Intelligent Vehicle

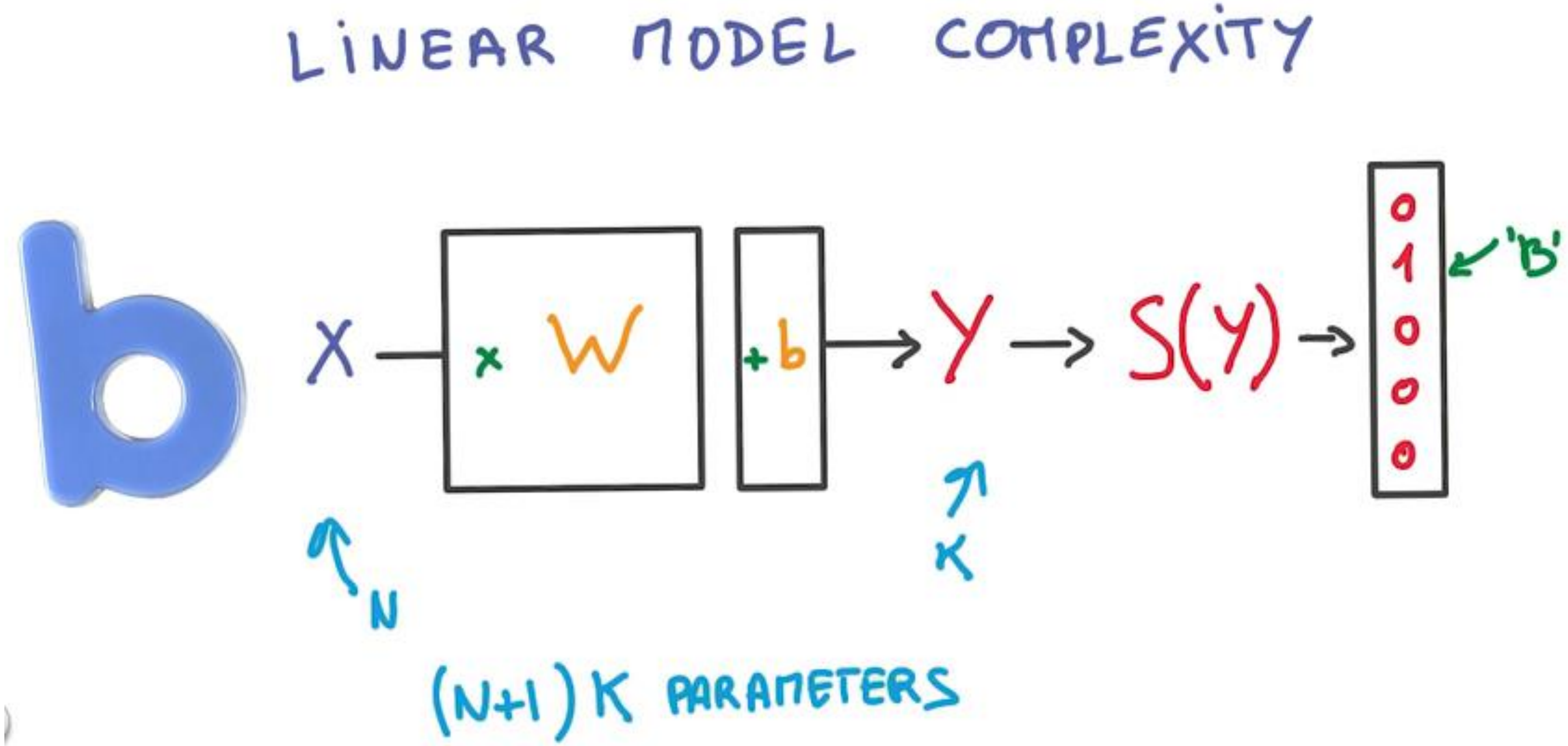
Introduction of TensorFlow Ⅱ

Yeqiang Qian

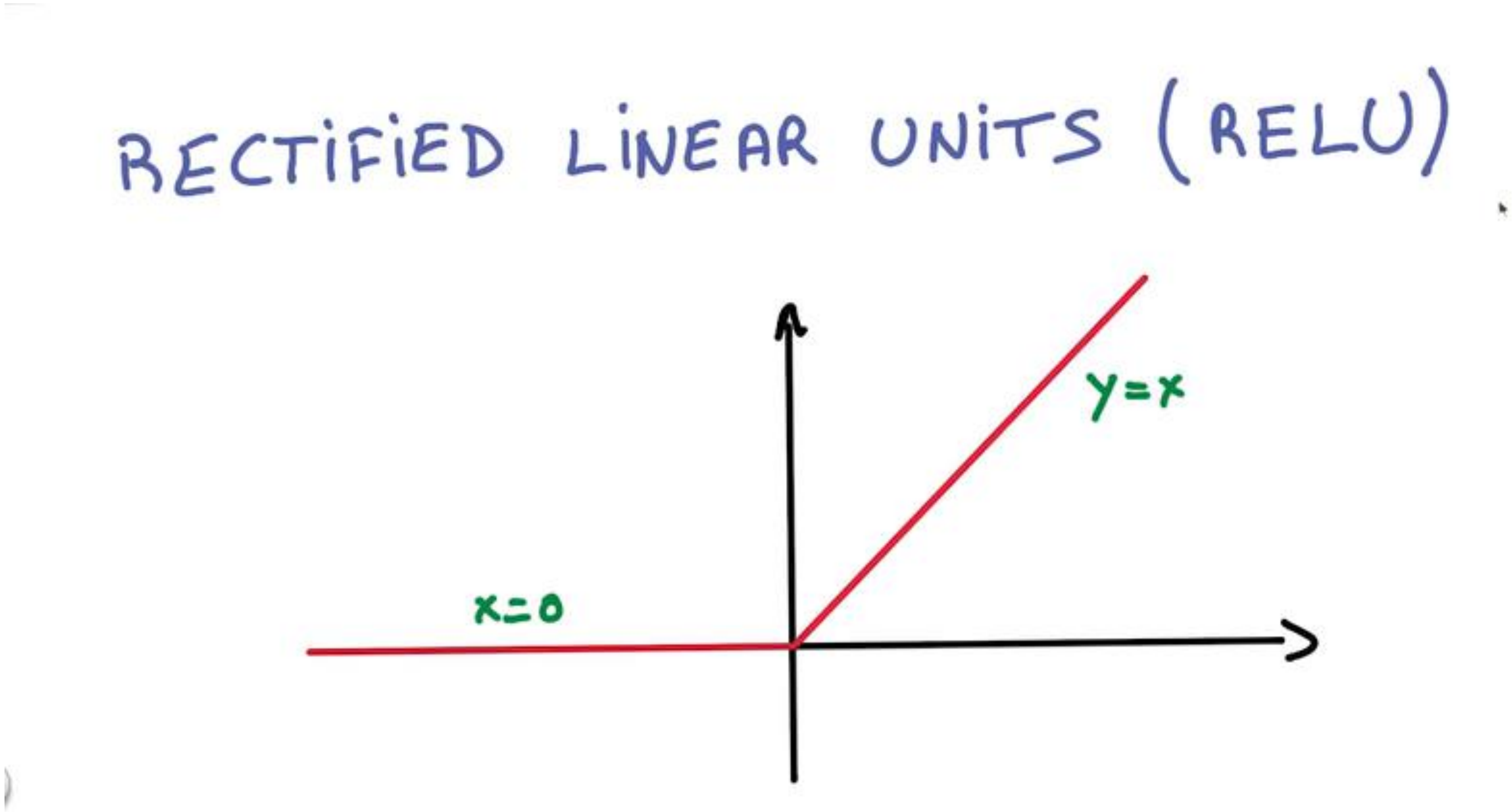
上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

