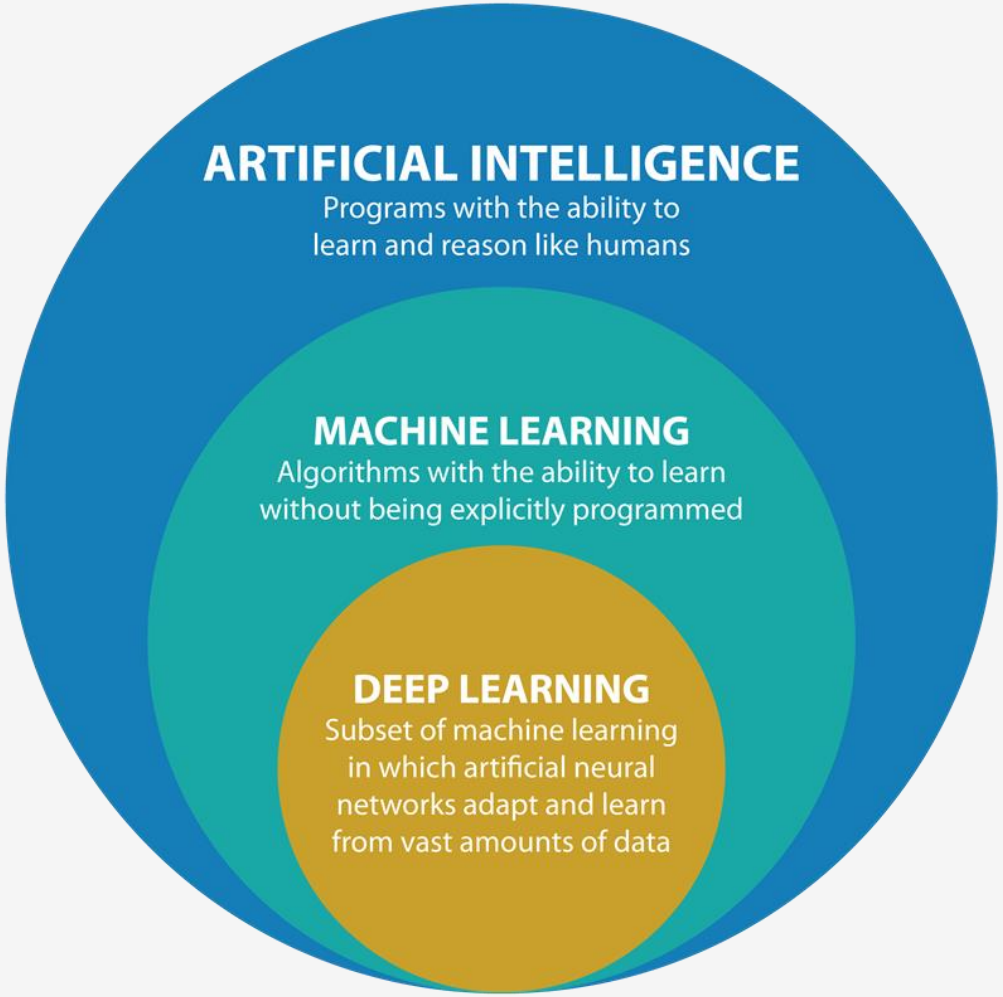
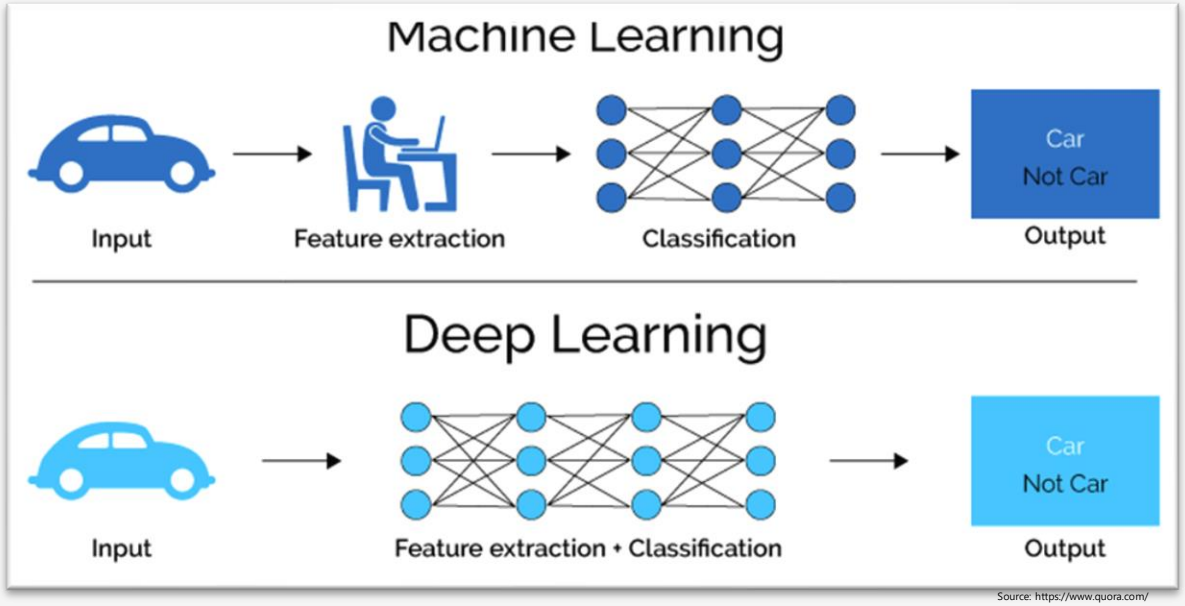
Overview of AI

