# Summer Course

Deep Learning (DL)

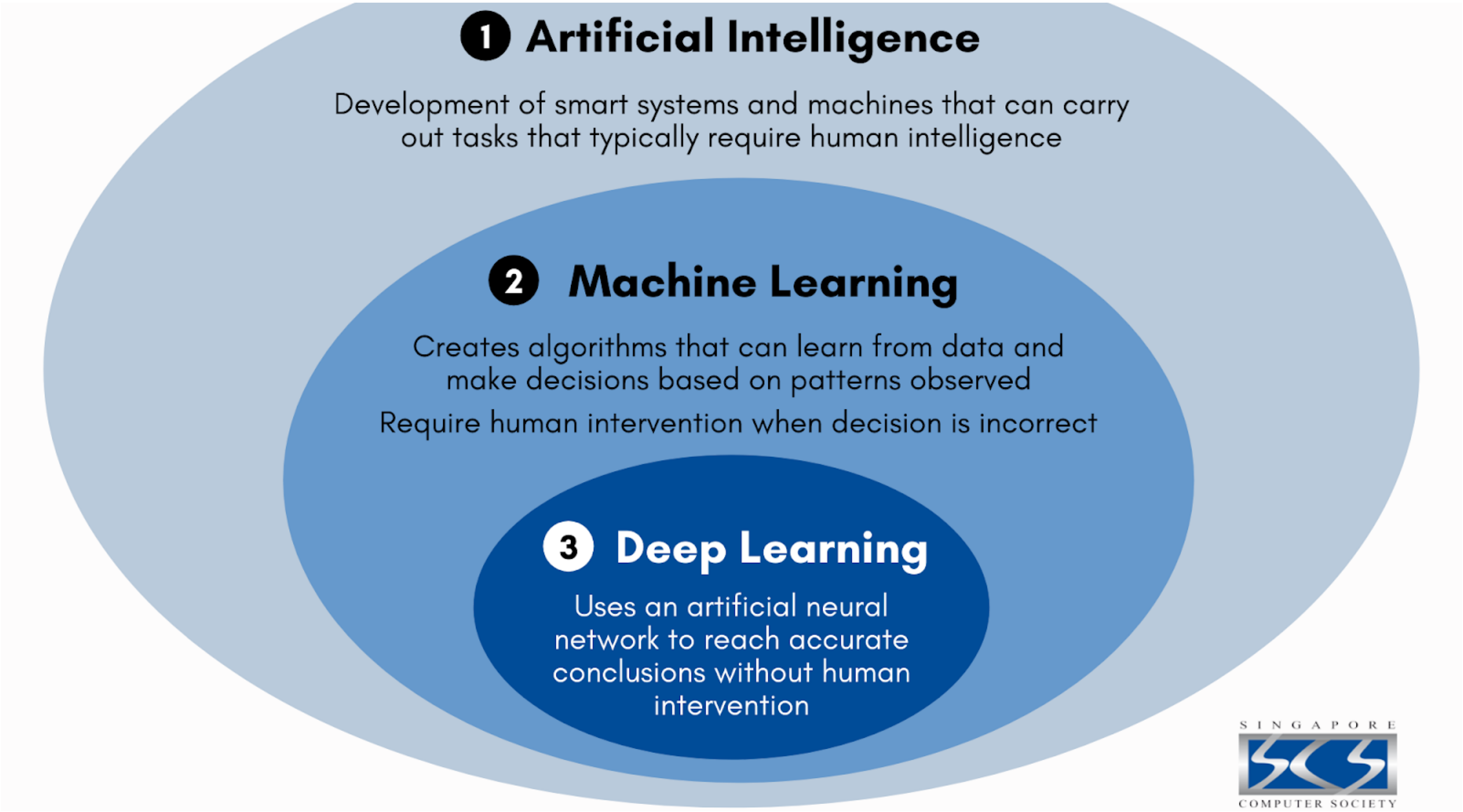
Lu,HsinLing– 8/7

# Agenda

- Machine Learning & Deep Learning
- Neural Networks
- Object Recognition and Detection

# Machine Learning & Deep Learning

## AI vs. ML vs. DL



**1 Artificial Intelligence**

Development of smart systems and machines that can carry out tasks that typically require human intelligence

**2 Machine Learning**

Creates algorithms that can learn from data and make decisions based on patterns observed

