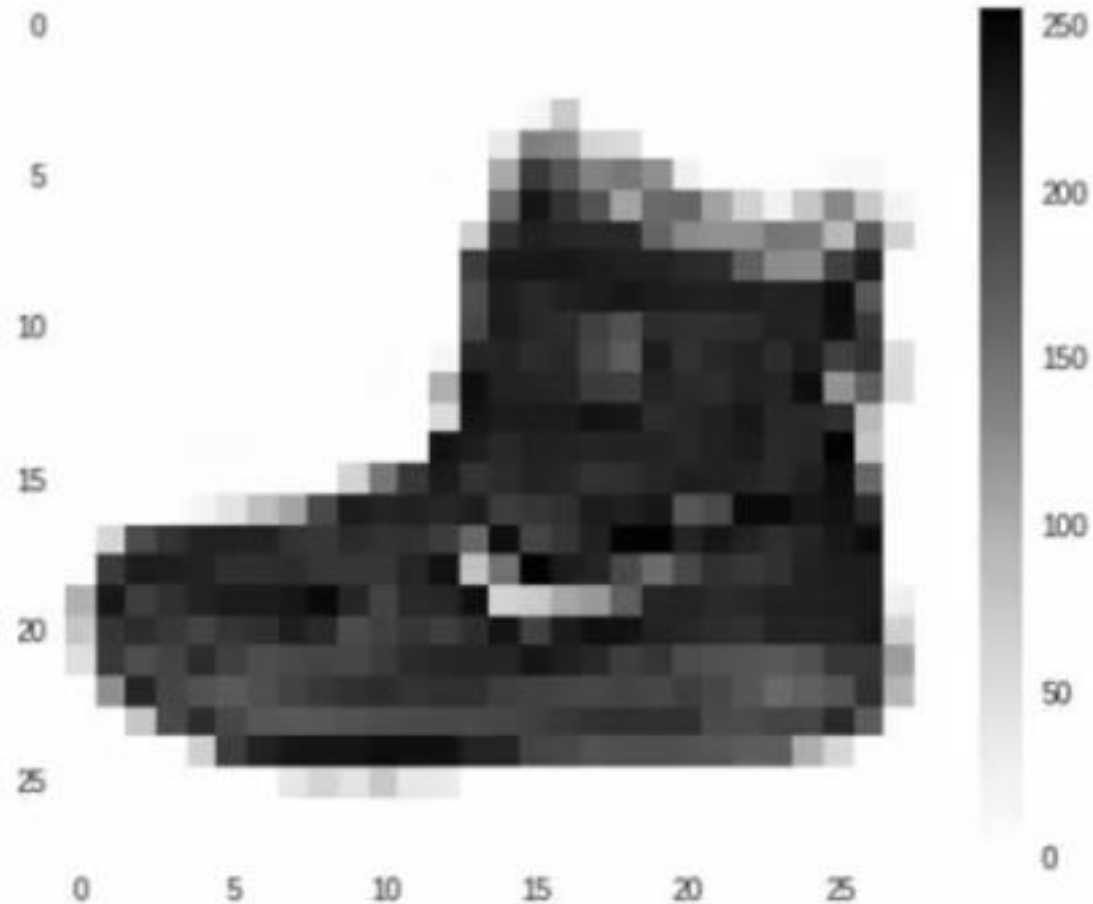
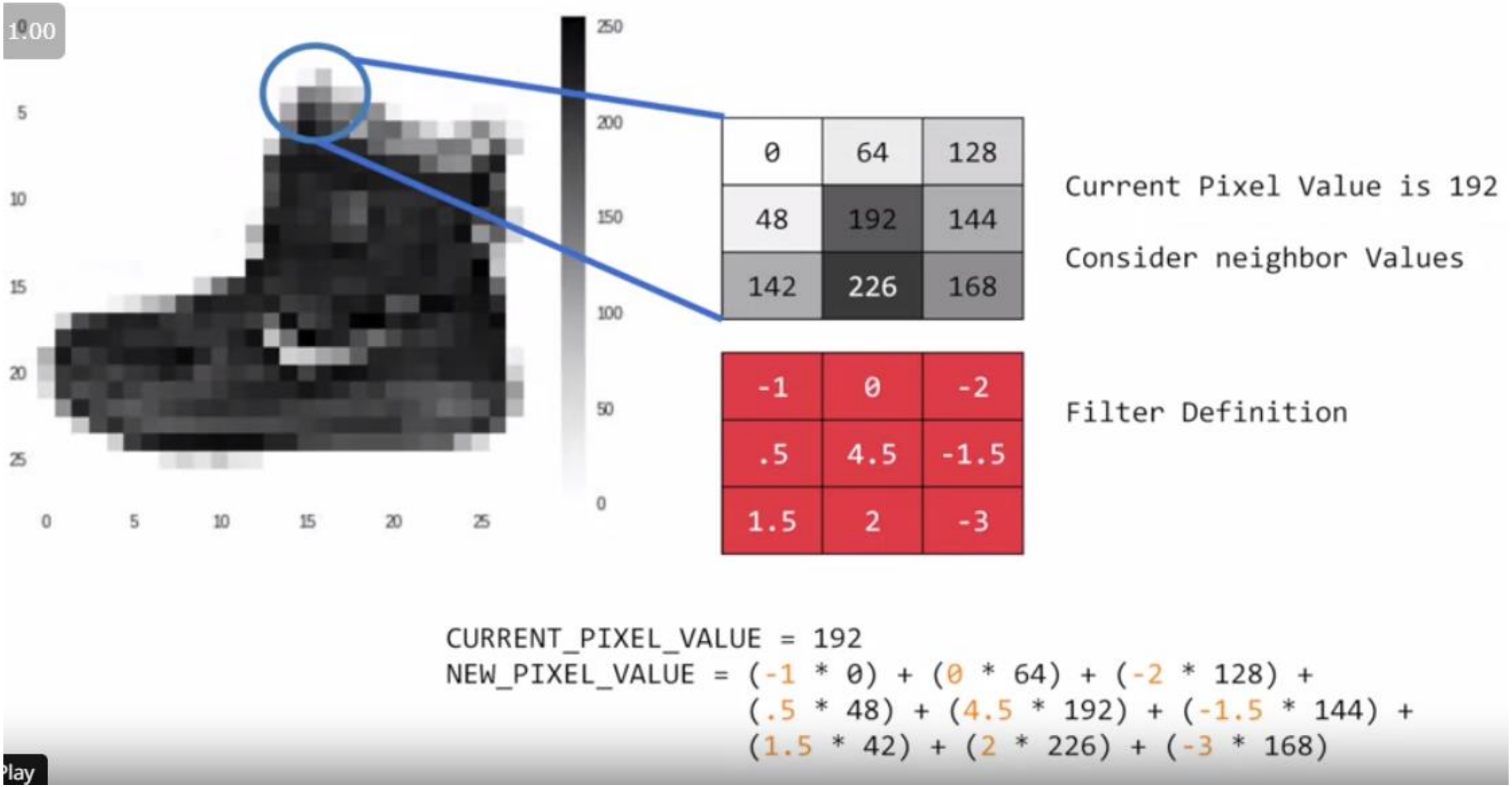
Week3. Enhancing Vision with Convolutional Neural Networks