Introduction to Deep learning / AI

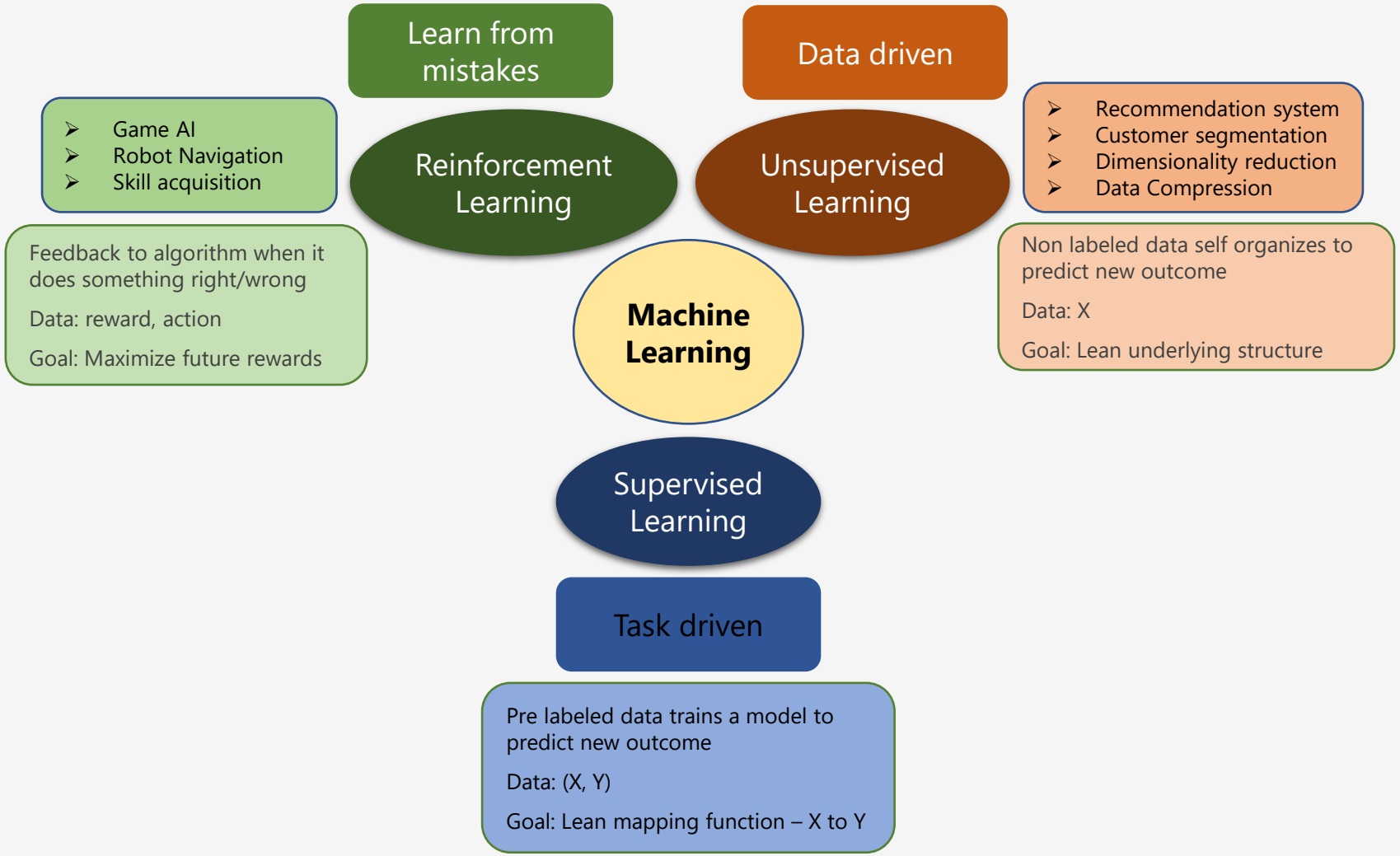


Intelligence: "the ability to learn, understand and think

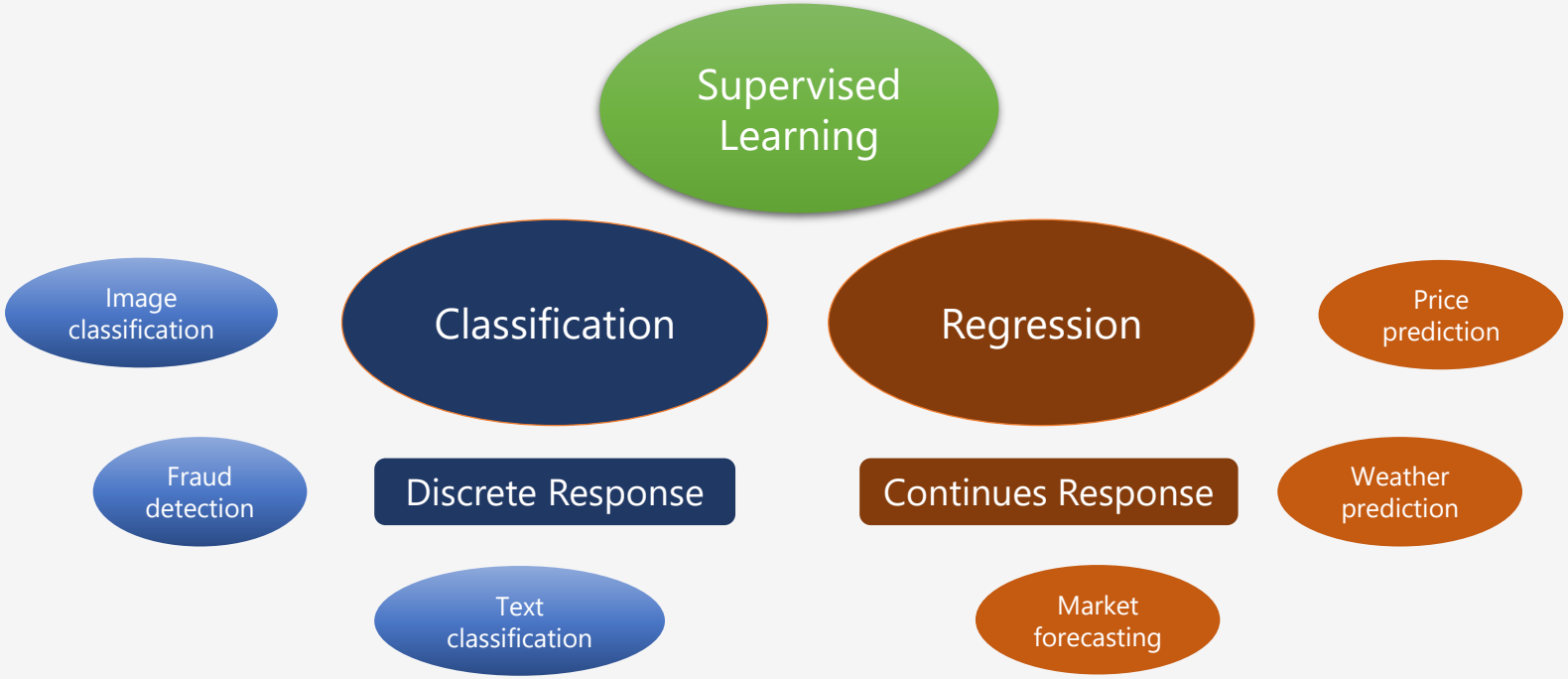


Source: https://www.quora.com/

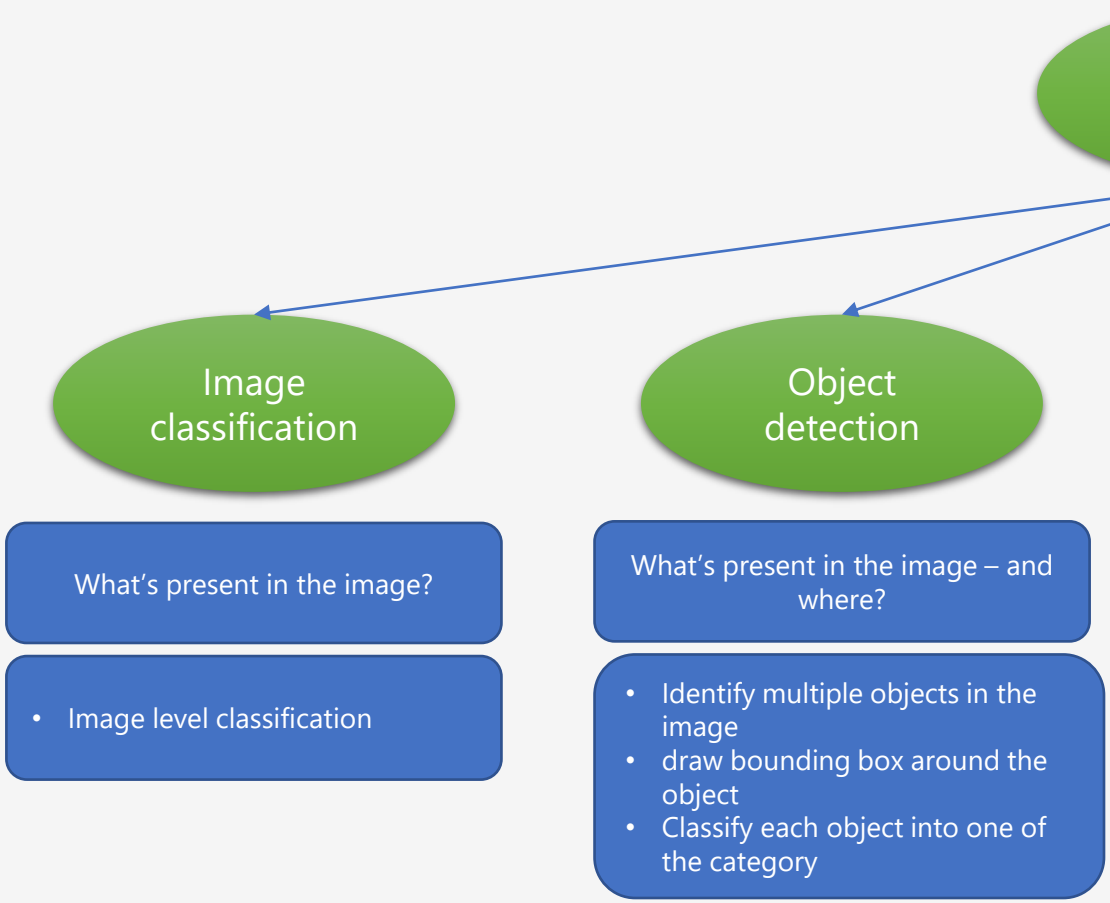
Major Types


Learn from mistakes
Data driven
Game AI
Robot Navigation
Skill acquisition
Reinforcement Learning
Unsupervised Learning
Recommendation system
Customer segmentation
Dimensionality reduction
Data Compression
Feedback to algorithm when it does something right/wrong
Data: reward, action
Goal: Maximize future rewards
Machine Learning
Non labeled data self organizes to predict new outcome
Data: X
Goal: Lean underlying structure
Supervised Learning
Task driven
Pre labeled data trains a model to predict new outcome
Data: (X, Y)
Goal: Lean mapping function – X to Y

Major Types

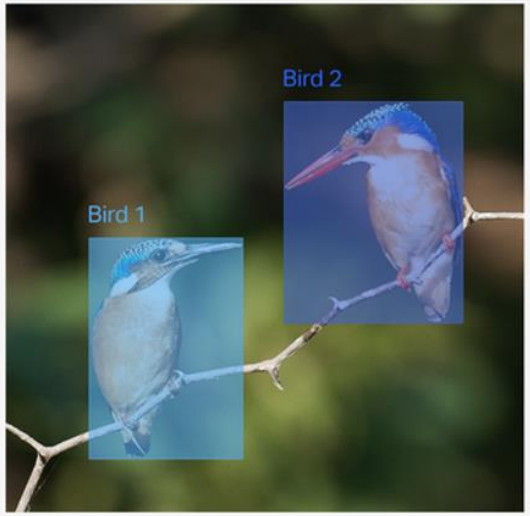

Supervised Learning
Classification
Regression
Image classification
Fraud detection
Text classification
Discrete Response
Continues Response
Price prediction
Weather prediction
Market forecasting

