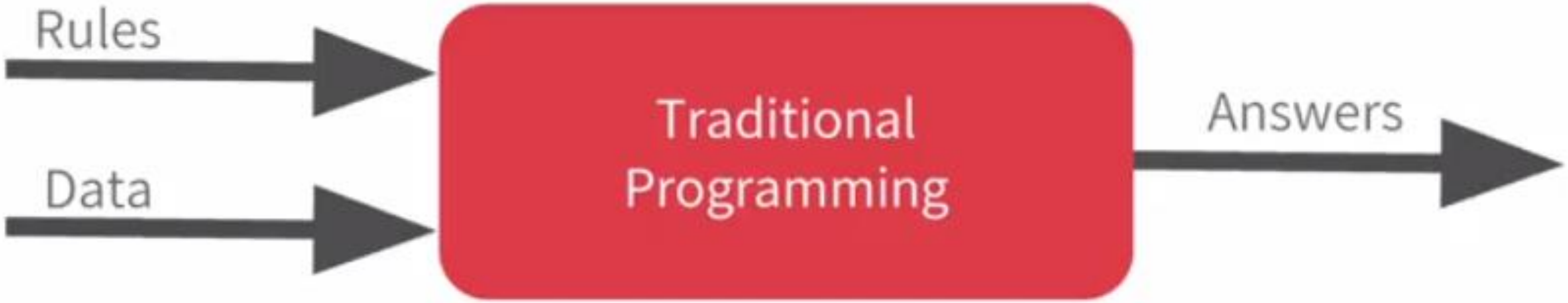
Week1. A New Programming Paradigm

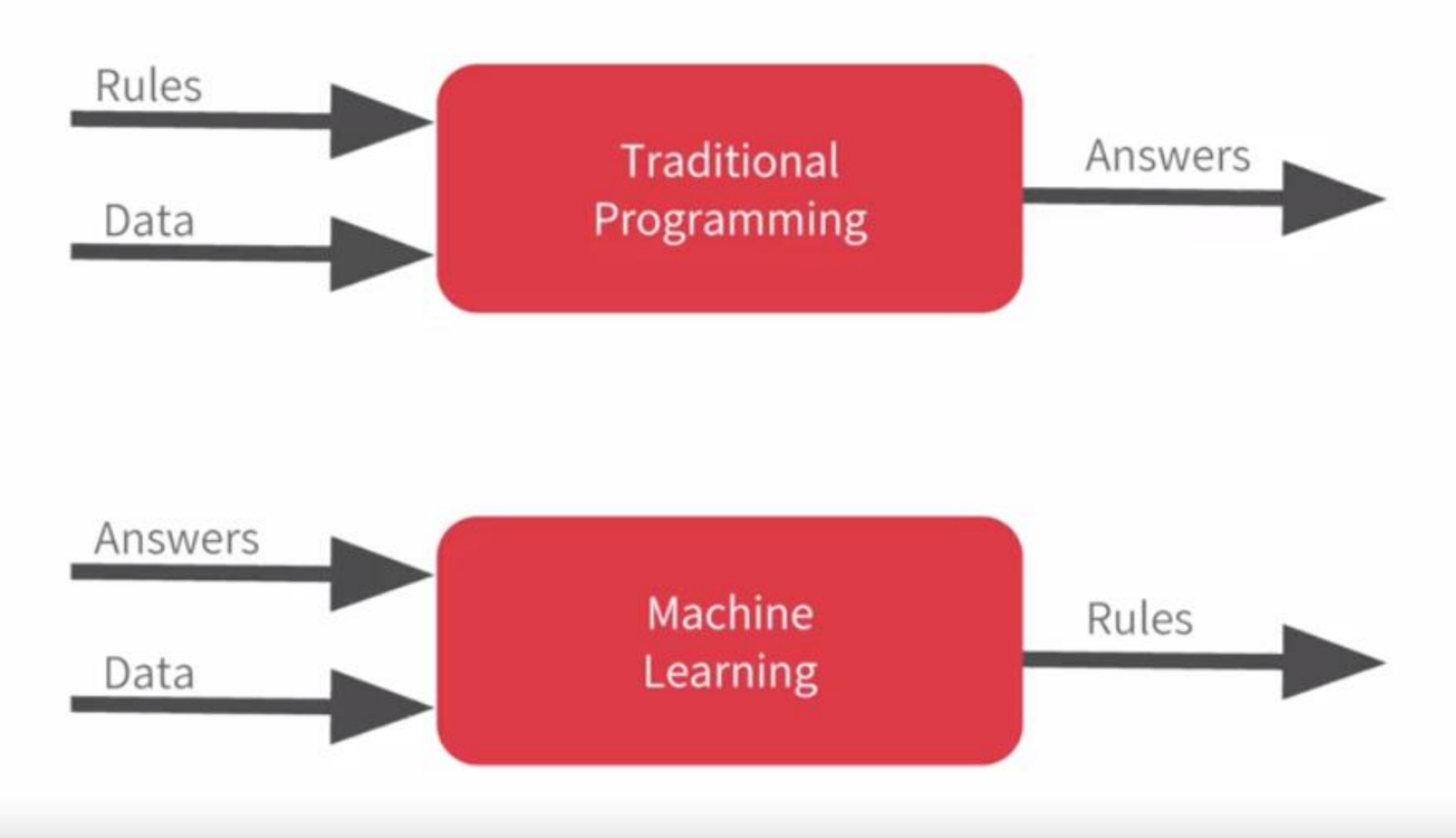


Rules

Data

Traditional Programming

Answers



Rules

Data

Traditional Programming

Answers

Answers

Data