Require human intervention when decision is incorrect

**3 Deep Learning**

Uses an artificial neural network to reach accurate conclusions without human intervention

# Machine Learning & Deep Learning

## What is Machine Learning?

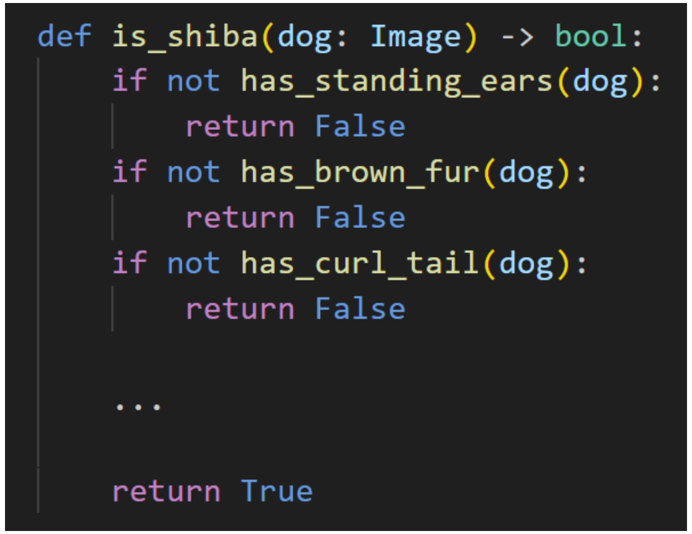
- What is “learning”?
  - Human learns from observations (experience)
  - 三折肱而成良醫 – learning from past experience.
  - 類推適用 – from 1 + 1 = 2 to 1 + 1 + 1 = 3

- How could machine “learn” something?
  - Traditional artificial intelligent: Expert system, Case-based Reasoning
  - Statistics: Probability Estimation (Classification), Regression

# Machine Learning & Deep Learning

## How do we recognize things?

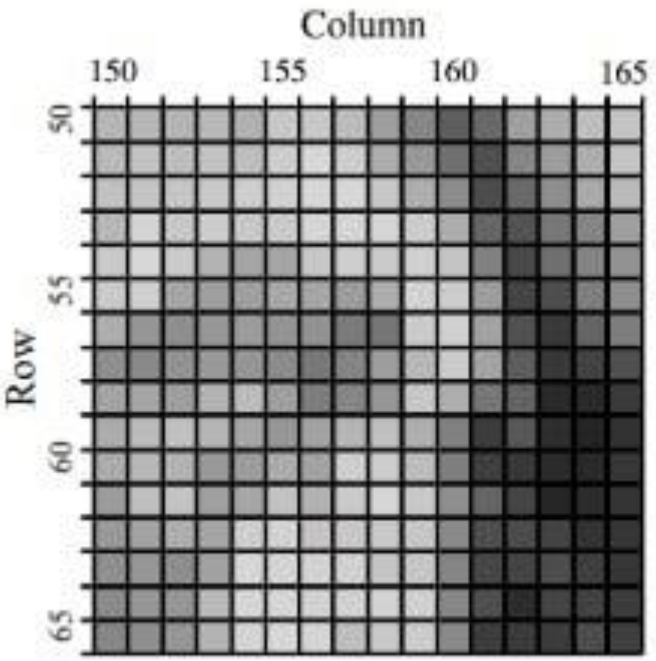
- Some visual clues (features)
  - Two standing ears
  - Brown fur
  - Curl tail

