04

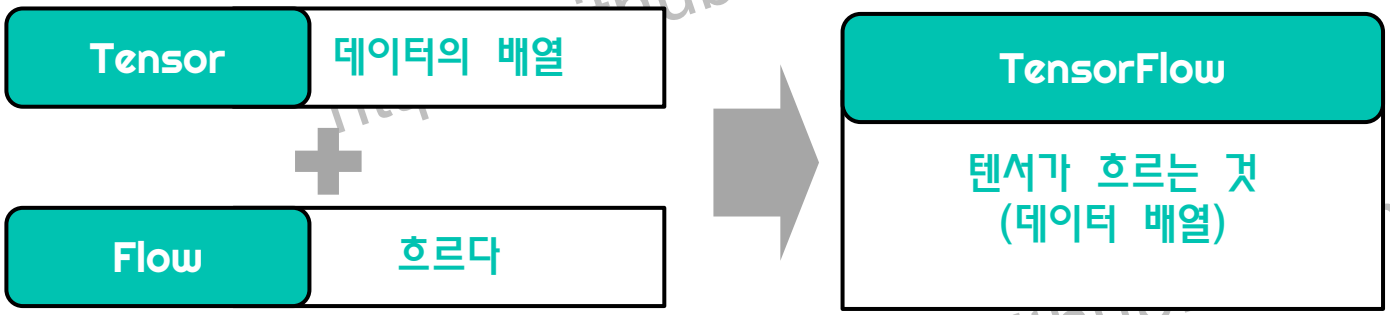
인공지능 개념

3. 텐서플로우

텐서플로우 기본개념

기본개념

- 텐서로 표현되는 데이터를 계산하고 실행하는 기능을 지원하는 프레임워크



텐서플로우 프로그램 기본구조

프로그램 기본구조

```
import tensorflow as tf                                                    ①

mnist = tf.keras.datasets.mnist                                            ②
(x_train, y_train), (x_test, y_test) = mnist.load_data()
print("* shape:", x_train.shape, y_train.shape)

model = tf.keras.models.Sequential([                                       ③
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimazer='adam',                                            ④
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy']
    )

hist= model.fit(x_train, y_train, epochs=5)                                ⑤

model.evaluate(x_test, y_test, verbose=2)                                  ⑥
```

텐서플로우 학습데이터 가져오기

학습데이터 가져오기

```
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
print("* shape train:", x_train.shape, y_train.shape)
print("* shape test:", x_test.shape, y_test.shape)
print("* test data:₩n", x_test)
```

```
* shape train: (60000, 28, 28) (60000,)
* shape test: (10000, 28, 28) (10000,)
* test data:
 [[[0 0 0 ... 0 0 0]
```

mnist 데이터

```
  ...
  [0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]]]
```

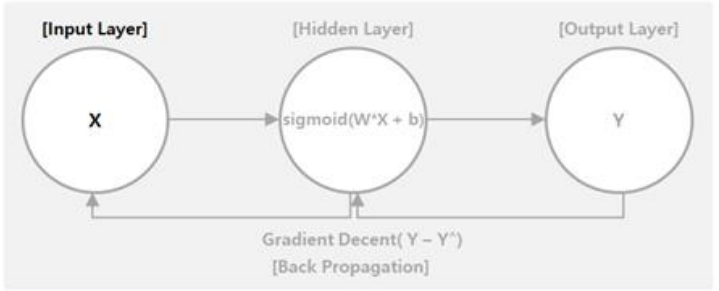
28

```
 [[0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]
  ...
  [0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]]
```