Fashion MNIST

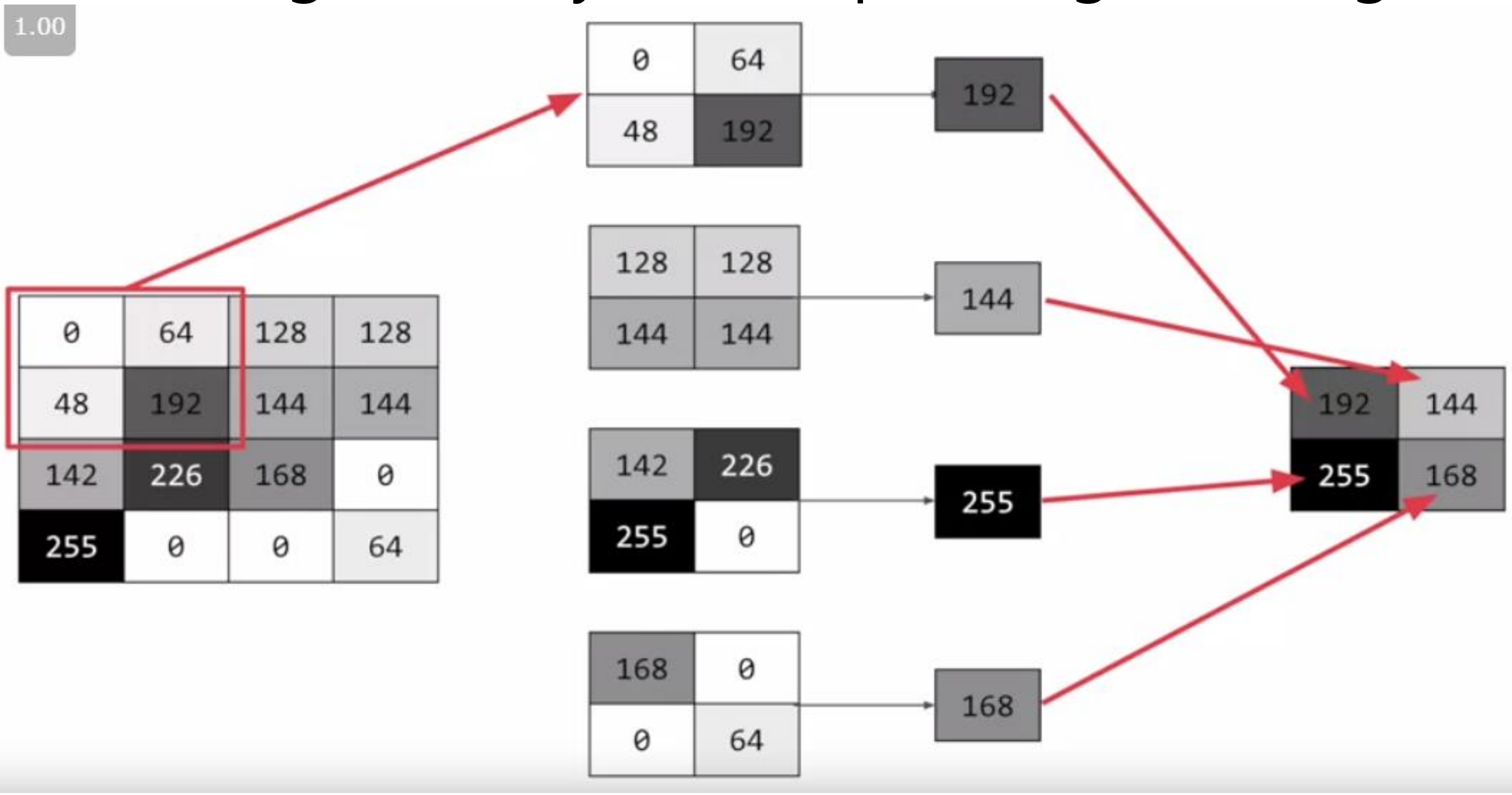
- 70k Images
- 10 Categories
- Images are 28x28
- Can train a neural net!



Pooling is a way of compressing an image



Pooling is a way of compressing an image



3*3크기를 가진 64개의 filter

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu',
                         input_shape=(28, 28, 1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

Convolutional Neural Networks (Course 4 of the Deep Learning Specialization)