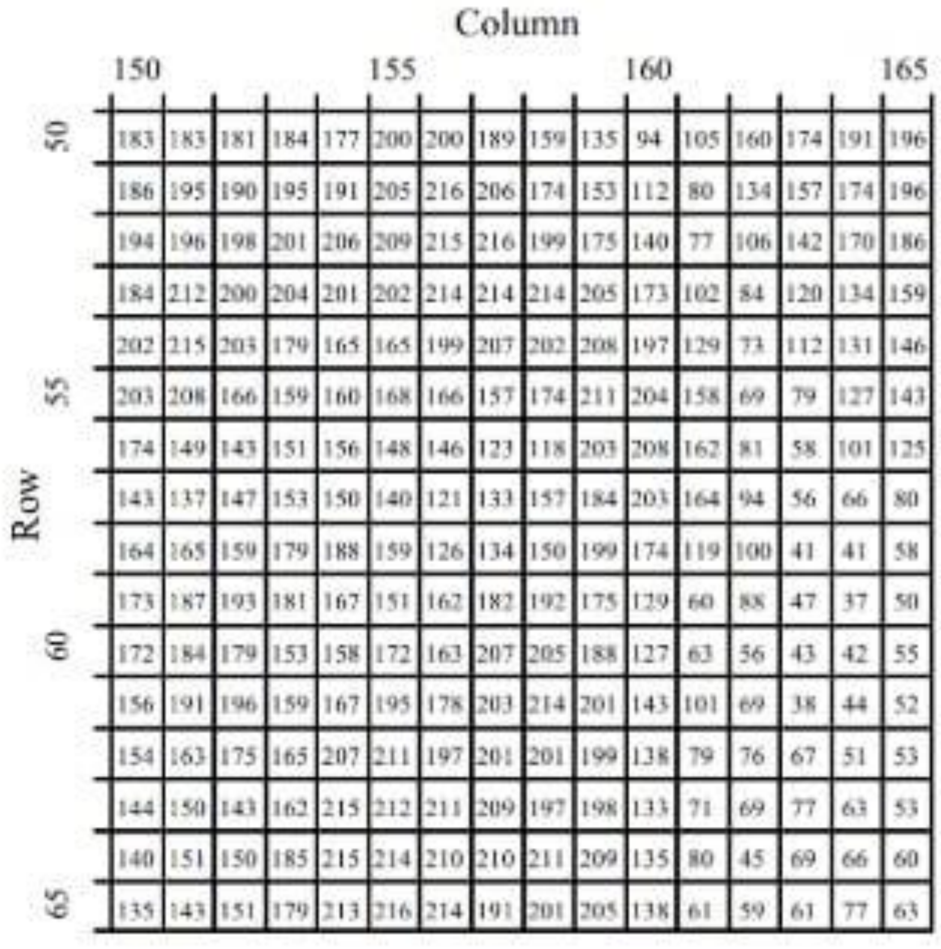


```python
def is_shiba(dog: Image) -> bool:
    if not has_standing_ears(dog):
        return False
    if not has_brown_fur(dog):
        return False
    if not has_curl_tail(dog):
        return False

    ...

    return True
```