Major Types



Classification

Detection



Major Types

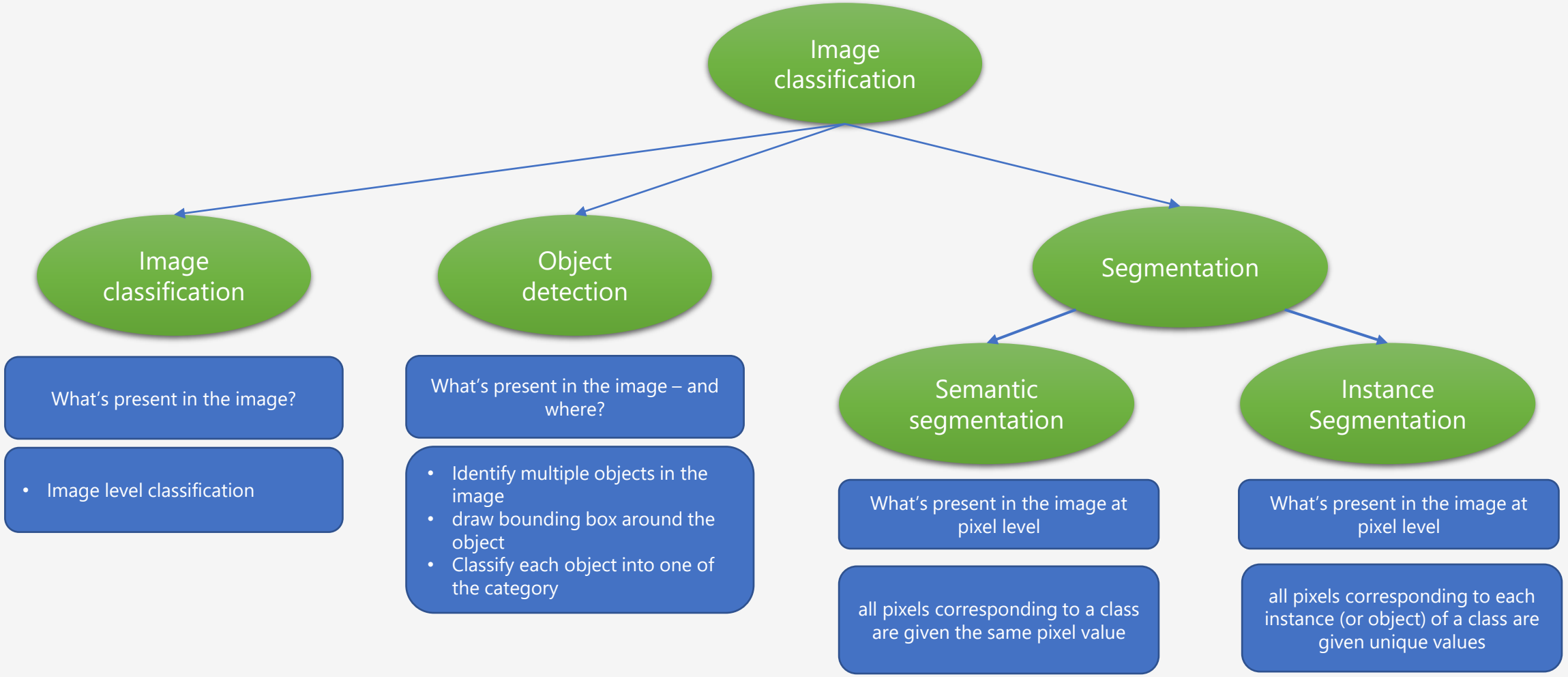
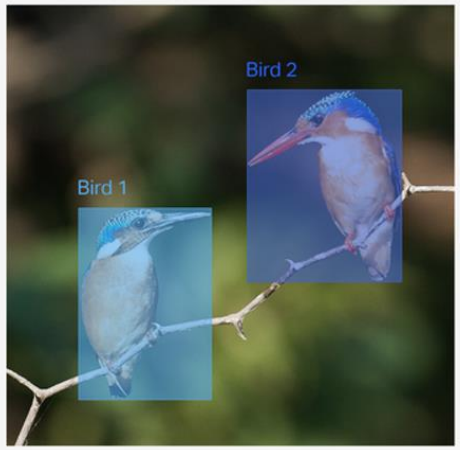


Image classification
Image classification
What's present in the image?
• Image level classification
Object detection
What's present in the image – and where?
• Identify multiple objects in the image
• draw bounding box around the object
• Classify each object into one of the category
Segmentation
Semantic segmentation
What's present in the image at pixel level
all pixels corresponding to a class are given the same pixel value
Instance Segmentation
What's present in the image at pixel level
all pixels corresponding to each instance (or object) of a class are given unique values