- Today’s agenda
  - DNN continuation
  - Introduction of Convolutional Neural Networks (CNN)
  - Practice： create your own CNN using TensorFlow

Linear Models are Limited

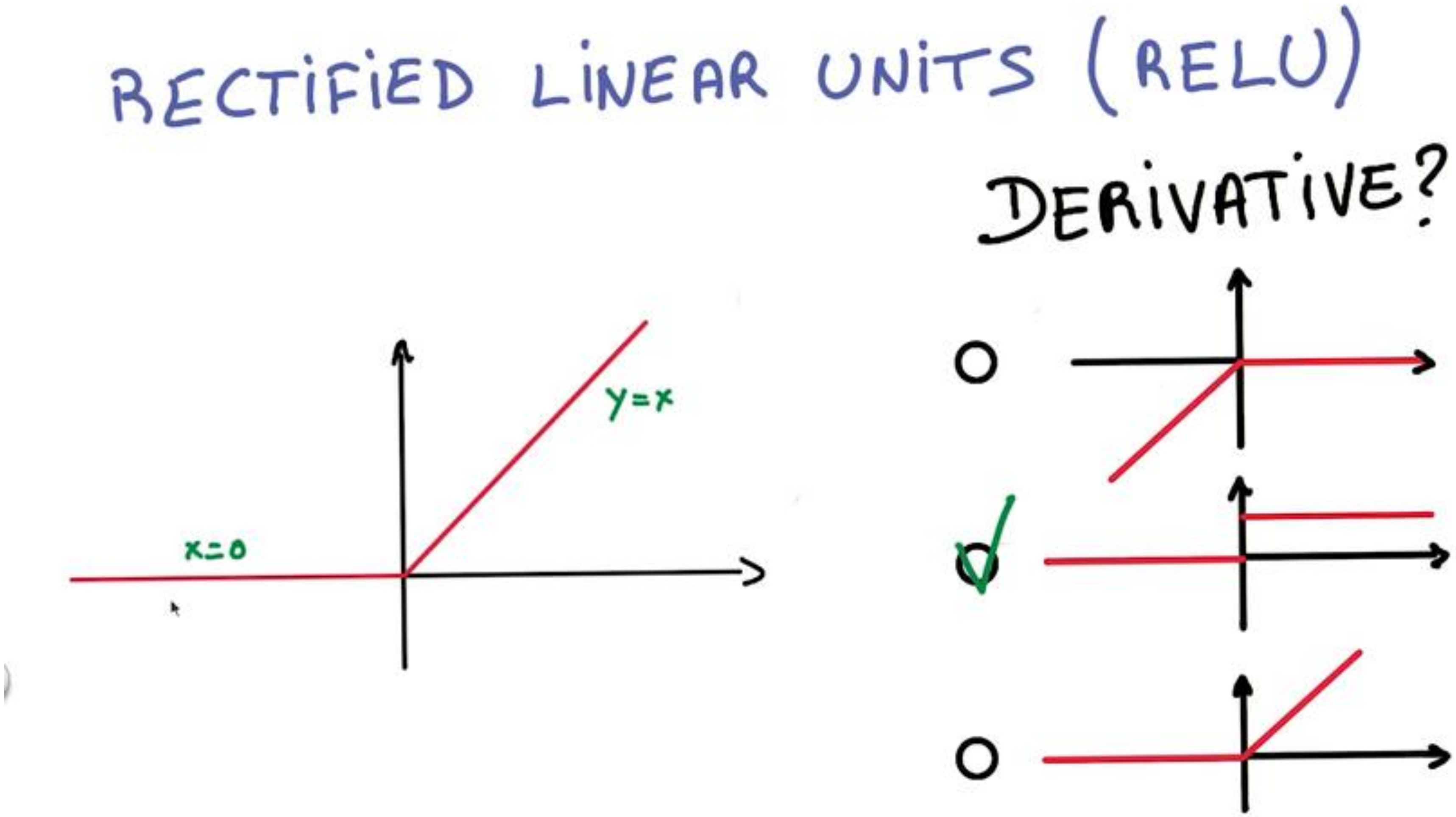


Rectified Linear Units



RECTIFIED LINEAR UNITS (RELU)