# Machine Learning & Deep Learning

## Digital images are just numbers

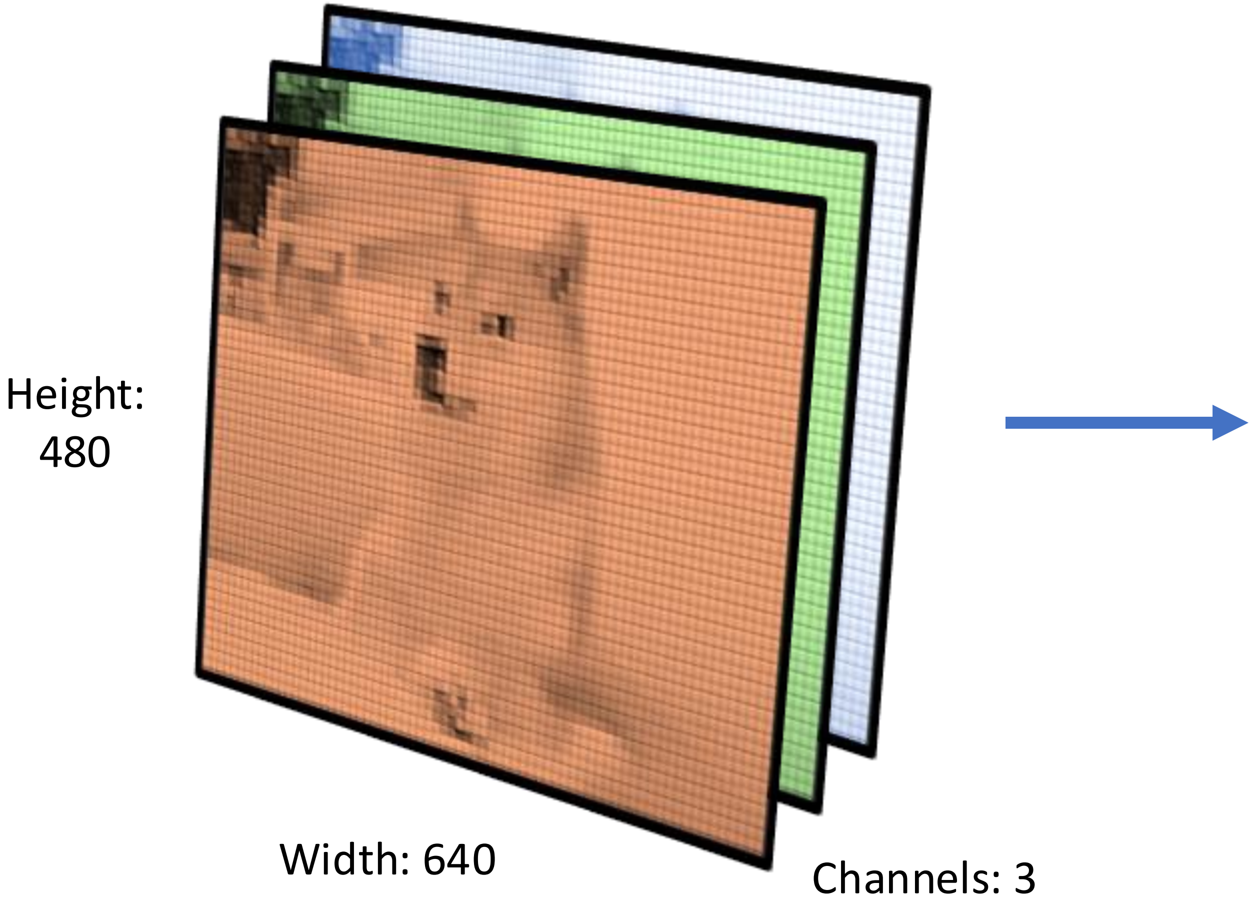



Column

| Row | 150 | | | | | 155 | | | | | 160 | | | | | 165 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 50 | 183 | 183 | 181 | 184 | 177 | 200 | 200 | 189 | 159 | 135 | 94 | 105 | 160 | 174 | 191 | 196 |
| | 186 | 195 | 190 | 195 | 191 | 205 | 216 | 206 | 174 | 153 | 112 | 80 | 134 | 157 | 174 | 196 |
| | 194 | 196 | 198 | 201 | 206 | 209 | 215 | 216 | 199 | 175 | 140 | 77 | 106 | 142 | 170 | 186 |
| | 184 | 212 | 200 | 204 | 201 | 202 | 214 | 214 | 214 | 205 | 173 | 102 | 84 | 120 | 134 | 159 |
| | 202 | 215 | 203 | 179 | 165 | 165 | 199 | 207 | 202 | 208 | 197 | 129 | 73 | 112 | 131 | 146 |
| 55 | 203 | 208 | 166 | 159 | 160 | 168 | 166 | 157 | 174 | 211 | 204 | 158 | 69 | 79 | 127 | 143 |
| | 174 | 149 | 143 | 151 | 156 | 148 | 146 | 123 | 118 | 203 | 208 | 162 | 81 | 58 | 101 | 125 |
| | 143 | 137 | 147 | 153 | 150 | 140 | 121 | 133 | 157 | 184 | 203 | 164 | 94 | 56 | 66 | 80 |
| | 164 | 165 | 159 | 179 | 188 | 159 | 126 | 134 | 150 | 199 | 174 | 119 | 100 | 41 | 41 | 58 |
| | 173 | 187 | 193 | 181 | 167 | 151 | 162 | 182 | 192 | 175 | 129 | 60 | 88 | 47 | 37 | 50 |
| 60 | 172 | 184 | 179 | 153 | 158 | 172 | 163 | 207 | 205 | 188 | 127 | 63 | 56 | 43 | 42 | 55 |
| | 156 | 191 | 196 | 159 | 167 | 195 | 178 | 203 | 214 | 201 | 143 | 101 | 69 | 38 | 44 | 52 |
| | 154 | 163 | 175 | 165 | 207 | 211 | 197 | 201 | 201 | 199 | 138 | 79 | 76 | 67 | 51 | 53 |
| | 144 | 150 | 143 | 162 | 215 | 212 | 211 | 209 | 197 | 198 | 133 | 71 | 69 | 77 | 63 | 53 |
| | 140 | 151 | 150 | 185 | 215 | 214 | 210 | 210 | 211 | 209 | 135 | 80 | 45 | 69 | 66 | 60 |
| 65 | 135 | 143 | 151 | 179 | 213 | 216 | 214 | 191 | 201 | 205 | 138 | 61 | 59 | 61 | 77 | 63 |

# Machine Learning & Deep Learning

## Digital images are just numbers



$$\begin{bmatrix} \begin{bmatrix} 255 \\ 239 \\ 102 \\ \cdot \\ \cdot \end{bmatrix} \\ \begin{bmatrix} 230 \\ 122 \\ 248 \\ \cdot \\ \cdot \end{bmatrix} \\ \cdot \\ \cdot \end{bmatrix} \in W \times H \times C$$

Which means,
this image is a 921600-D vector.