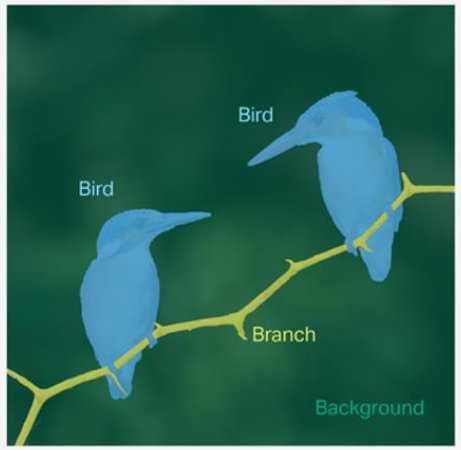
Major Types

Classification

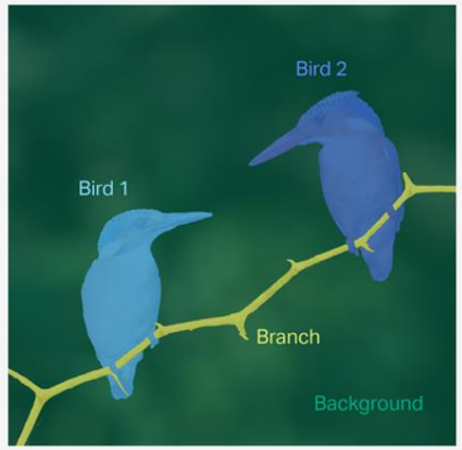
Detection



Semantic segmentation

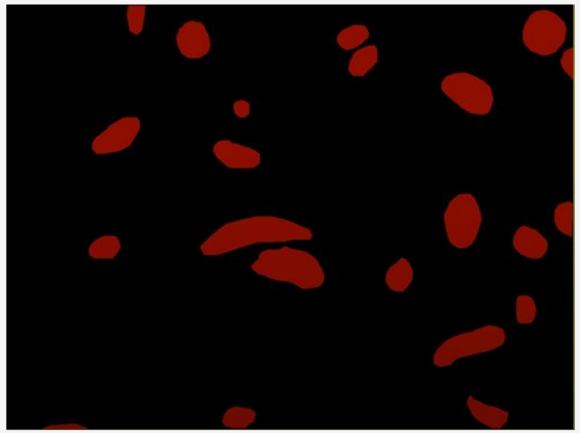


Instance segmentation

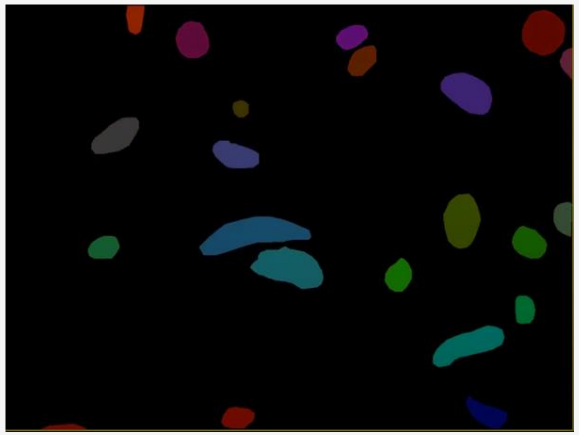


Original image

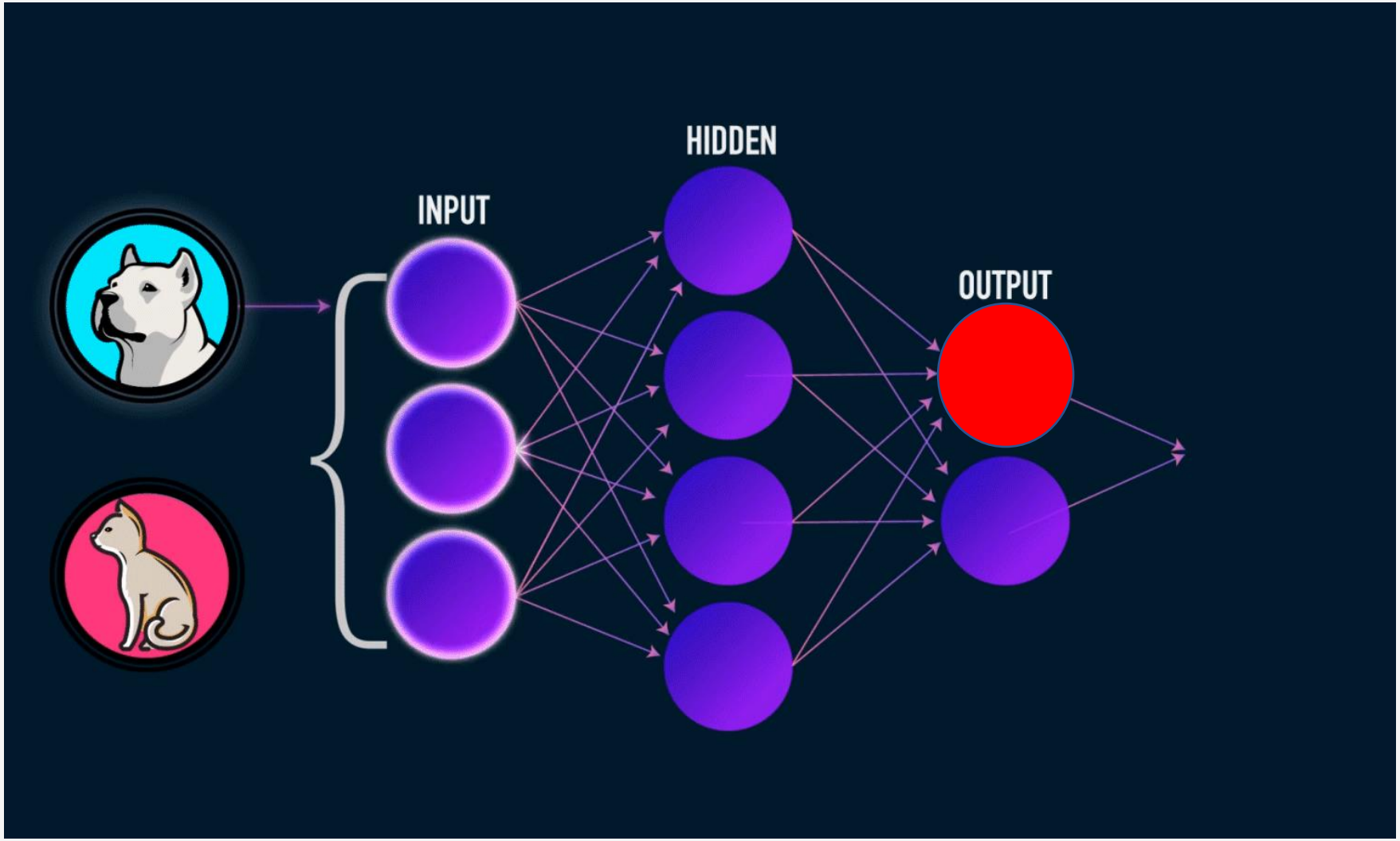
Semantic segmentation



Instance segmentation

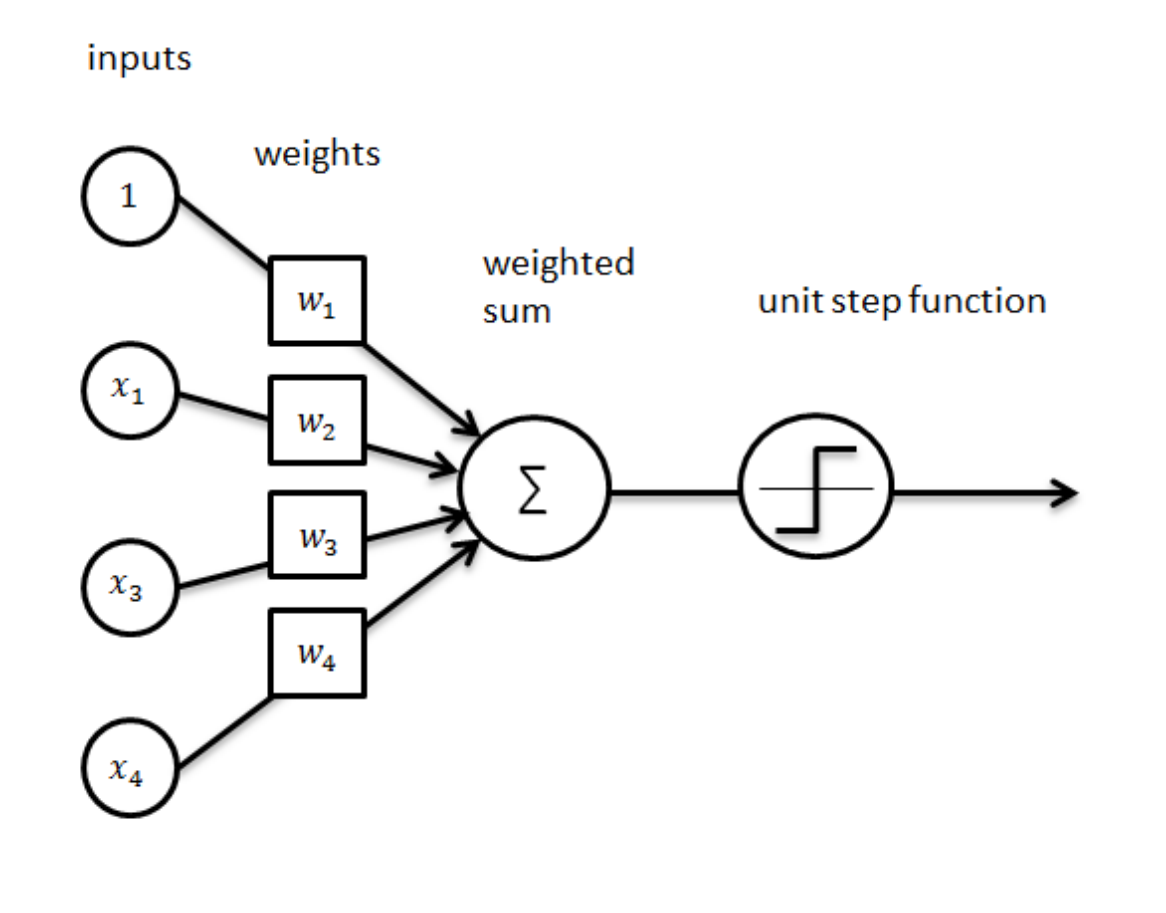