Machine Learning

Rules

Week2. Introduction to Computer Vision

Object Recognition

Human

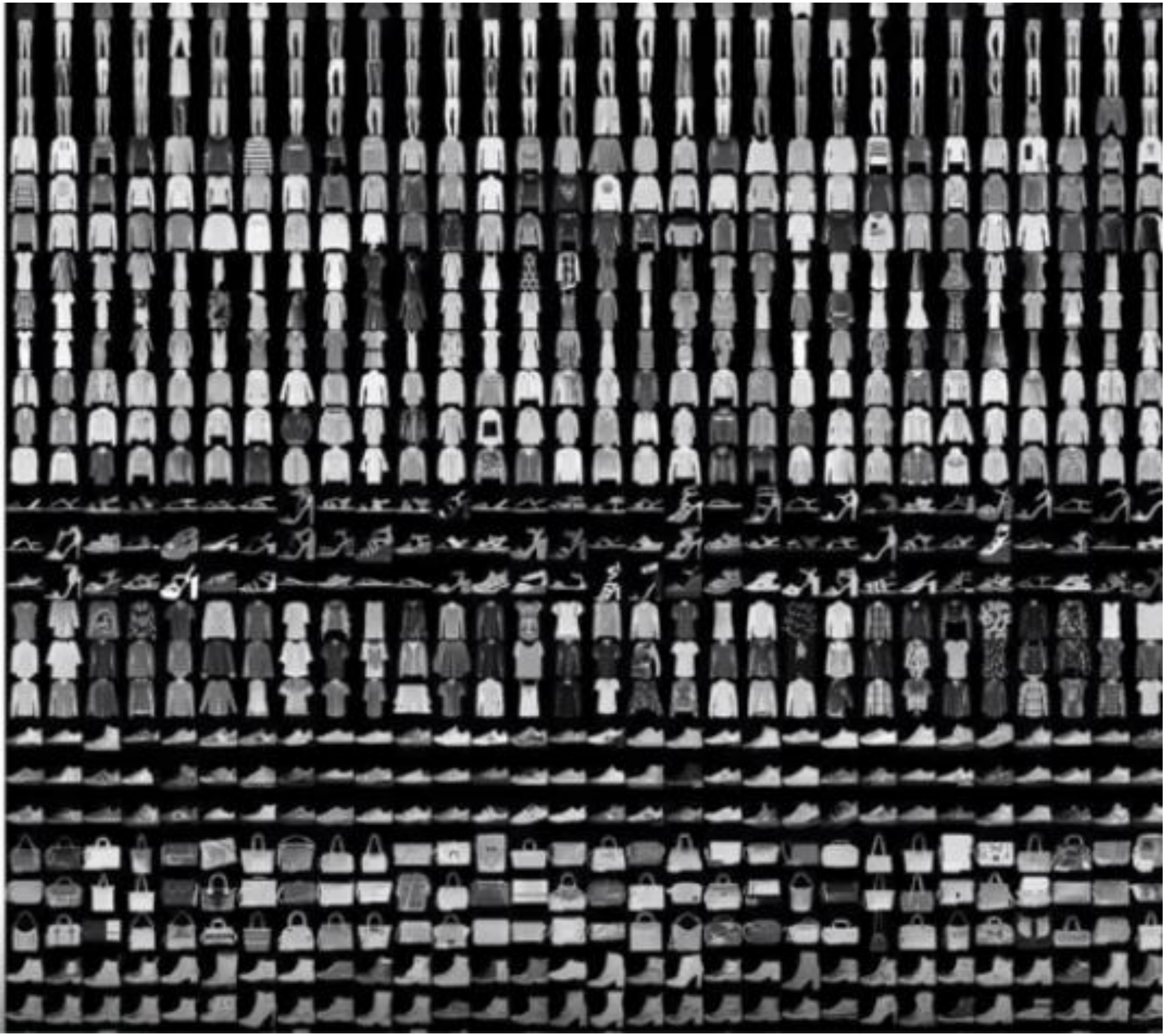
- 사진을 보고 어느 것이 옷 사진이고, 어느 것이 신발사진인지 쉽게 구분할 수 있다.
- 사람이 입고 있는 옷에 대해서도 어떤 색의 옷인지 어떤 종류의 옷인지 쉽게 구분할 수 있다.