# Machine Learning & Deep Learning

## Binary classification

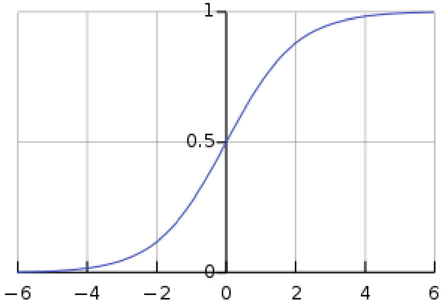
- $is\_shiba(X) = \sigma(w_1 x_1 + w_2 x_2 + \cdots + w_{921600} x_{92600})$
- $is\_shiba(X) \in [0, 1]$

- $if\ is\_shiba(X) \geq 0.5, X$ is shiba
- $if\ is\_shiba(X) < 0.5, X$ is not shiba

- Maybe a more complex model?
- $is\_shiba(X) = \sigma(w_{1,1} x_1 + w_{1,2} {x_1}^2 + w_{1,3} {x_3}^3 + \cdots)$

Sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}} \in [0, 1]$$



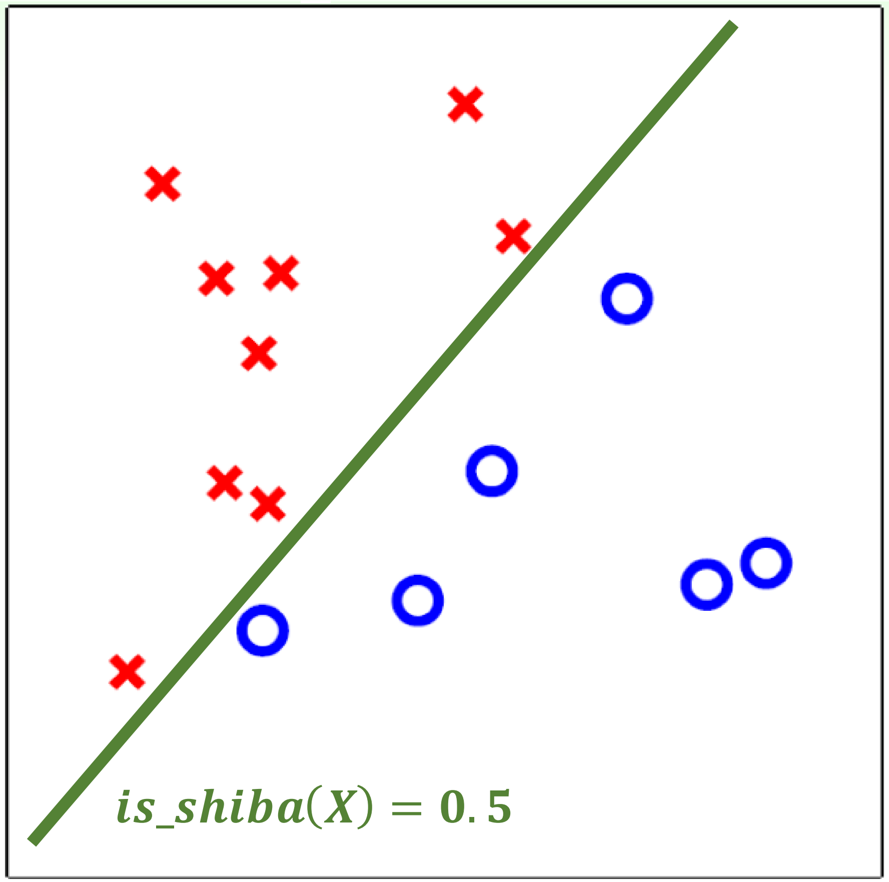
# Machine Learning & Deep Learning

## Binary classification

- Say we have n random dog images
    - Some of them are Shiba.
    - All images are 640x480 res.

      (Which means all of them are 921600-D vectors)

- We can plot these n vectors.
- $is\_shiba(X) = 0.5$ is the decision boundary.