Artificial Neural Network



Source: https://www.towardsdatascience.com

Structure of the Neuron

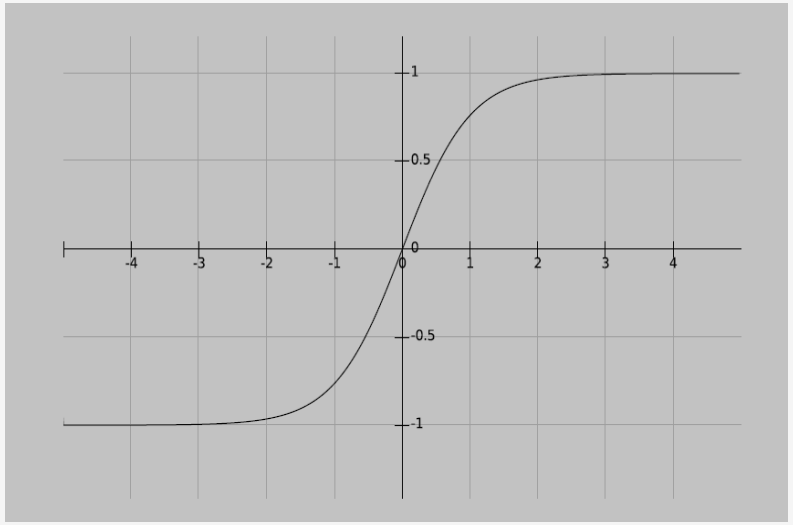


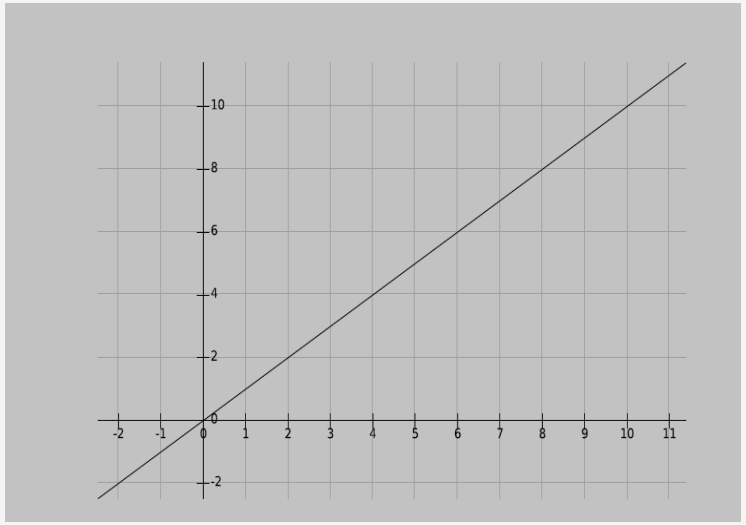
$$z = g(\sum x * w)$$

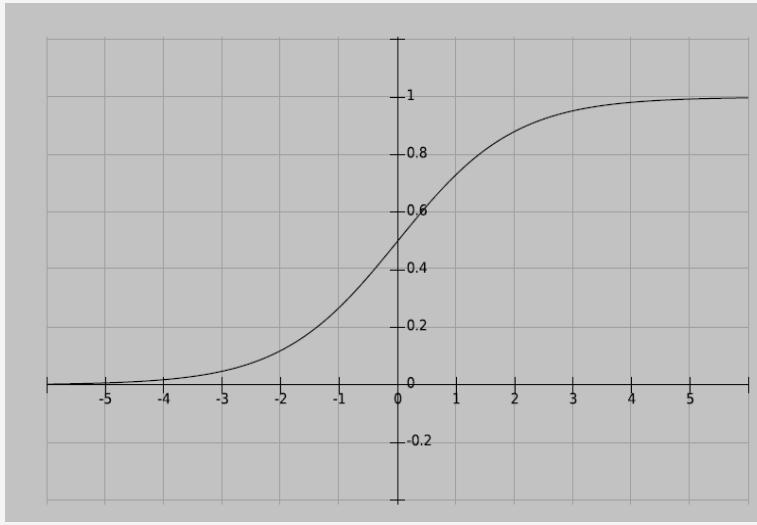
Activation functions

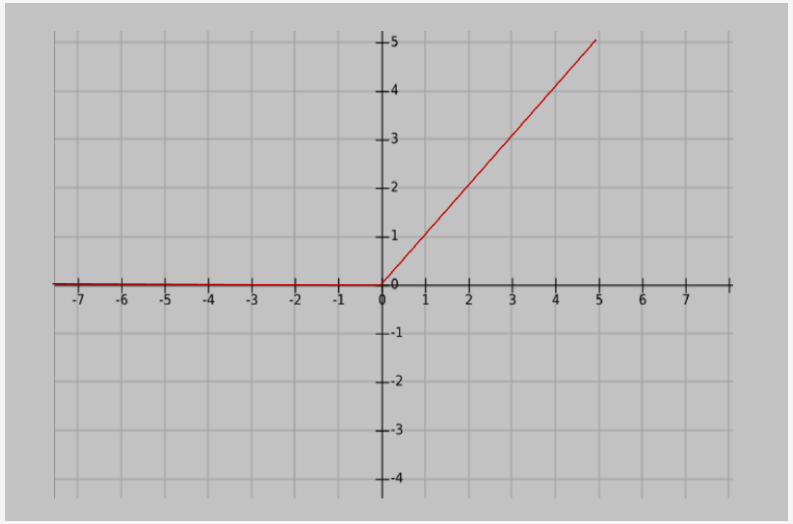
- Defines if neuron is activated or not
- Adds non linearity