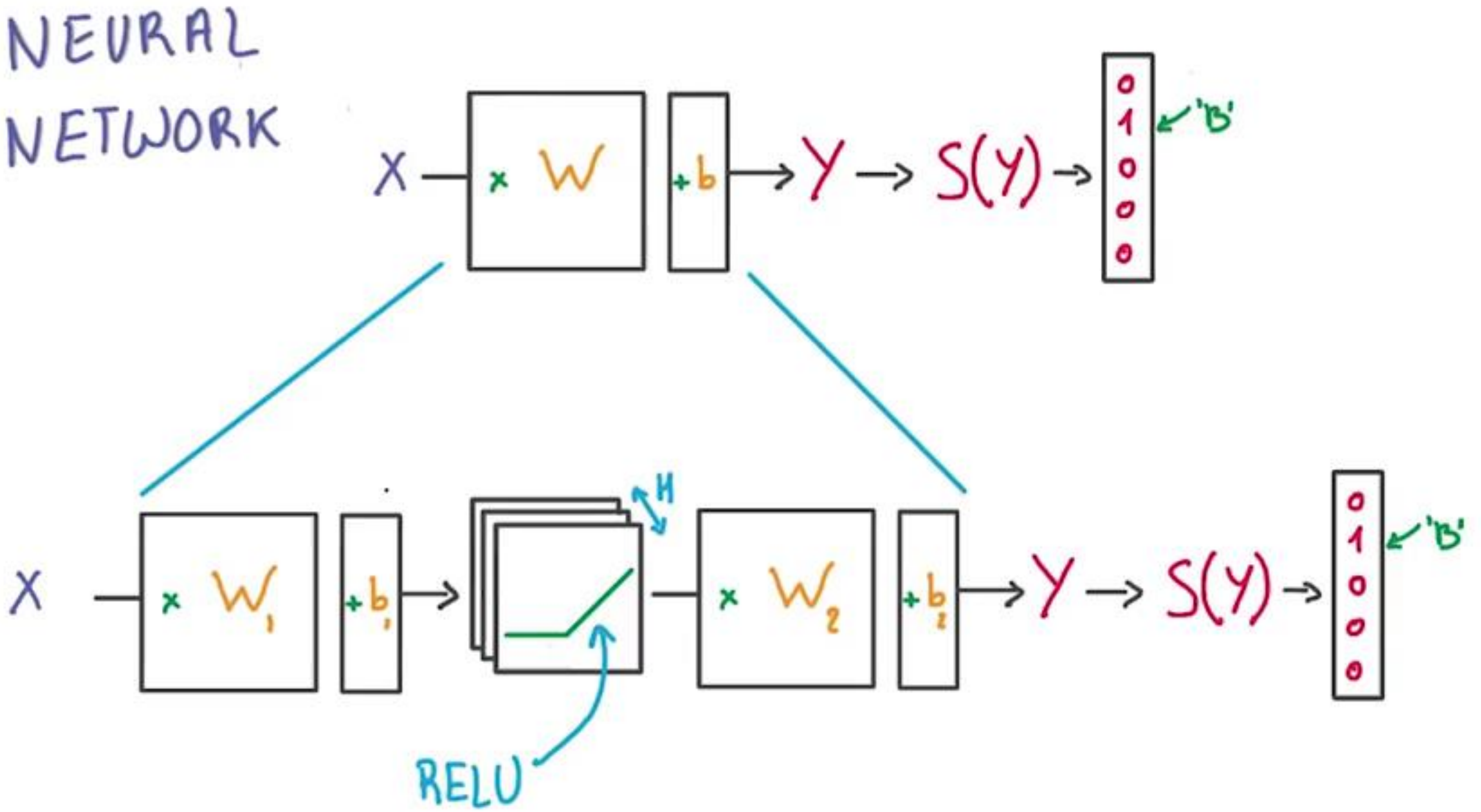
Rectified Linear Units Solution



RECTIFIED LINEAR UNITS (RELU)

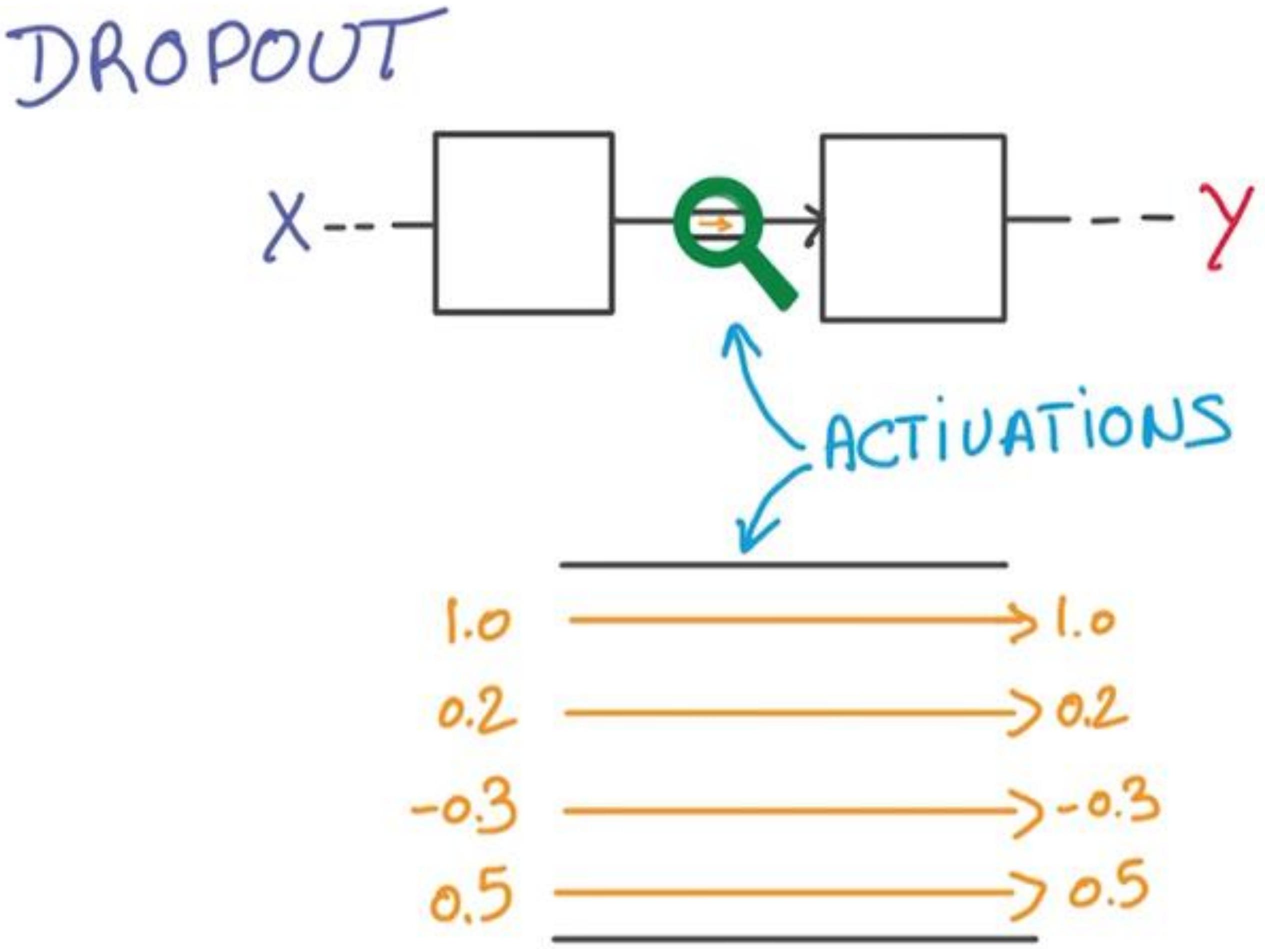
DERIVATIVE?