Computer

- 사람이 쉽게 구분할 수 있는 이미지 구분하는 문제를 컴퓨터는 어려워한다.
- 모든 이미지를 숫자로 인식
- Pixel마다 밝기 정도에 따라 값을 가짐
- 값에 해당하는 이미지가 label 되어있어야 함.

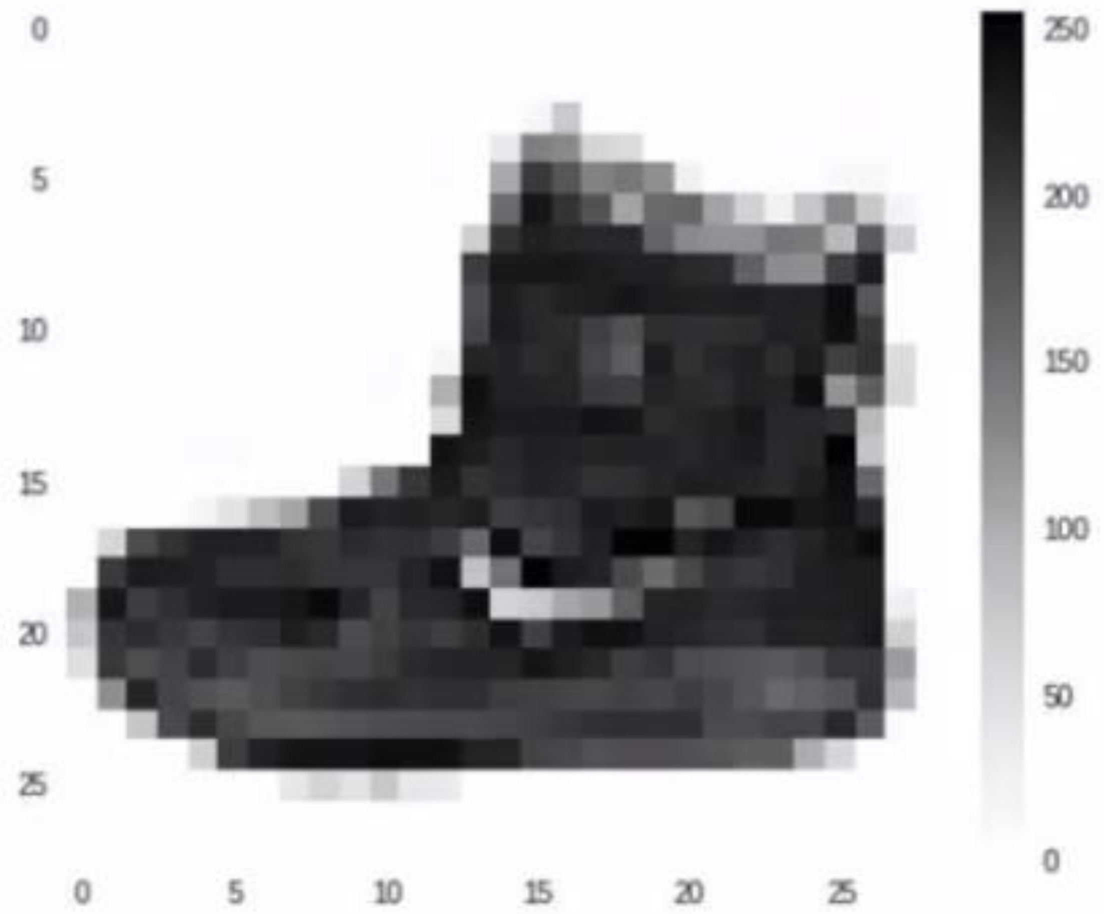
Fashion MNIST

- 70k Images
- 10 Categories
- Images are 28x28
- Can train a neural net!



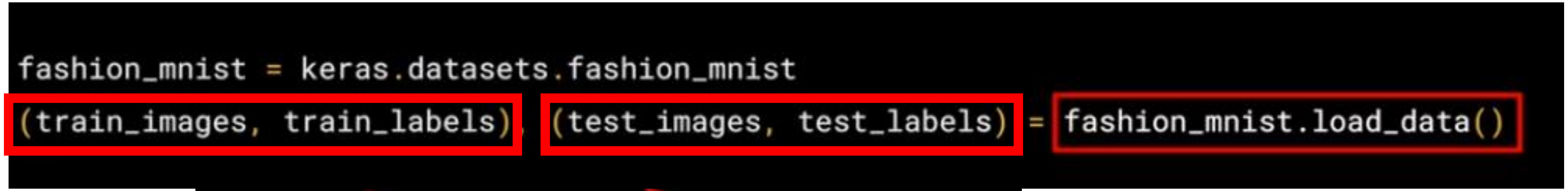
Fashion MNIST

- 70k Images
- 10 Categories
- Images are 28x28
- Can train a neural net!

