

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist #사람과 친함, 쉬움
```

```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

```
x_train.shape
```

```
↳ (60000, 28, 28)
```

```
x_test.shape
```

```
↳ (10000, 28, 28)
```

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)), #flatten 평평하게 => 한줄로 만들기 총 784개의 칸 #
    tf.keras.layers.Dense(256, activation=tf.nn.relu),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax) # 확률 그릇 10칸 #dense 다 연결함 ex)거미줄 등
])
# softmax 0~1 값 출력
# 값들은 다더하면 1
# 확률을 표현
# softmax(x)=exp(xi)/시그마 exp(xi) # 시험
model.compile(optimizer='adam',
               loss='sparse_categorical_crossentropy',
               metrics=['accuracy']) # adam, momentum # 공굴리기=모멘텀 아담=절반의 모멘텀
# 공굴리기 가장 낮은 곳을 찾아감 # 로컬 미니엄_정답'같이' 보이는 값, 글로벌 미니엄_정답 # 아담을 쓰지
```

거미줄 문제

1. 형상 무시: 한줄로 만듦
2. 너무많은 거미줄

convolution 입체 필터랑 겹치고 더함

```
model.summary()
```

```
↳
```

```
model.fit(x_train, y_train, epochs=5) # 모델을 돌림
```

```
↳
```

```
model.evaluate(x_test, y_test)
```



## ▼ Training accuracy

```
model.evaluate(x_train, y_train)
```