Network of ReLUs

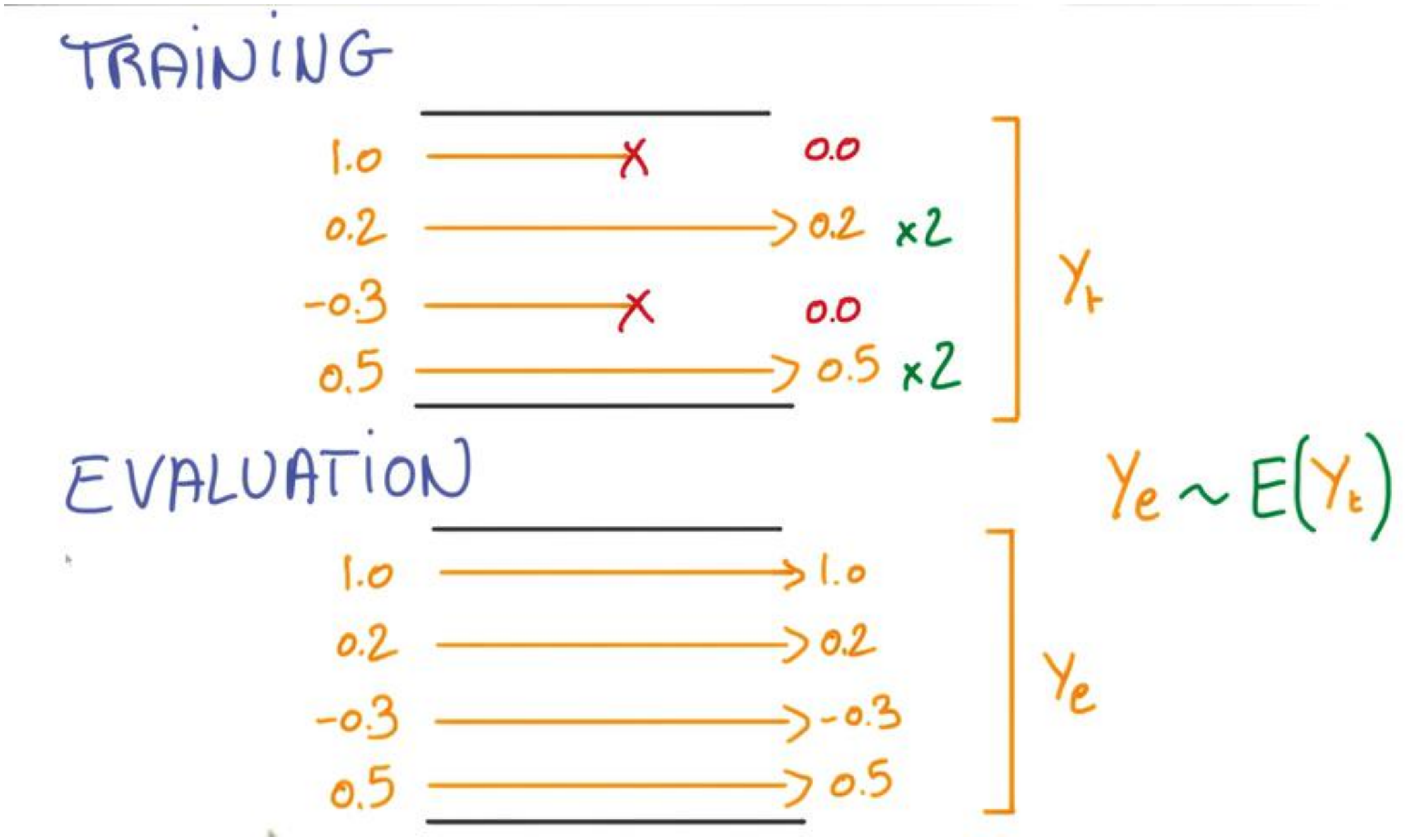


Regularization

Dropout



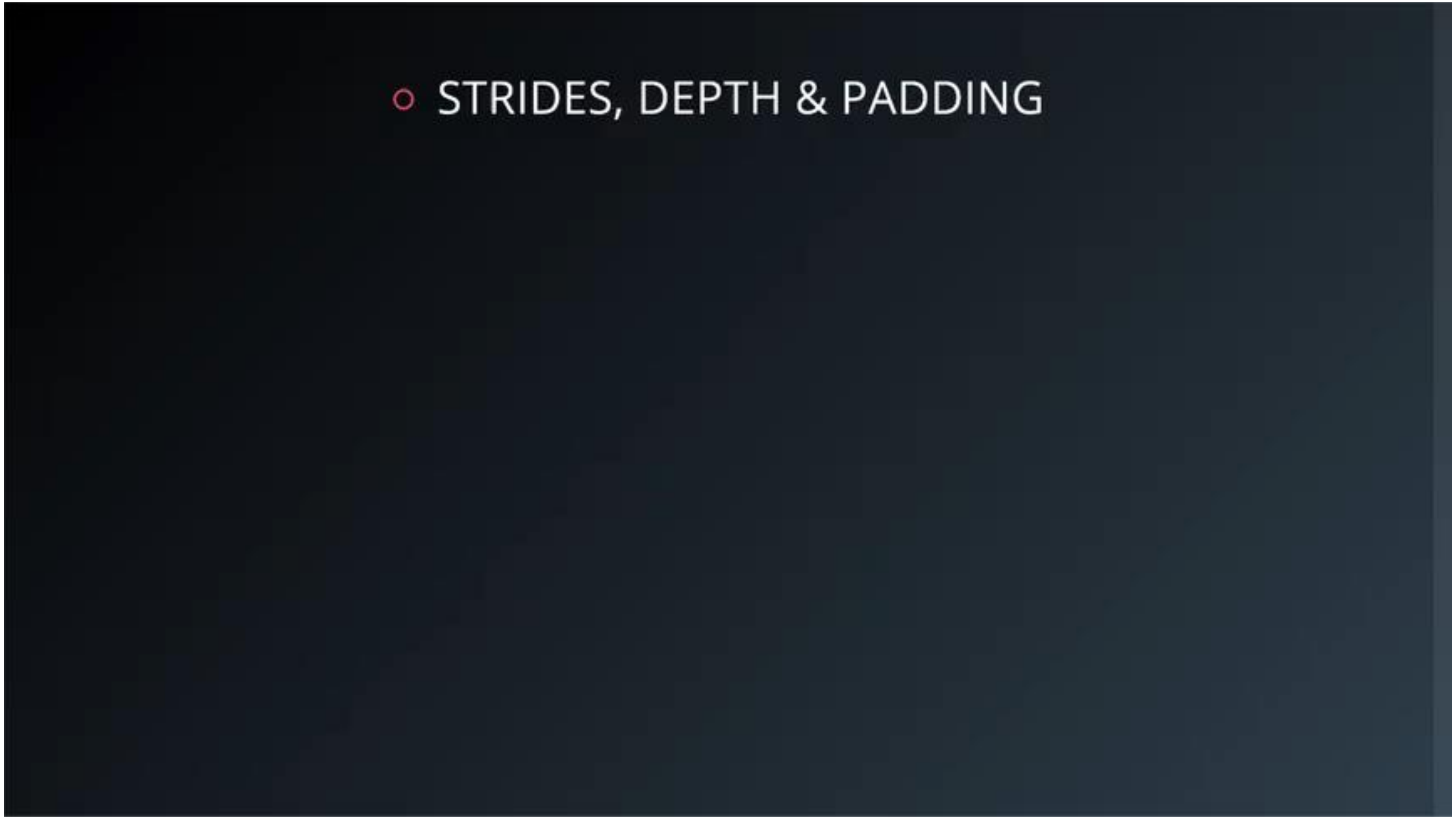
Dropout Pt. 2


TRAINING
1.0
0.2
-0.3
0.5
0.0
0.2 x2
0.0
0.5 x2
Y_t
EVALUATION
$Y_e \sim E(Y_t)$
1.0
0.2
-0.3
0.5
1.0
0.2
-0.3
0.5
Y_e