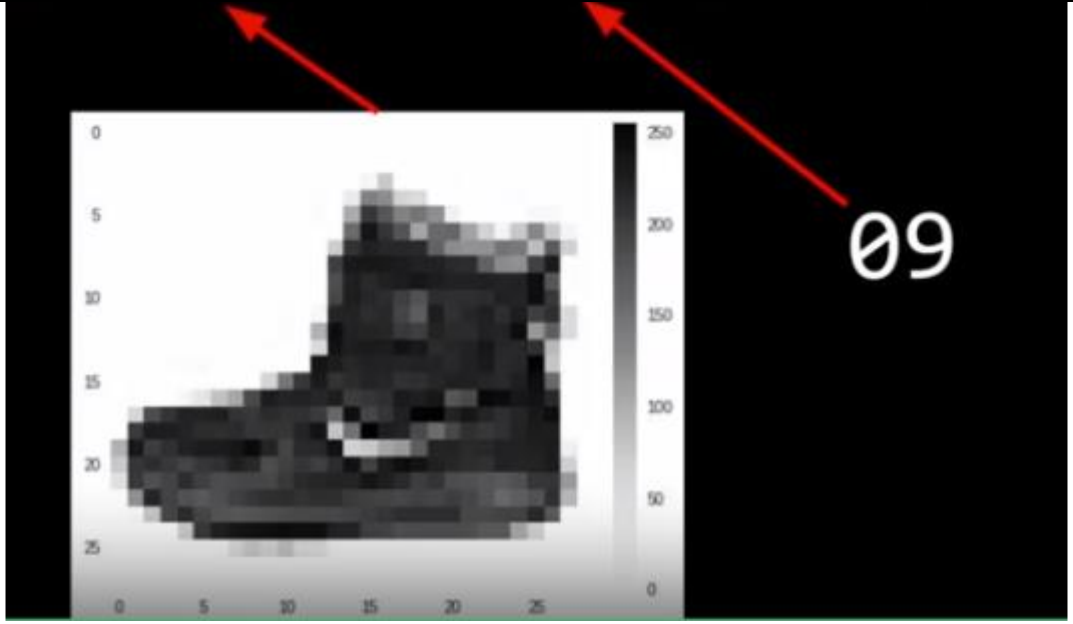


60000 / 70000

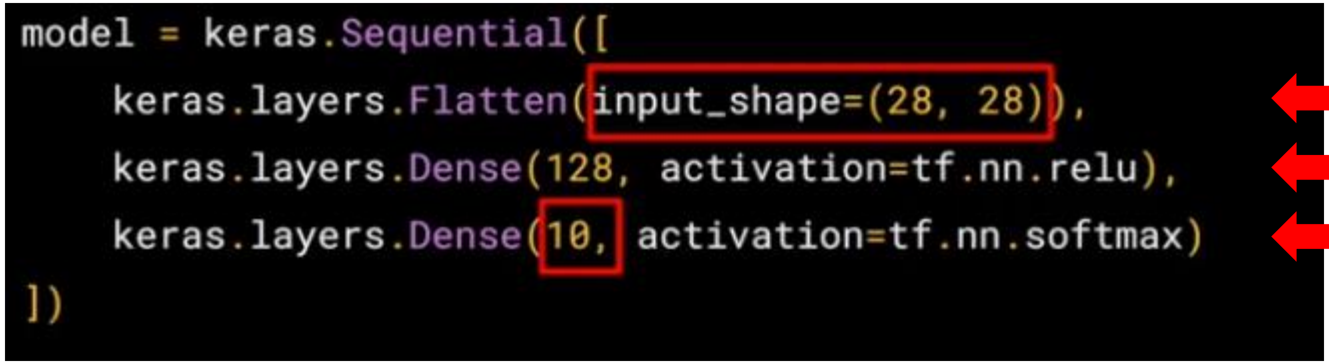
10000 / 70000



```