# Machine Learning & Deep Learning

## Binary classification

- How do we find the coefficient $w$? → Learn from existing data

- Some algorithm to find $w$
  - Perceptron Learning Algorithm (PLA)
  - Support Vector Machine (SVM)
  - Adaptive Boosting (Adaboosting)
  - Decision Tree, Random Forest
  - Neural Network (NN)

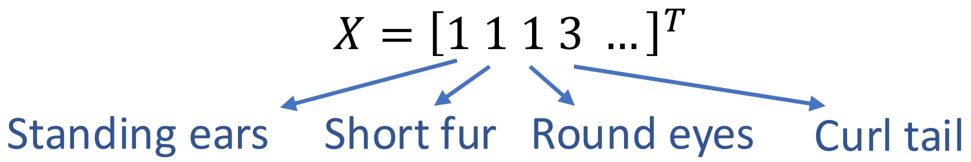
- However, this approach is not good enough (using a 921600-D vectors)

# Machine Learning & Deep Learning

## Data preprocessing & feature extraction

- Representing images with 921600-D vectors is not useful
  - Maybe not every pixel matters.
  - Pixel values (intensities) are very low-level features.
  - ~~The computation cost is to high.~~

- Can we represent the content (the dog) in better way?
  - Down sampling
  - Even some semantic information provided by human?



$$X = [1\ 1\ 1\ 3\ \ldots]^T$$

Standing ears

Short fur

Round eyes

Curl tail

# Machine Learning & Deep Learning

## Data preprocessing & feature extraction

- Data Preprocessing
  - Dimensionality Reduction
  - Normalization
  - Data Cleaning
  - Data Augmentation

- Feature Extraction
  - Relatively low-level features: Gradient, Edge, Texture intensity, etc.
  - Relatively higher-level features: Average Semantic intensity, etc.

# Machine Learning & Deep Learning

## How to perform Machine Learning?

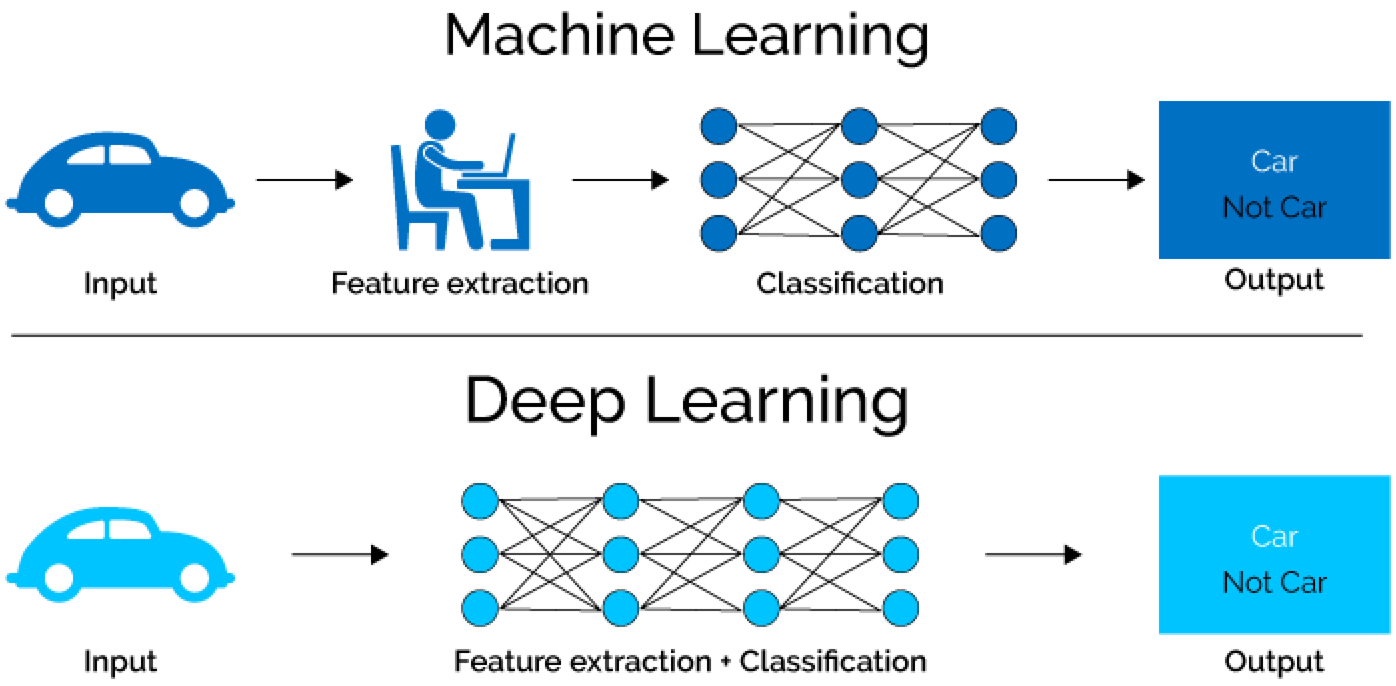
1. Prepare a dataset
   - Labeling
   - Data preprocessing