Statistical Invariance

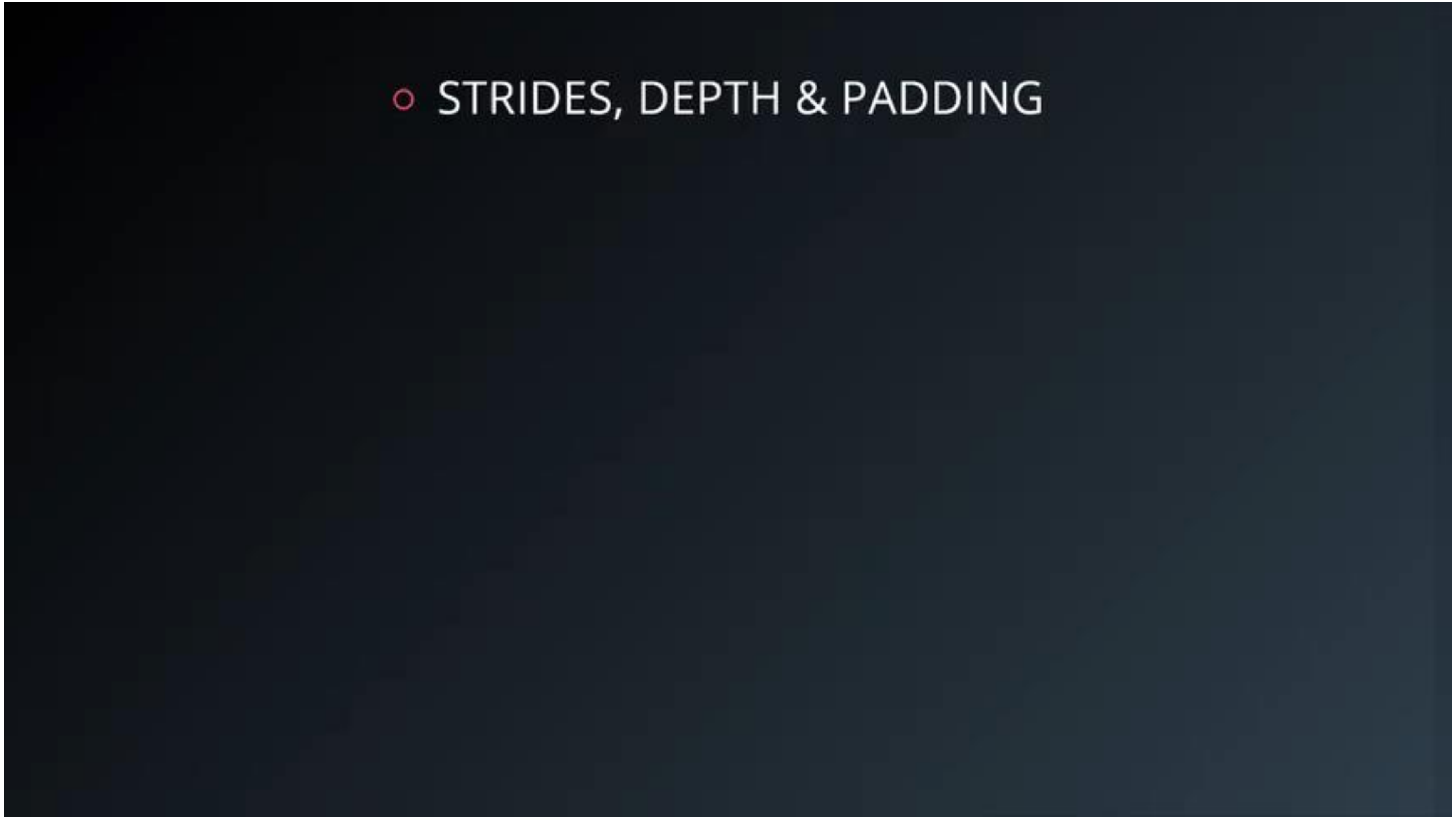
Convolutional Networks

Feature Map Sizes



- STRIDES, DEPTH & PADDING

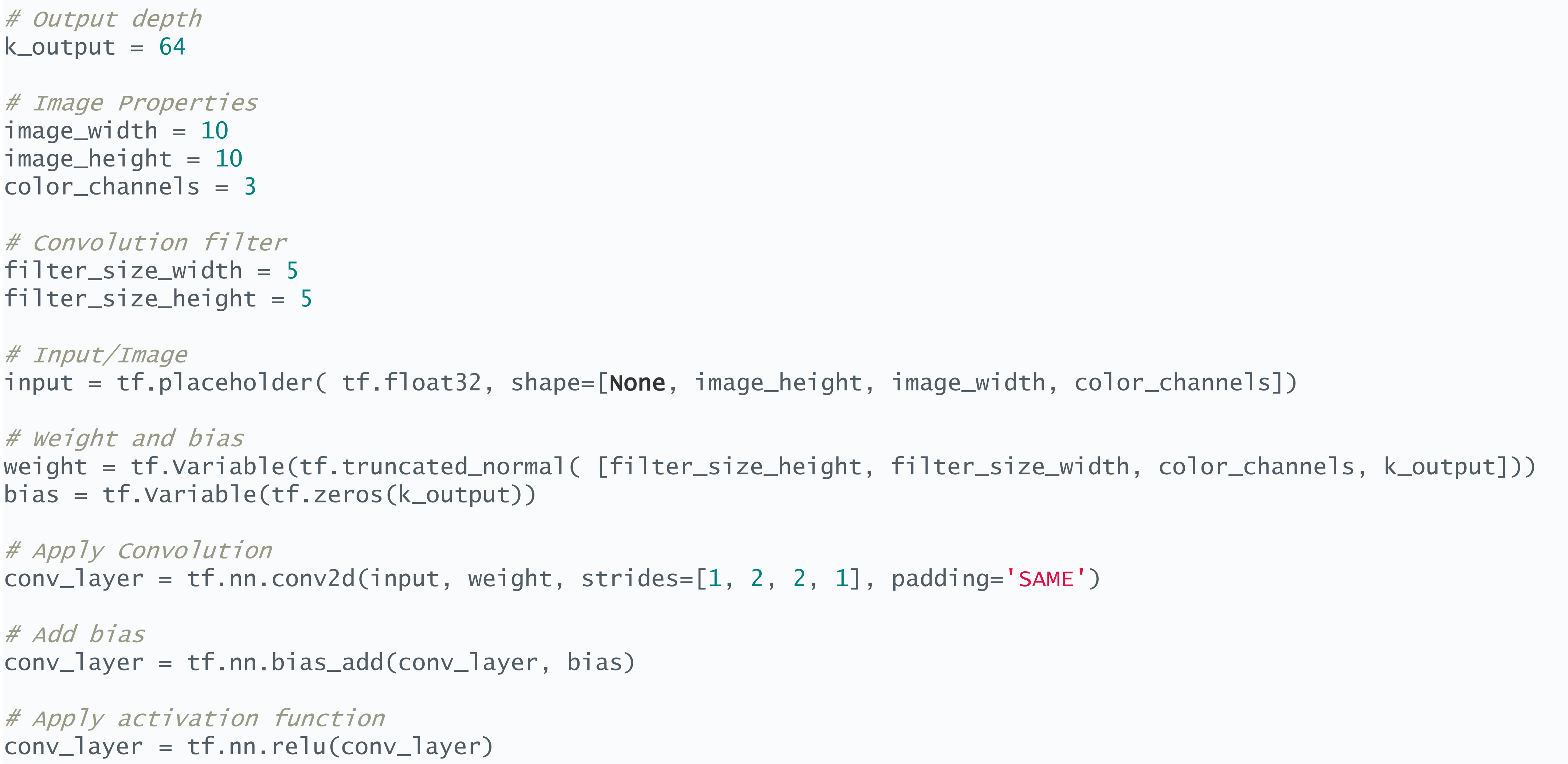
Feature Map Sizes Solution



- STRIDES, DEPTH & PADDING

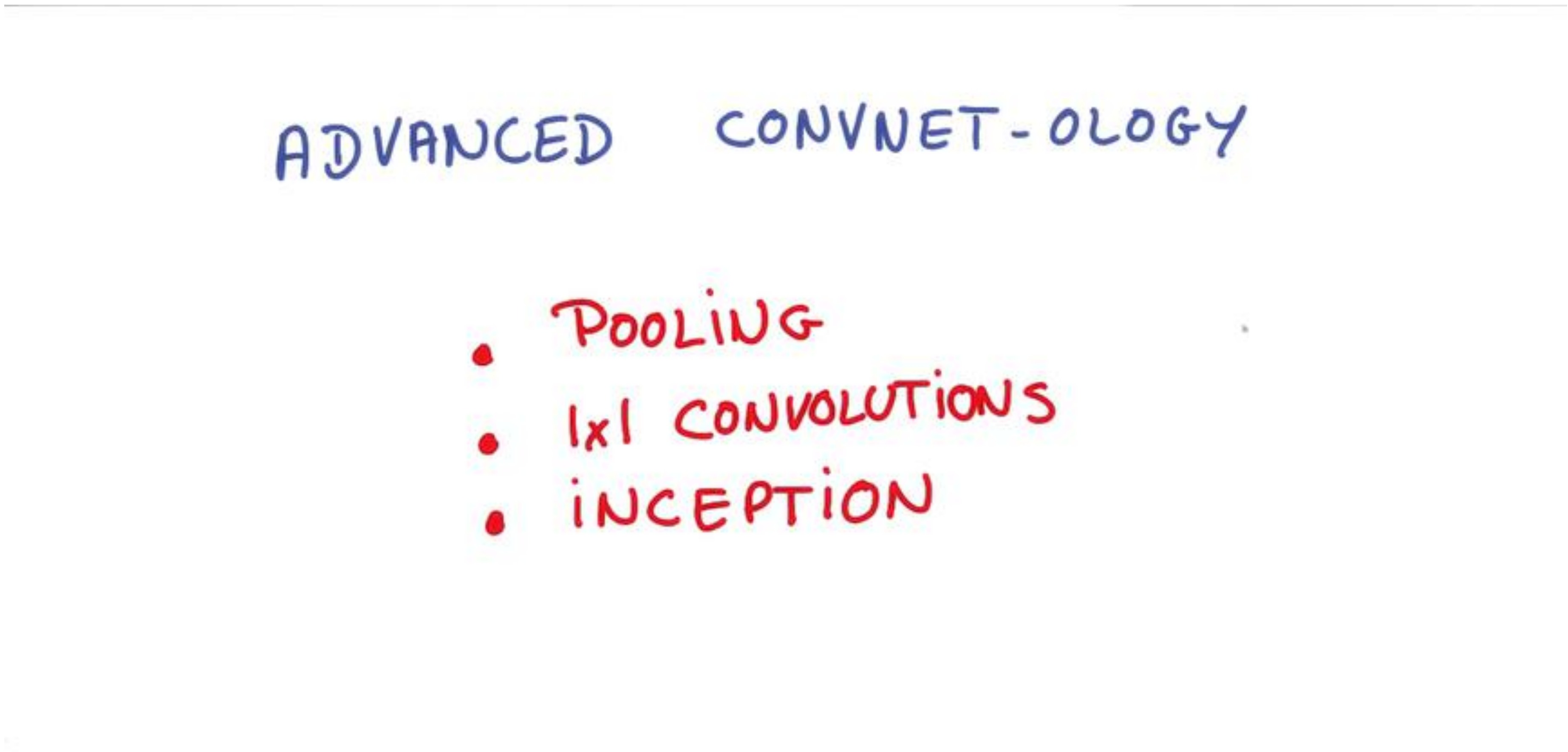
Convolutions continued

TensorFlow Convolution Layer



```
# Output depth
k_output = 64

# Image Properties
image_width = 10