fashion_mnist = keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```



Layer 3개

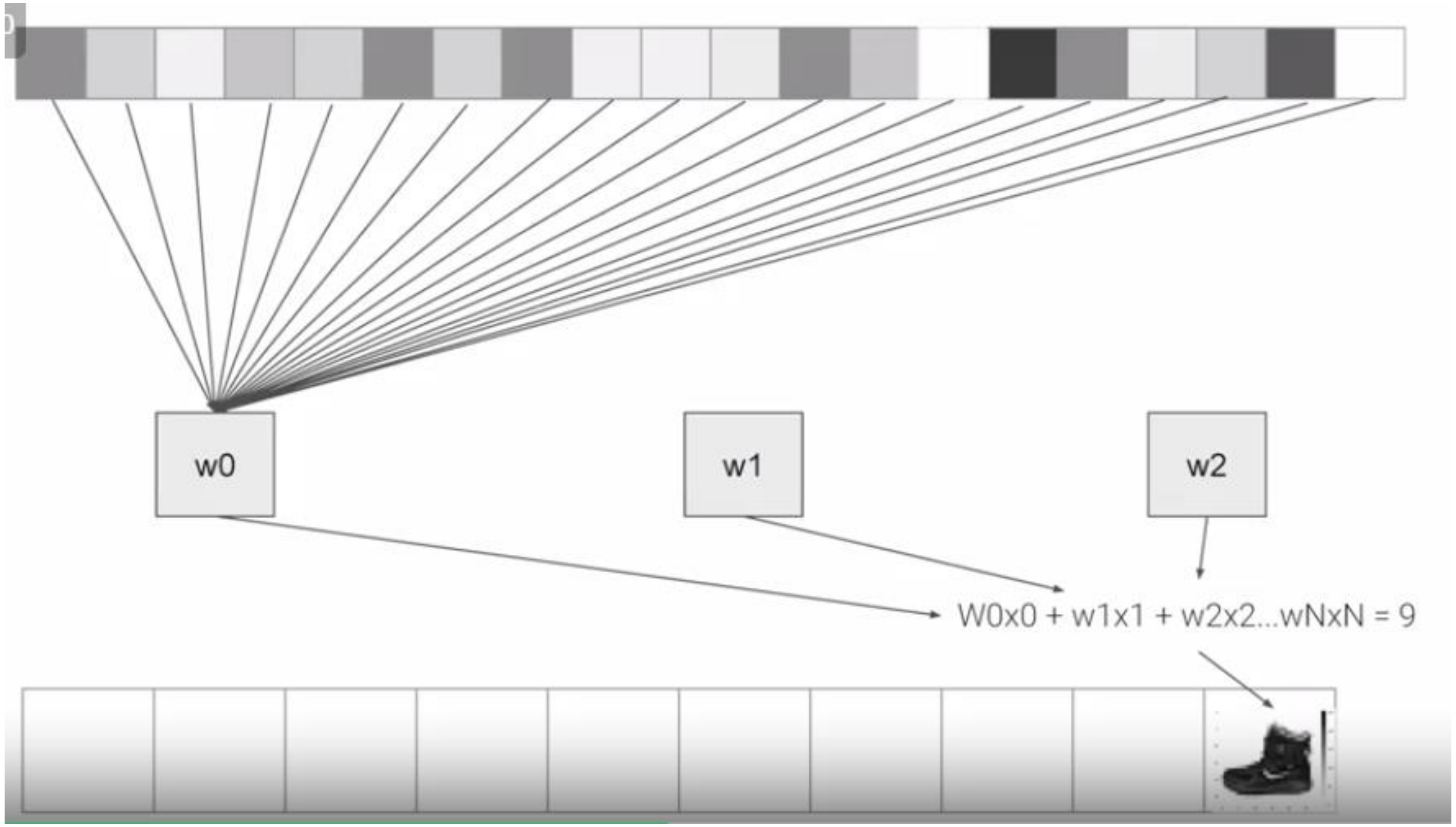