- Linear
- Sigmoid
- Tanh
- ReLU

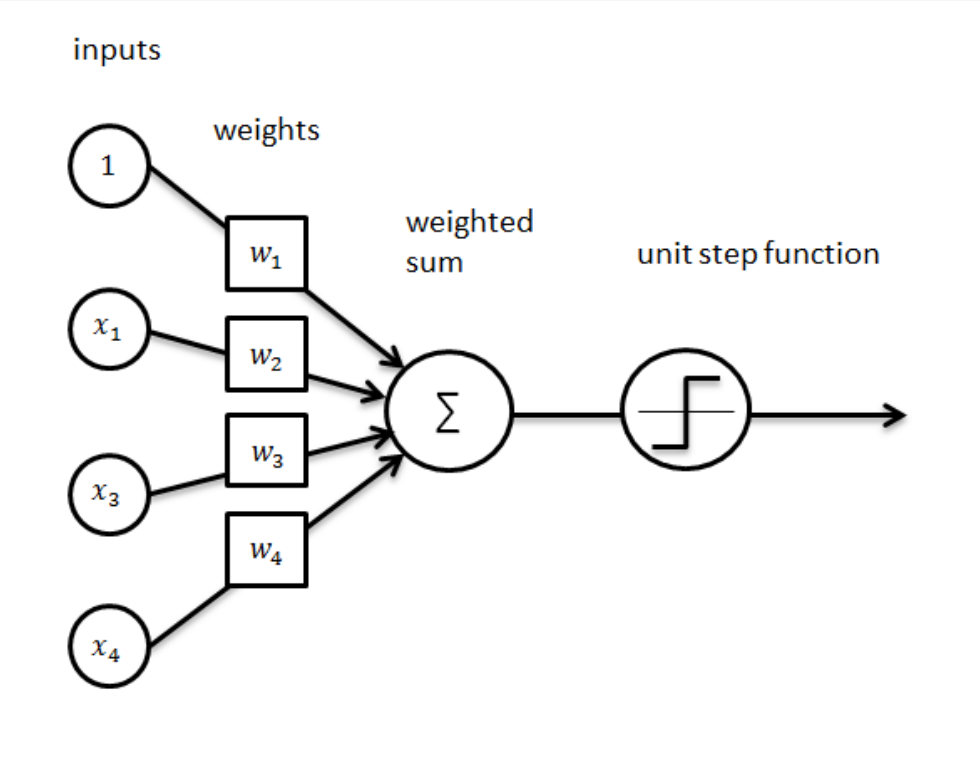








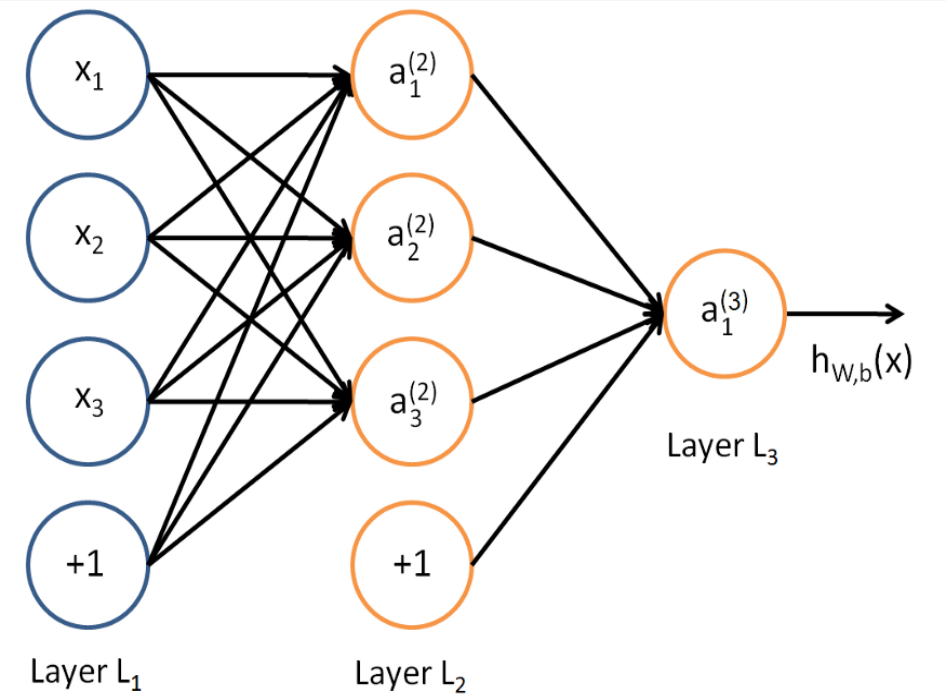
Structure of the Neuron



$$z = g(\sum x * w)$$

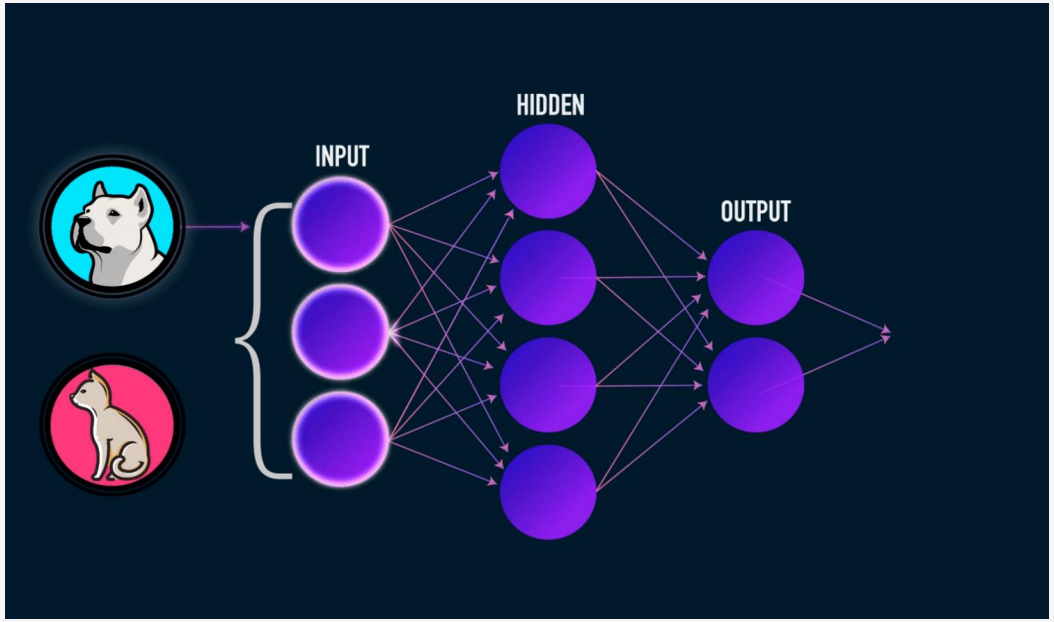
```python
class MyDenseLayer(tf.keras.layers.Layer):
  def __init__(self, input_dim, output_dim):
    super(MyDenseLayer, self).__init__()

    # Initialize weights and bias
    self.W = self.add_weight([input_dim, output_dim])
    self.b = self.add_weight([1, output_dim])

  def call(self, inputs):
    # Forward propagate the inputs
    z = tf.matmul(inputs, self.W) + self.b

    # Feed through a non-linear activation
    output = tf.math.sigmoid(z)

    return output
```

Structure of the Neuron




```python
class MyDenseLayer(tf.keras.layers.Layer):
  def __init__(self, input_dim, output_dim):
    super(MyDenseLayer, self).__init__()

    # Initialize weights and bias
    self.W = self.add_weight([input_dim, output_dim])
    self.b = self.add_weight([1, output_dim])

  def call(self, inputs):
    # Forward propagate the inputs
    z = tf.matmul(inputs, self.W) + self.b

    # Feed through a non-linear activation
    output = tf.math.sigmoid(z)

    return output
```

Neural Network

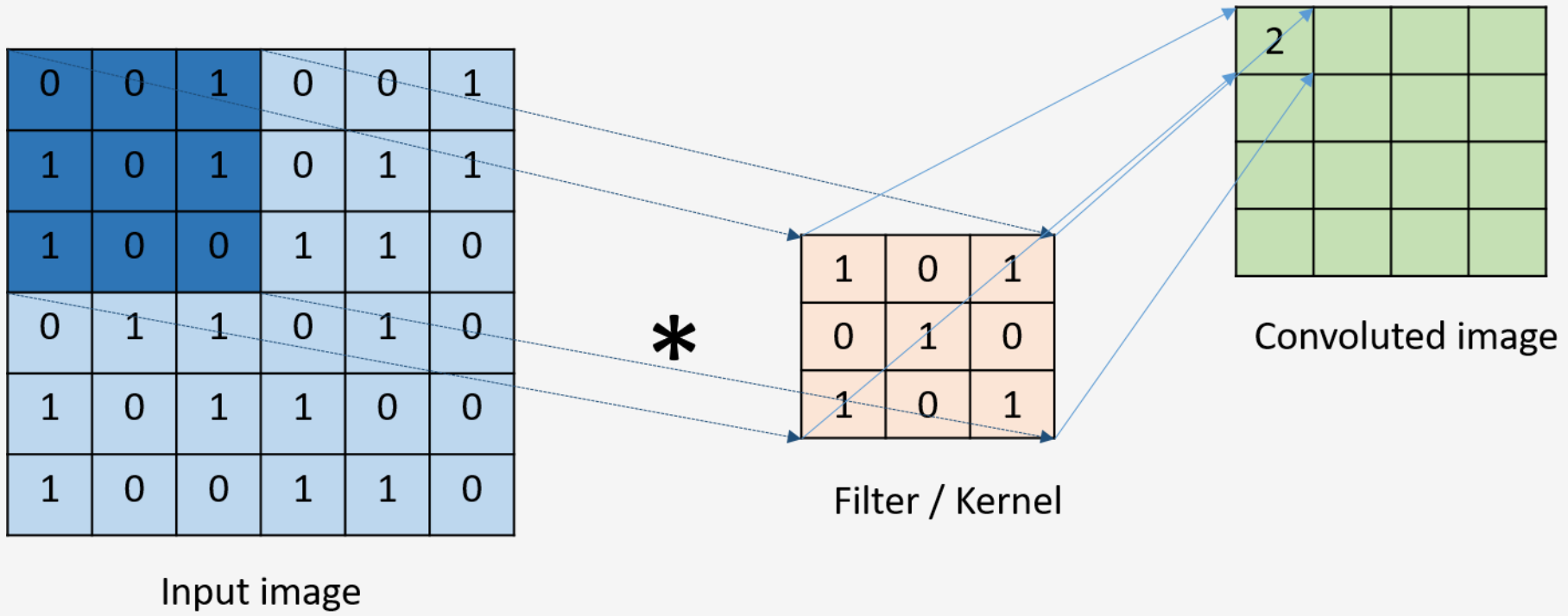



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_14 (Dense)             (None, 4)                 16
_________________________________________________________________
dense_15 (Dense)             (None, 2)                 10
=================================================================
Total params: 26
Trainable params: 26
Non-trainable params: 0
_________________________________________________________________
```