image_height = 10
color_channels = 3

# Convolution filter
filter_size_width = 5
filter_size_height = 5

# Input/Image
input = tf.placeholder( tf.float32, shape=[None, image_height, image_width, color_channels])

# Weight and bias
weight = tf.Variable(tf.truncated_normal( [filter_size_height, filter_size_width, color_channels, k_output]))
bias = tf.Variable(tf.zeros(k_output))

# Apply Convolution
conv_layer = tf.nn.conv2d(input, weight, strides=[1, 2, 2, 1], padding='SAME')

# Add bias
conv_layer = tf.nn.bias_add(conv_layer, bias)

# Apply activation function
conv_layer = tf.nn.relu(conv_layer)
```

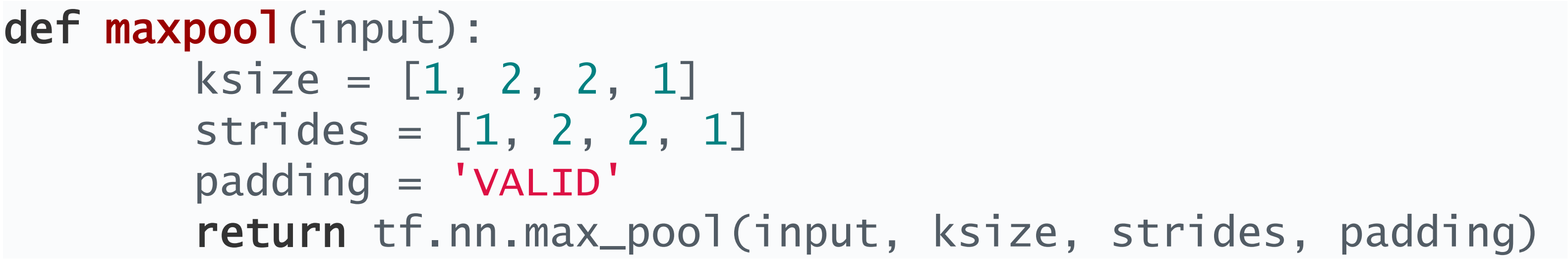
Explore The Design Space



ADVANCED CONVNET-OLOGY

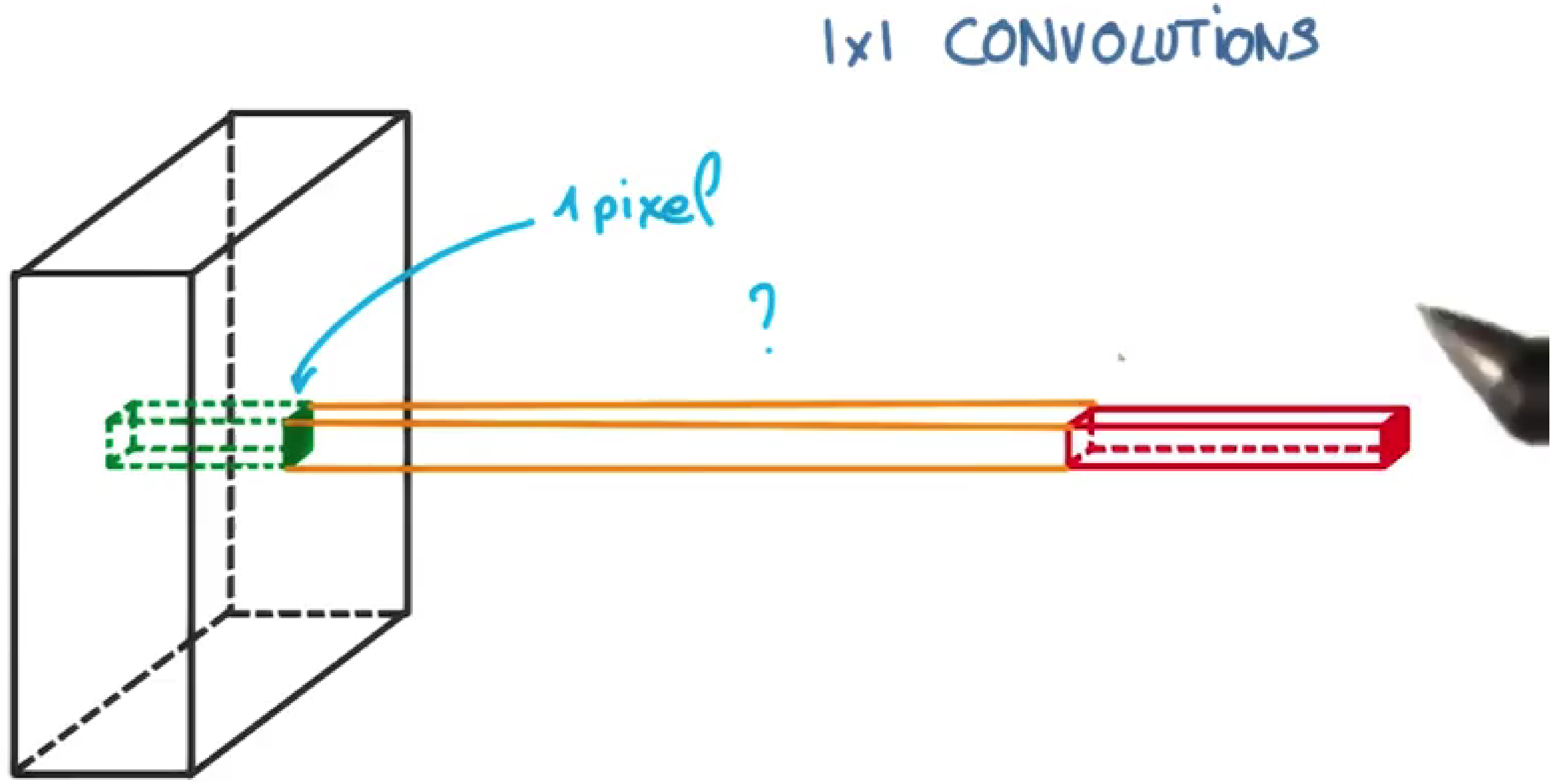
- POOLING
- 1x1 CONVOLUTIONS
- INCEPTION