```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])
```

Image size

Hidden layer

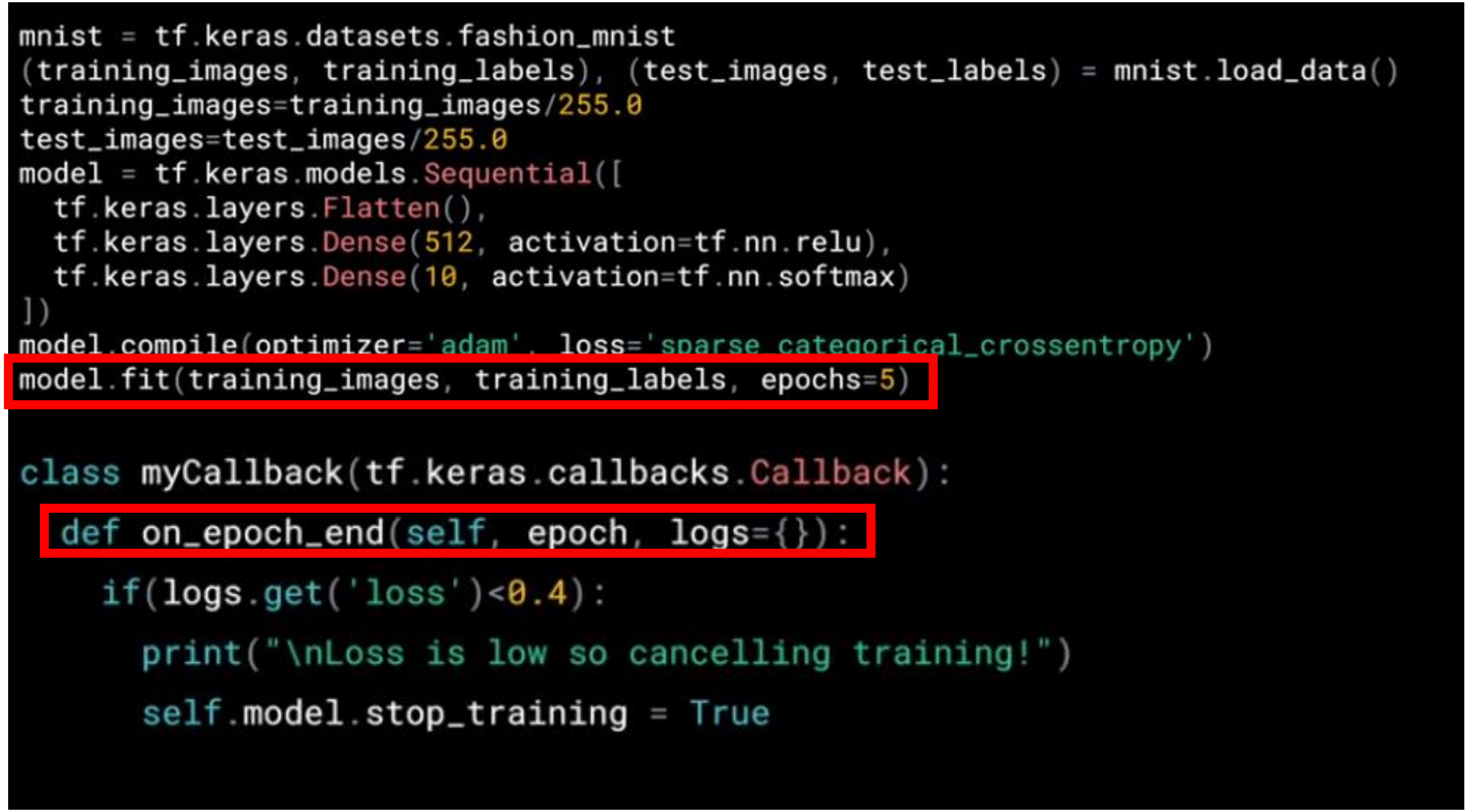
10 categories



Further study of machine learning by Andrew Ng

https://youtu.be/fXOsFF95ifk

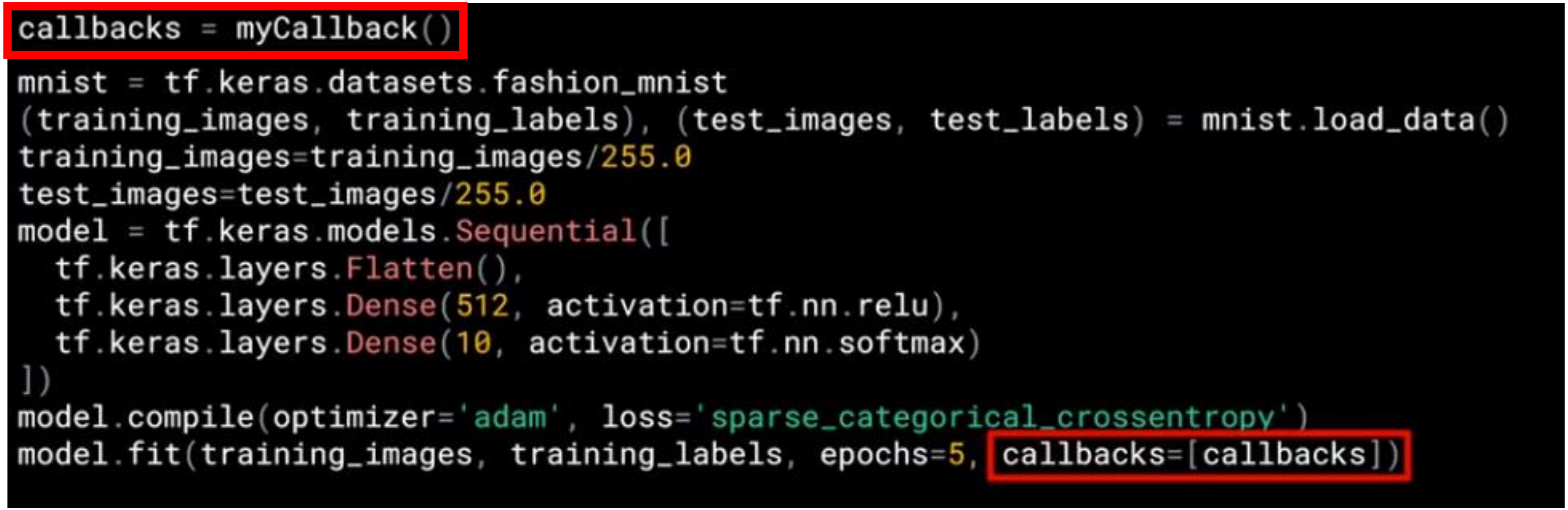
Using Callbacks to control training



```
mnist = tf.keras.datasets.fashion_mnist
(training_images, training_labels), (test_images, test_labels) = mnist.load_data()
training_images=training_images/255.0
test_images=test_images/255.0
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(512, activation=tf.nn.relu),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy')
model.fit(training_images, training_labels, epochs=5)

class myCallback(tf.keras.callbacks.Callback):
  def on_epoch_end(self, epoch, logs={}):
    if(logs.get('loss')<0.4):
      print("\nLoss is low so cancelling training!")
      self.model.stop_training = True
```

Using Callbacks to control training



```
callbacks = myCallback()
mnist = tf.keras.datasets.fashion_mnist
(training_images, training_labels), (test_images, test_labels) = mnist.load_data()
training_images=training_images/255.0
test_images=test_images/255.0
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(512, activation=tf.nn.relu),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy')
model.fit(training_images, training_labels, epochs=5, callbacks=[callbacks])
```