28

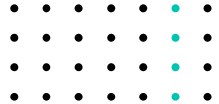
60000

mnist 이미지



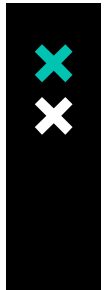
텐서플로우

훈련데이터와 검증데이터



훈련데이터와 검증데이터

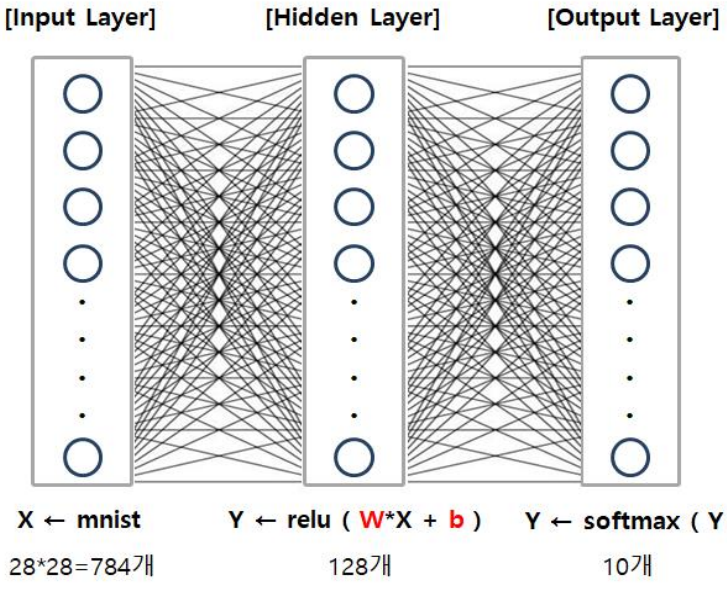
- 모델을 학습할 때 사용된 데이터를 가지고 모델을 검증하면 결과가 왜곡될 수 있다.
- 학습데이터와 검증데이터를 별도로 분리
- 전체 데이터 중 보통 70~80% 정도를 학습 데이터로 사용, 나머지를 검증 데이터로 사용



텐서플로우 인공신경망 구성

인공신경망 구성

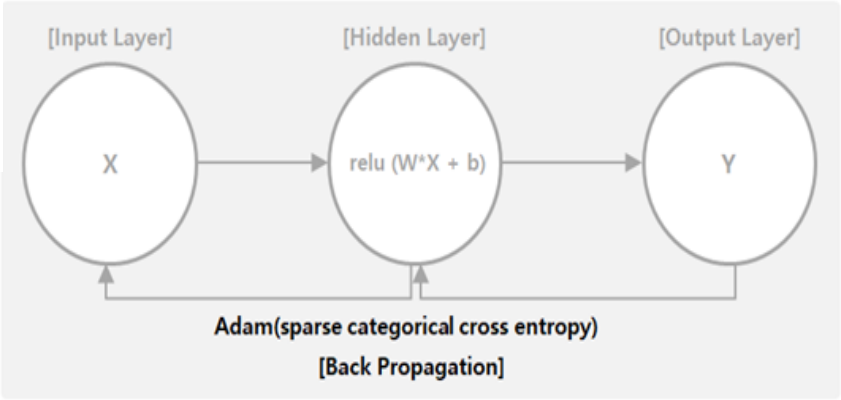
```
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(10, activation='softmax')
])
```



텐서플로우 학습환경 설정

학습환경 설정

```
model.compile(optimazer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy']
    )
```



½ (Y – Y^)²	sparse categorical cross entropy
Gradient Decent	Adam

텐서플로우 모델 학습

모델 학습

```
hist= model.fit(x_train, y_train, epochs=5)
```

```
Train on 60000 samples
Epoch 1/5
60000/60000 [==============================] - 8s 133us/sample - loss: 0.1850 - accuracy: 0.9839
Epoch 2/5
60000/60000 [==============================] - 7s 121us/sample - loss: 0.1829 - accuracy: 0.9844
Epoch 3/5
60000/60000 [==============================] - 8s 127us/sample - loss: 0.2034 - accuracy: 0.9845
Epoch 4/5
60000/60000 [==============================] - 8s 133us/sample - loss: 0.1866 - accuracy: 0.9850
Epoch 5/5
60000/60000 [==============================] - 7s 119us/sample - loss: 0.1925 - accuracy: 0.9850
```

텐서플로우 모델 검증

모델 검증

```
model.evaluate(x_test, y_test, verbose=2)
```

```
10000/1 - 1s - loss: 0.5757 - accuracy: 0.9646
[1.1514647204219093, 0.9646]
```