learning
- 이미지 데이터 세트를 로드하고 전처리 하여 딥러닝
- 합성곱 레이어층 만들기
- 모델 구조 출력
- 모델 컴파일
- 모델 훈련·학습
- 모델 평가

```
[1] 1 from tensorflow import keras
```

```
[2] 1 fashion_mnist=keras.datasets.fashion_mnist
```

```
[3] 1 (X_train0, y_train0), (X_test0, y_test0) = fashion_mnist.load_data()
```

```
1 model = keras.models.Sequential([
2     keras.layers.Flatten(input_shape=(28,28)),
3     keras.layers.Dense(512, activation='relu'),
4     keras.layers.Dense(10, activation="softmax")
5 ])
6 #모델 실행
```

```
[5] 1 train_images, test_images = X_train0 / 255.0, X_test0 / 255.0
    2 # 스케일링
```

```
[6] 1 class_name=['t-shirts/top', 'trouser', 'pullover', 'dress', 'coat', 'sandal', 'shirt', 'sneaker', 'bag', 'ankle boot']
```

```
[7] 1 import matplotlib.pyplot as plt
```

```
[8] 1 plt.figure(figsize=(10,10))
2   for i in range(25):
3       plt.subplot(5,5, i+1)
4       plt.xticks([])
5       plt.yticks([])
6       plt.grid(False)
7       plt.imshow(train_images[i], cmap=plt.cm.binary)
8       plt.xlabel(class_name[y_train0[i]])
9   plt.show()
```



```
[9] 1 model.compile(optimizer='adam',
2               loss='sparse_categorical_crossentropy',
3               metrics=['accuracy'],
4               )
```

```
1 model.fit(train_images, y_train0, epochs=10)
2 #5번 학습
```

```
Epoch 1/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.4727 - accuracy: 0.8304
Epoch 2/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.3570 - accuracy: 0.8694
Epoch 3/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.3218 - accuracy: 0.8818
Epoch 4/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.2993 - accuracy: 0.8893
Epoch 5/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.2792 - accuracy: 0.8961
Epoch 6/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.2643 - accuracy: 0.9018
Epoch 7/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.2527 - accuracy: 0.9058
Epoch 8/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.2414 - accuracy: 0.9092
Epoch 9/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.2316 - accuracy: 0.9133
Epoch 10/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.2213 - accuracy: 0.9162
<tensorflow.python.keras.callbacks.History at 0x7fec43bb74d0>
```

```
[11] 1 model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 784)	0
dense (Dense)	(None, 512)	401920
dense_1 (Dense)	(None, 10)	5130
Total params: 407,050		
Trainable params: 407,050		
Non-trainable params: 0		

+ 코드 + 텍스트

```
total params: 407,050
[11] Trainable params: 407,050
Non-trainable params: 0
```

```
1 loss, accuracy = model.evaluate(test_images, y_test0)
```

```
313/313 [=====] - 1s 2ms/step - loss: 0.3288 - accuracy: 0.8892
```

```
[13] 1 import numpy as np
```

```
[14] 1 predictions=model.predict(test_images)
```

```
1 predictions[20]
```

```
array([6.8089319e-03, 5.9788485e-06, 9.9217165e-01, 1.0066461e-06,
        6.1119534e-04, 3.8438115e-09, 3.9289283e-04, 3.8214625e-06,
        1.7889206e-06, 2.7068063e-06], dtype=float32)
```

```
[17] 1 np.argmax(predictions[20]) # array 내에서 최대값
```

```
2
```

```
[19] 1 plt.figure()
2 plt.imshow(test_images[20])
3 plt.colorbar()
4 plt.grid(False)
5 plt.show()
6
```