2. Choose a model
   - Perceptron, SVM, Neural Network, etc.

3. Apply to a learning algorithm
   - PLA, Linear Regression, Gradient Descending, etc.

# Machine Learning & Deep Learning

## What is Deep Learning?



# Machine Learning & Deep Learning

## How to perform Deep Learning?

1. Prepare a dataset
   - Labeling
   - ~~Data preprocessing~~ The machine also learns to perform data preprocessing.

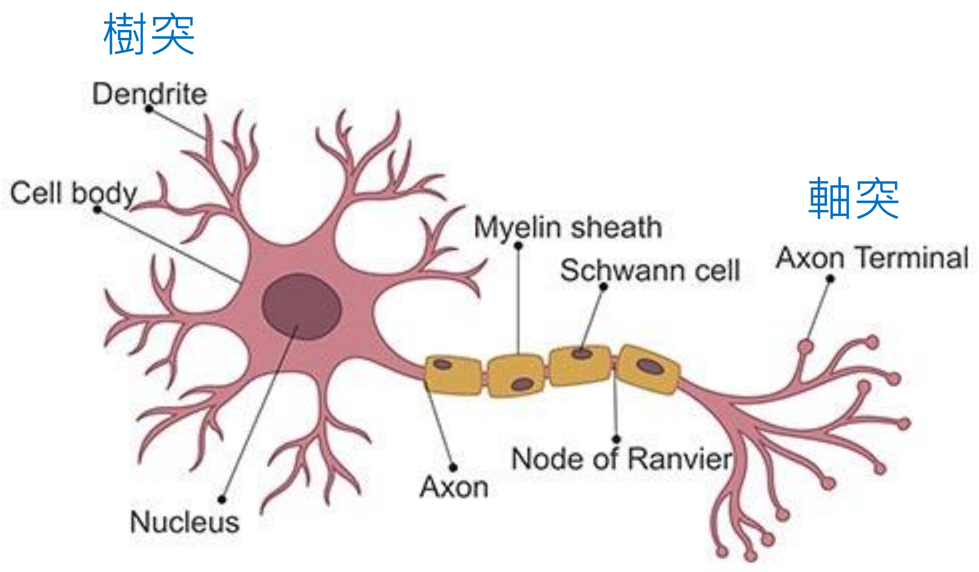
2. Choose a model
   - Neural Network (NN, CNN, RNN, LSTM, GAN, Transformer, diffusion model)

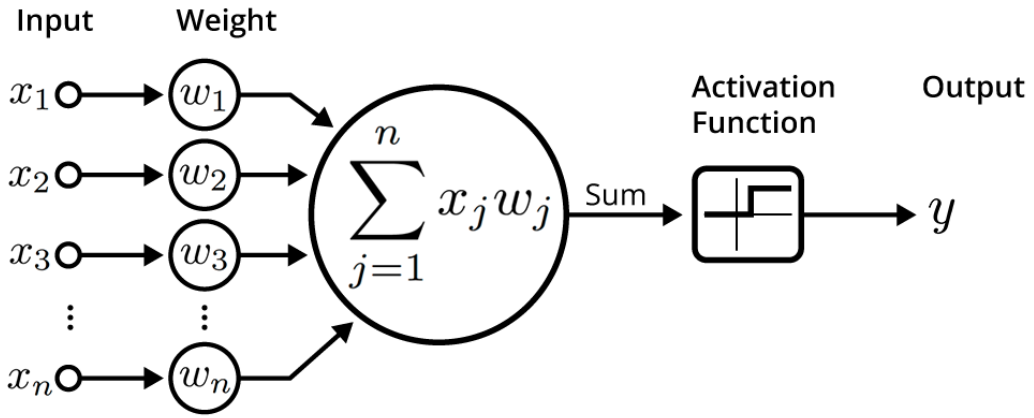
3. Apply to a learning algorithm
   - loss function(s)
   - Gradient Descending (Backpropagation)

# Neural Network

## What is Neural Network?

- Human brain has 8.6 x $10^{10}$ (eighty six billion) neurons (神經元).
- Neuron receives information and pass to others if needed (activated).
- Some scientists try to simulate human brain with computer.