```python
import tensorflow as tf

model = tf.keras.Sequential(
    [
        tf.keras.layers.Dense(4, input_shape=(3,), activation='relu'),
        tf.keras.layers.Dense(2, activation='sigmoid')
    ]
)
```

Types of layers

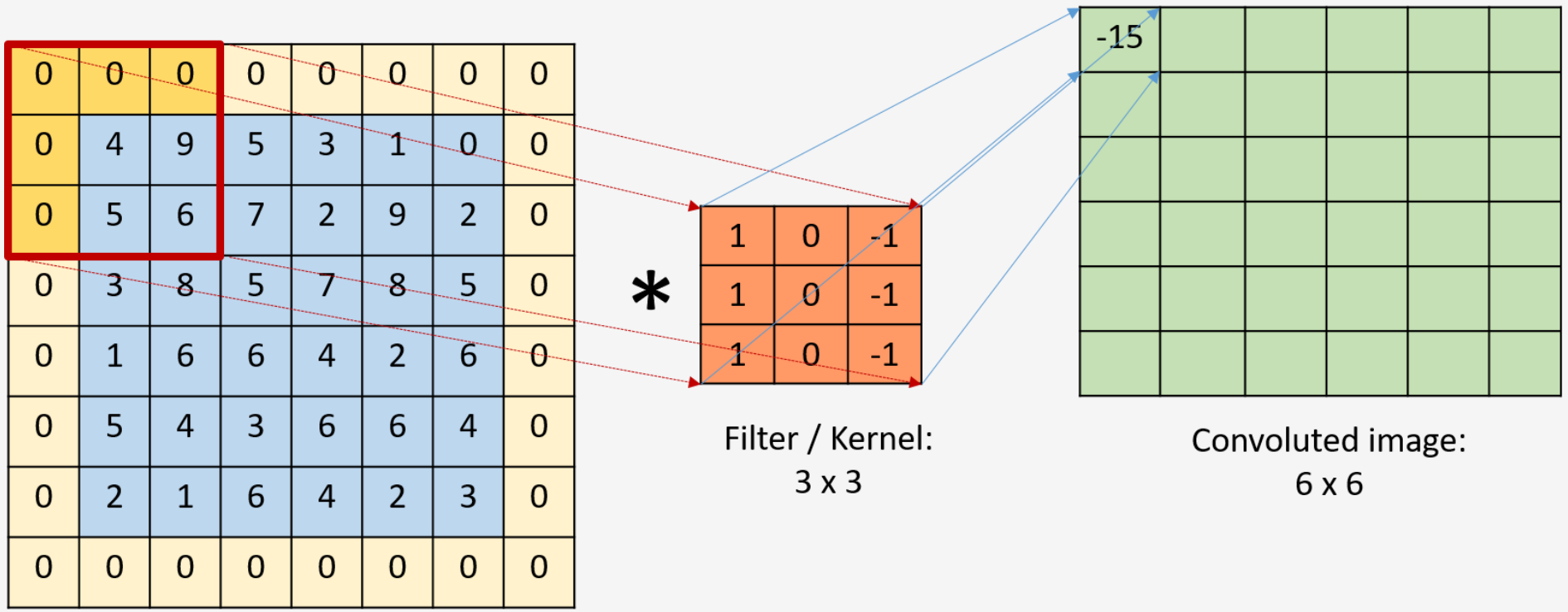
Convolution:



Change of dimension with convolution operation
Image size: $[n \times n]$
Filter size: $[f \times f]$
Result image size: $[(n - f + 1) \times (n - f + 1)]$

Types of layers

Padded convolution:



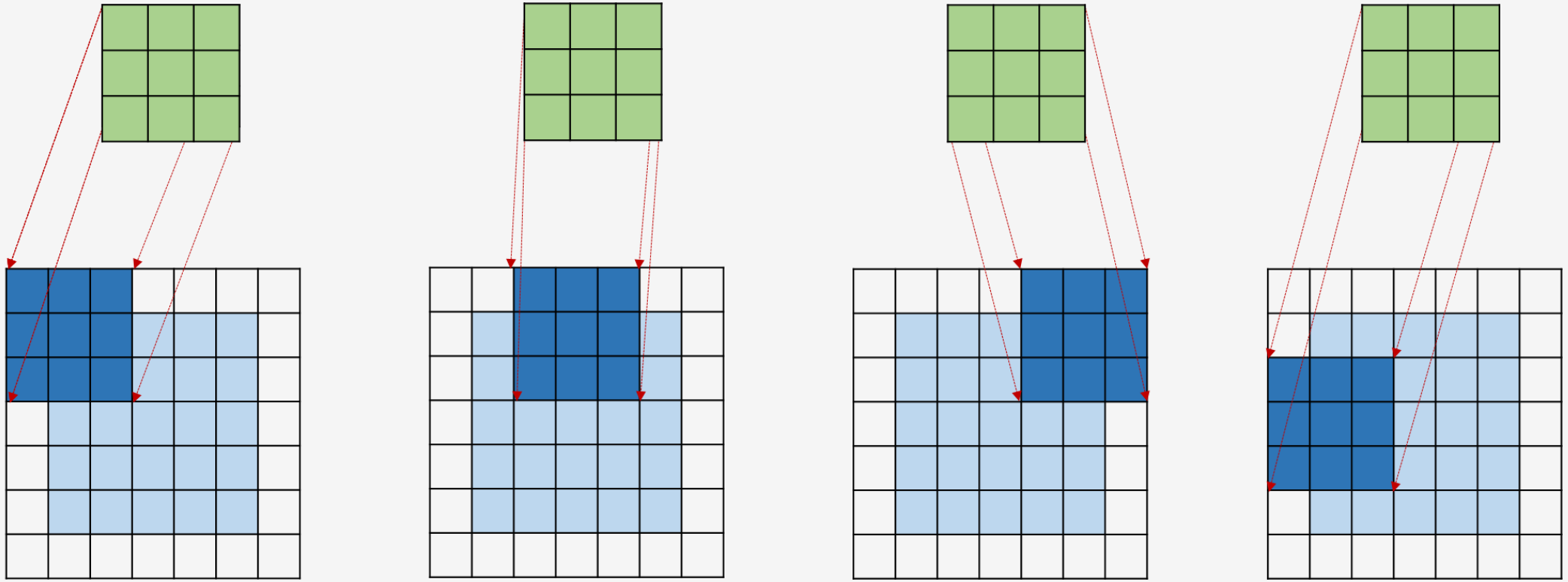
Input image : 6 x 6