TensorFlow Pooling Layer

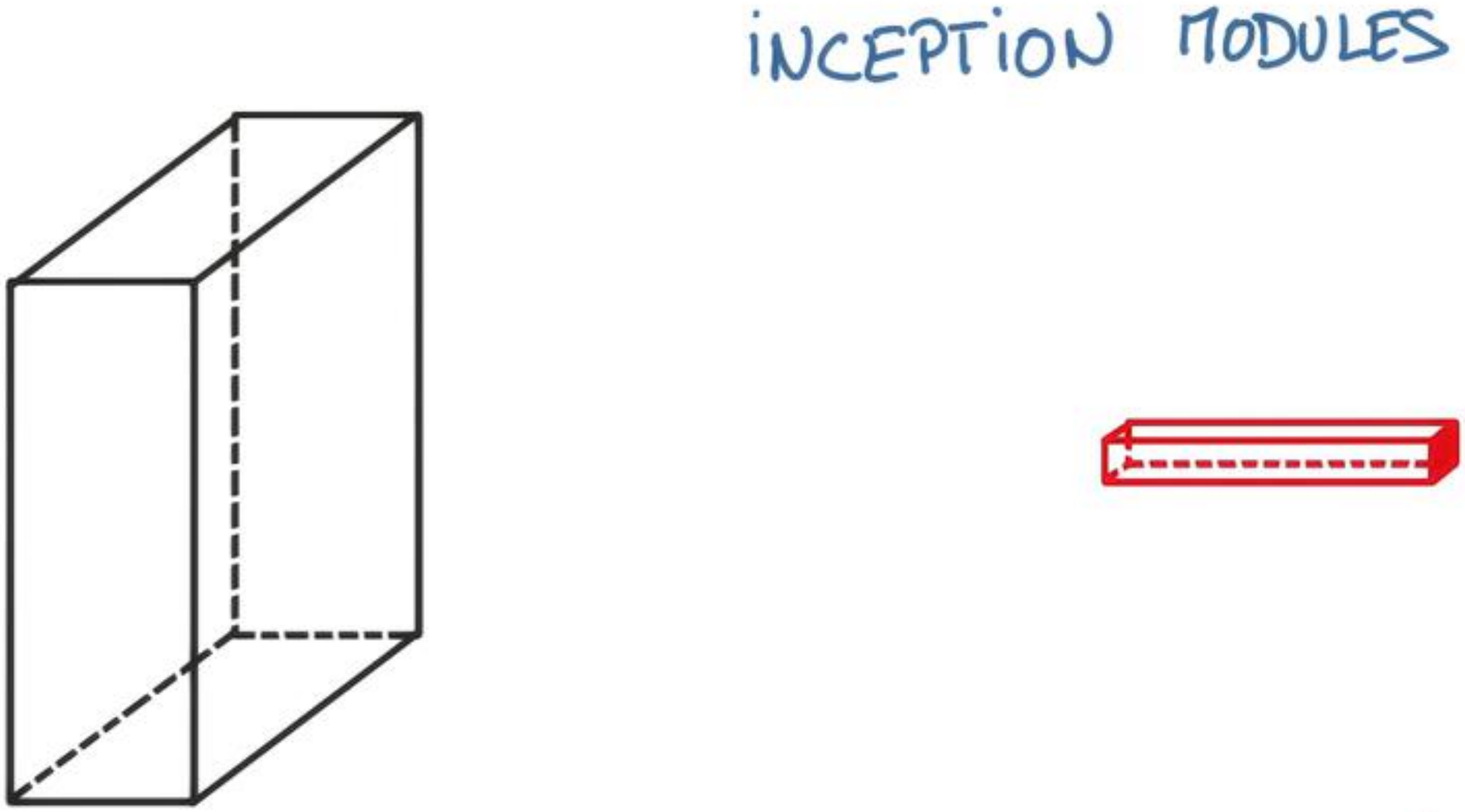


```
def maxpool(input):
        ksize = [1, 2, 2, 1]
        strides = [1, 2, 2, 1]
        padding = 'VALID'
        return tf.nn.max_pool(input, ksize, strides, padding)
```

1x1 Convolutions



Inception Module



LeNet In TensorFlow

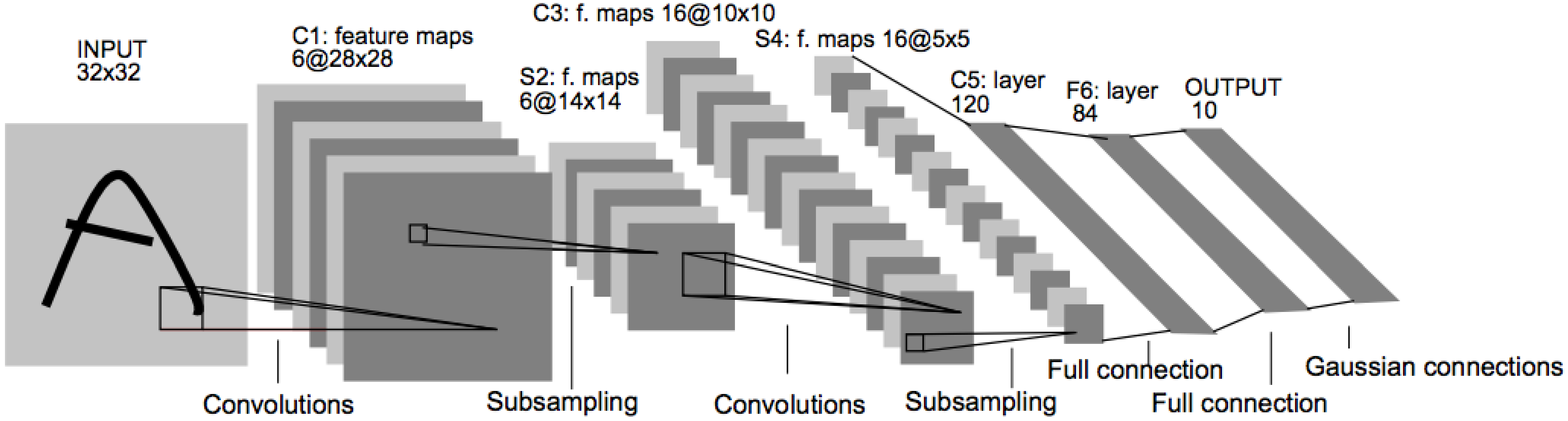


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Practice

```
0.sudo pip install jupyter
1.Link: https://jbox.sjtu.edu.cn/l/uoaCwx Password: wyvh
```

- Today’s agenda
  - DNN continuation
  - Introduction of CNN
  - Practice：create your own CNN using TensorFlow

Thank you！