42 videos • 415,722 views • Last updated on Nov 7, 2017

Deeplearning.ai

SUBSCRIBE 28K

1 C4W1L01 Computer Vision — Deeplearning.ai — 5:44

2 C4W1L02 Edge Detection Examples — Deeplearning.ai — 11:31

3 C4W1L03 More Edge Detection — Deeplearning.ai — 7:58

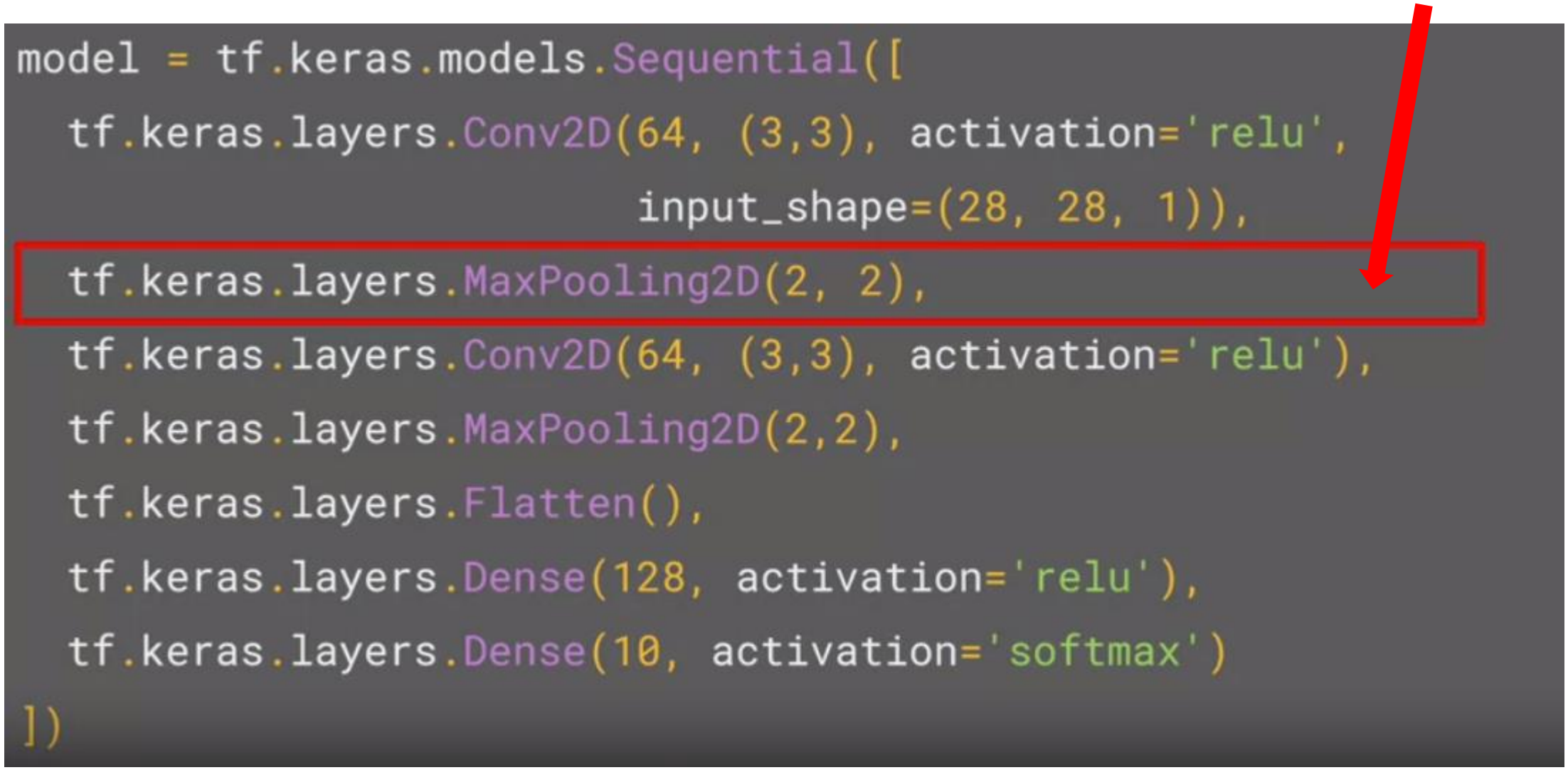
4 C4W1L04 Padding — Deeplearning.ai — 9:50

5 C4W1L05 Strided Convolutions — Deeplearning.ai — 9:02

6 C4W1L06 Convolutions Over Volumes — Deeplearning.ai — 10:45

https://bit.ly/2UGa7uH

2*2 중에서 가장 큰 값 선택



```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu',
                         input_shape=(28, 28, 1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

Convolution layer 추가로 dense layer 생성

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu',
                         input_shape=(28, 28, 1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

```
model.summary()
```

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_12 (Conv2D)           (None, 26, 26, 64)        640
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 13, 13, 64)        0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 11, 11, 64)        36928
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 5, 5, 64)          0
_________________________________________________________________
flatten_5 (Flatten)          (None, 1600)              0
_________________________________________________________________
dense_10 (Dense)             (None, 128)               204928
_________________________________________________________________
dense_11 (Dense)             (None, 10)                1290
=================================================================
```