Padding: 1

Filter / Kernel:
3 x 3

Convoluted image:
6 x 6

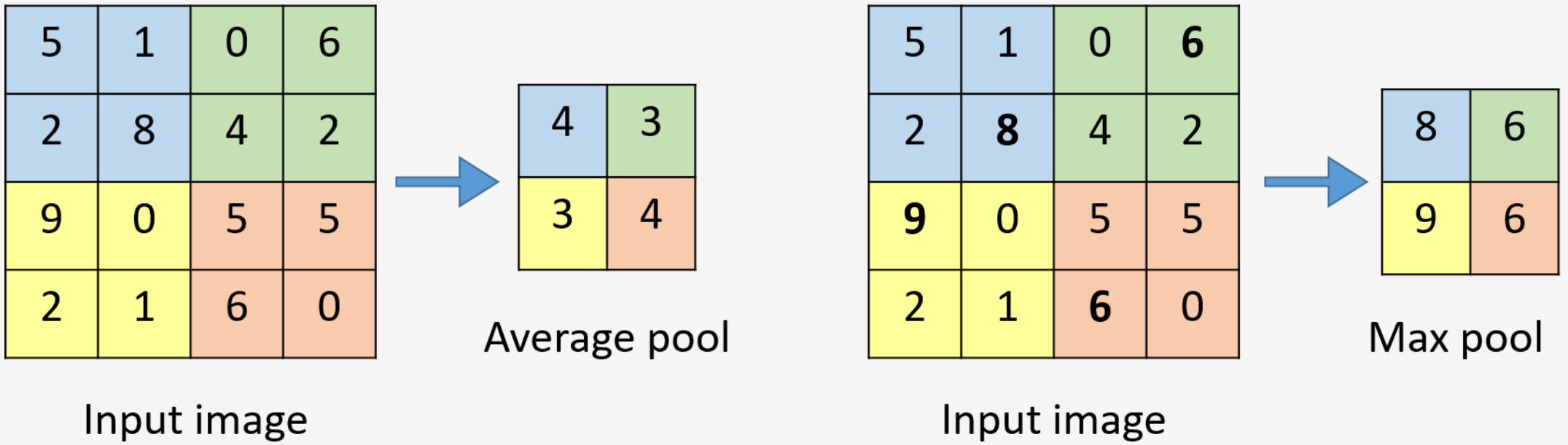
Types of layers

Strided convolution:



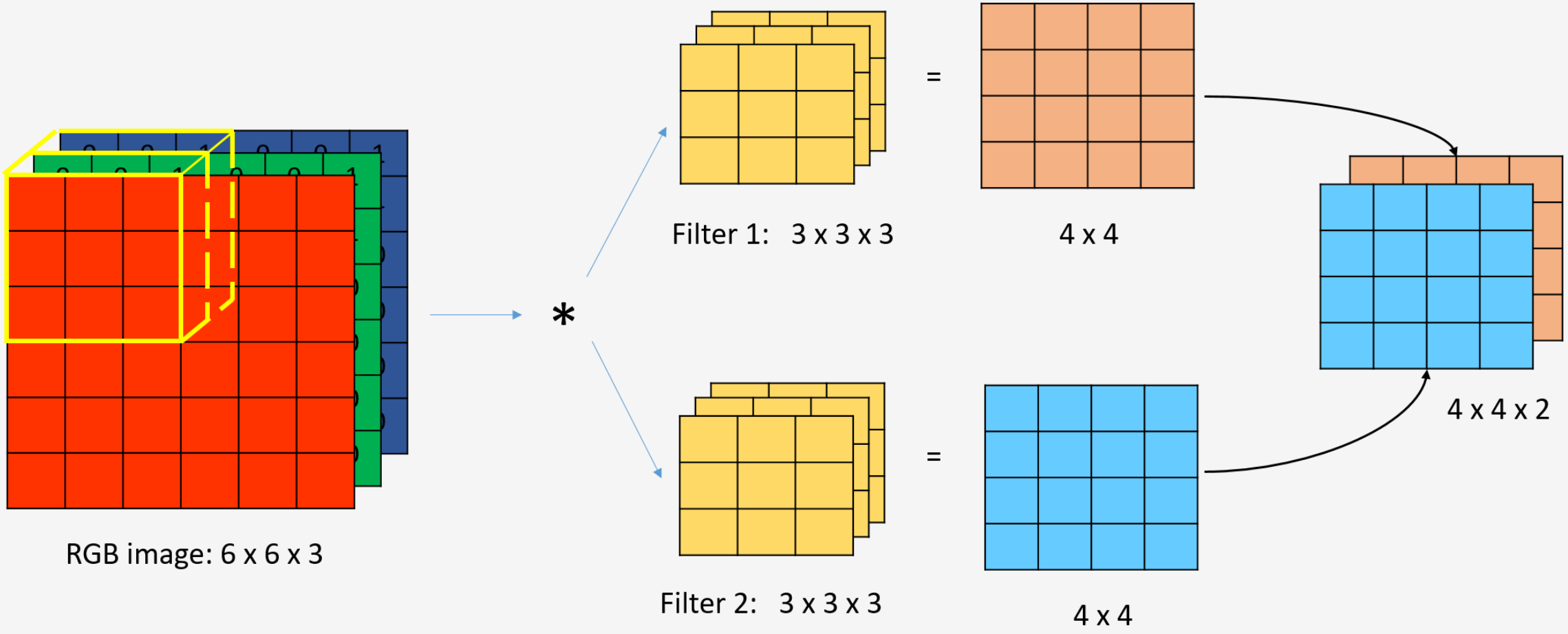
Types of layers

Pooling layer:



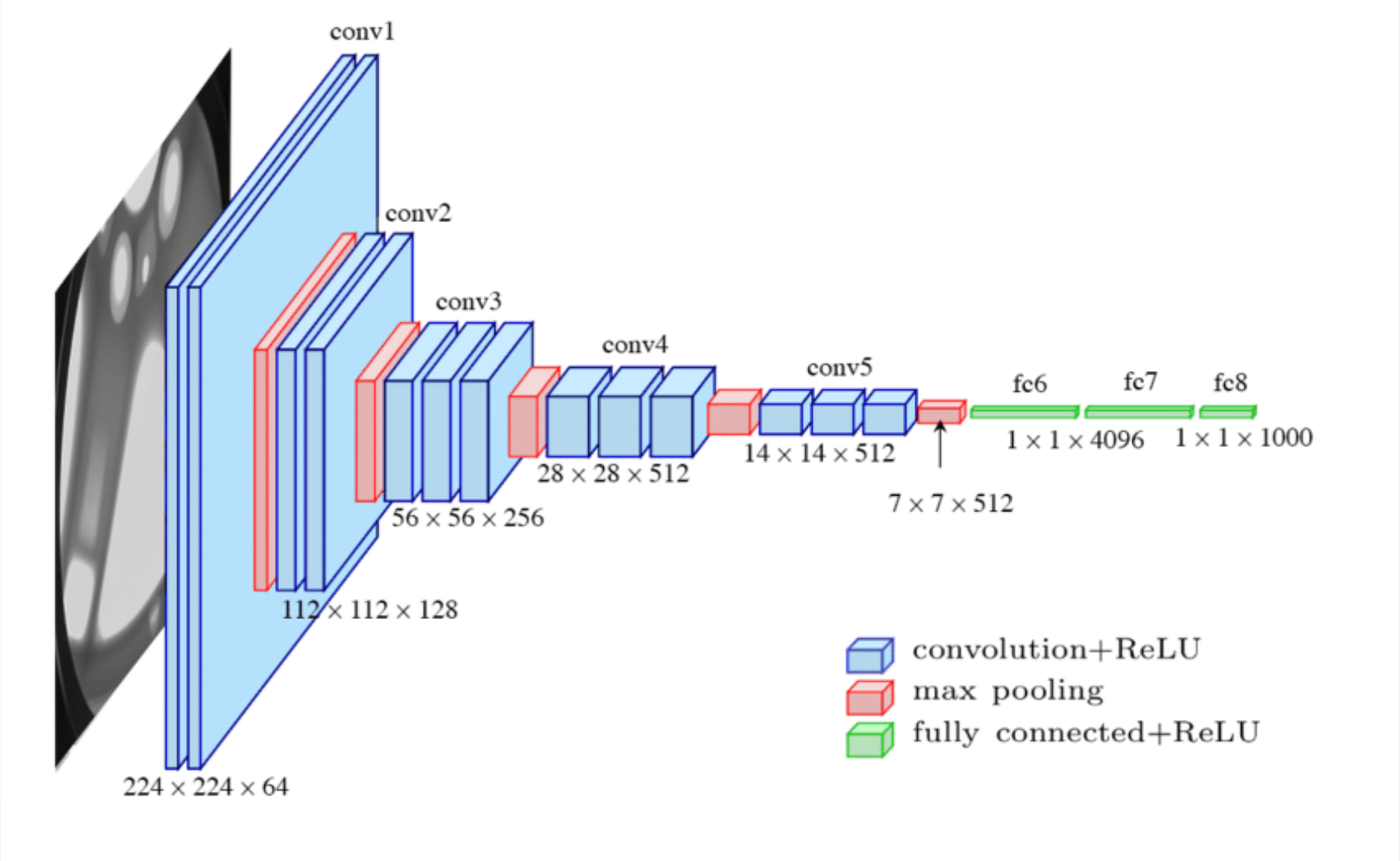
Convolution Neural Network

3D Convolution



Convolution Neural Network

VGG16 network

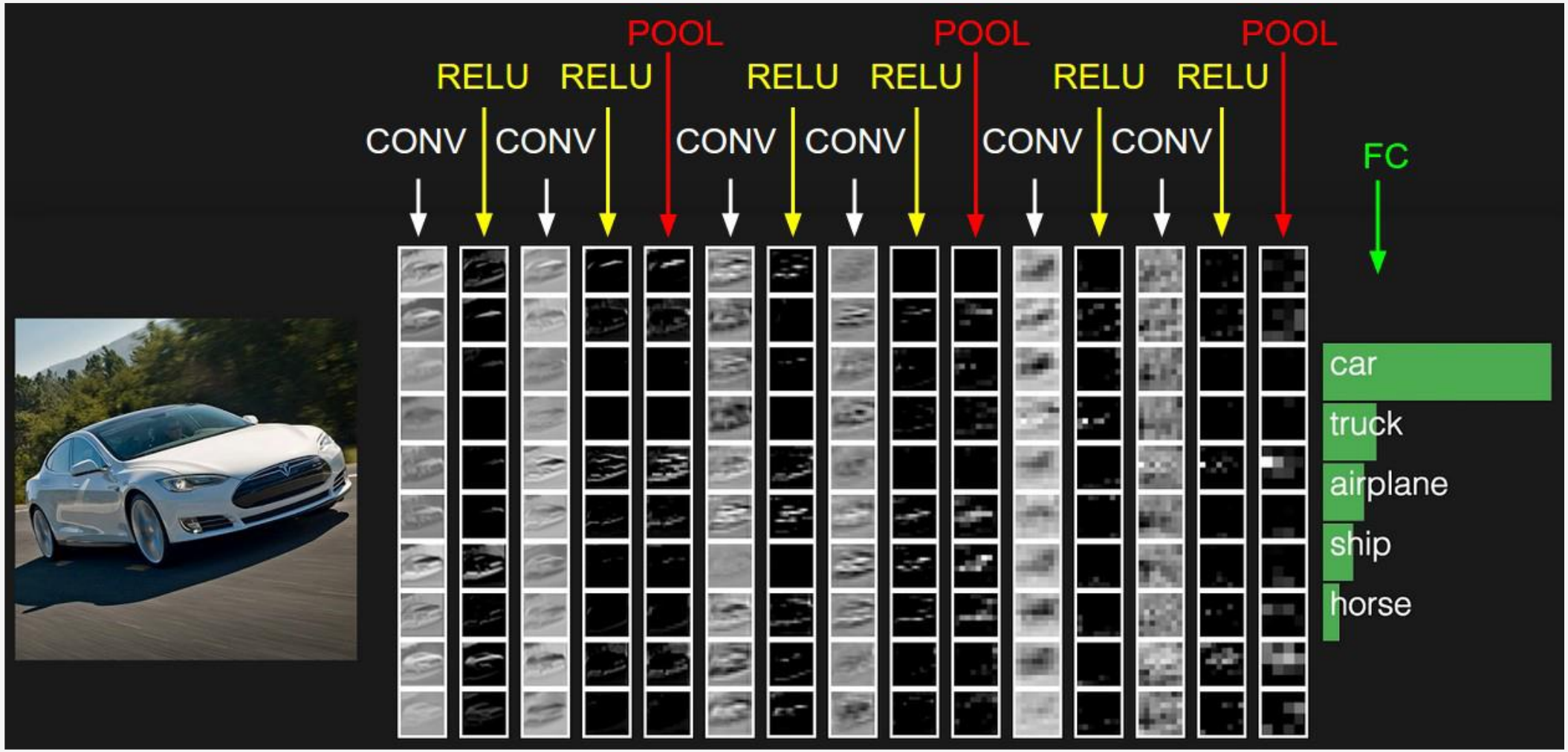


Convolution Neural Network

VGG16 network implementation in python

Convolution Neural Network

Visualization of activations in CNN



Questions?