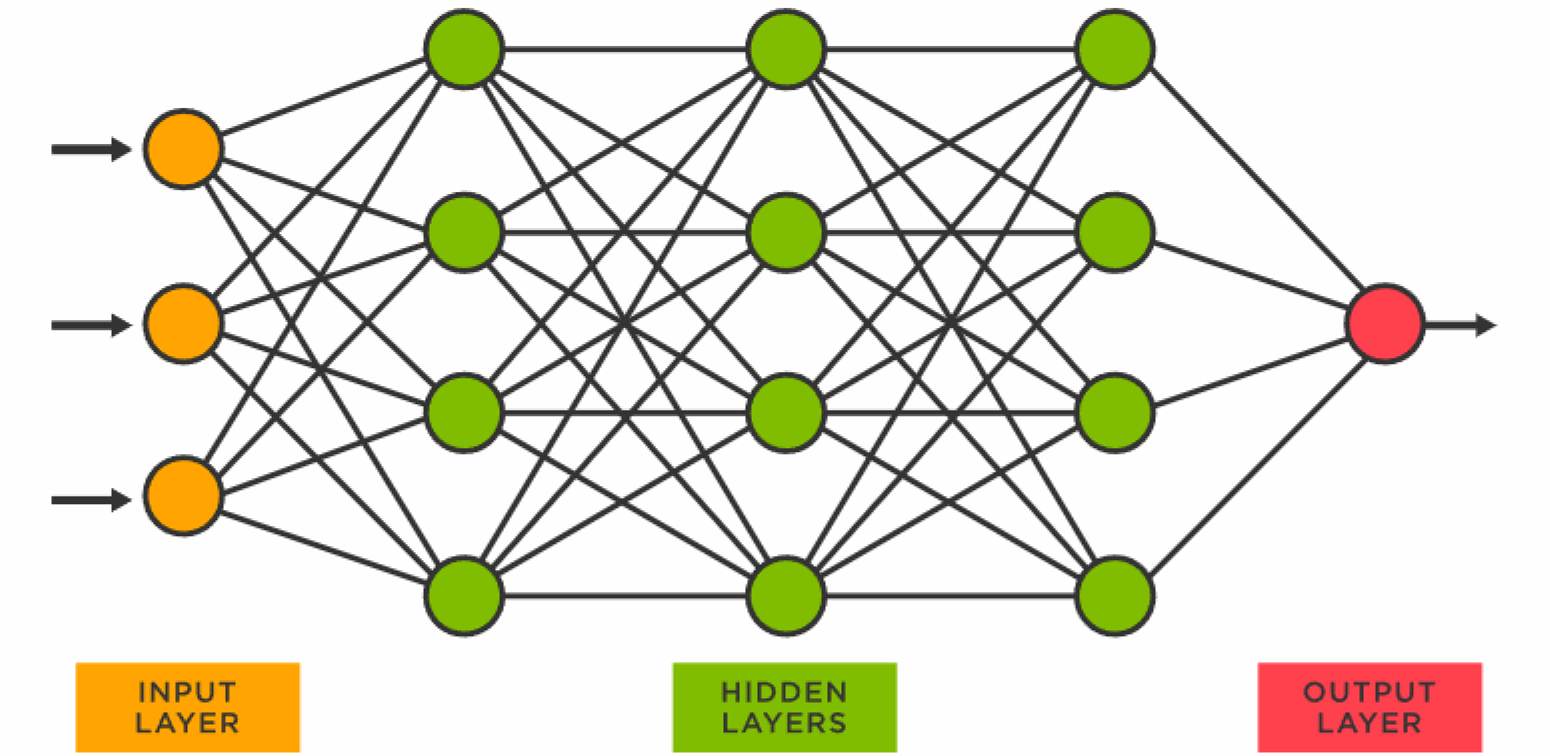


An illustration of an artificial neuron. Source: Becoming Human.

Artificial neurons are only a weighted sum followed by activation function!

# Neural Network

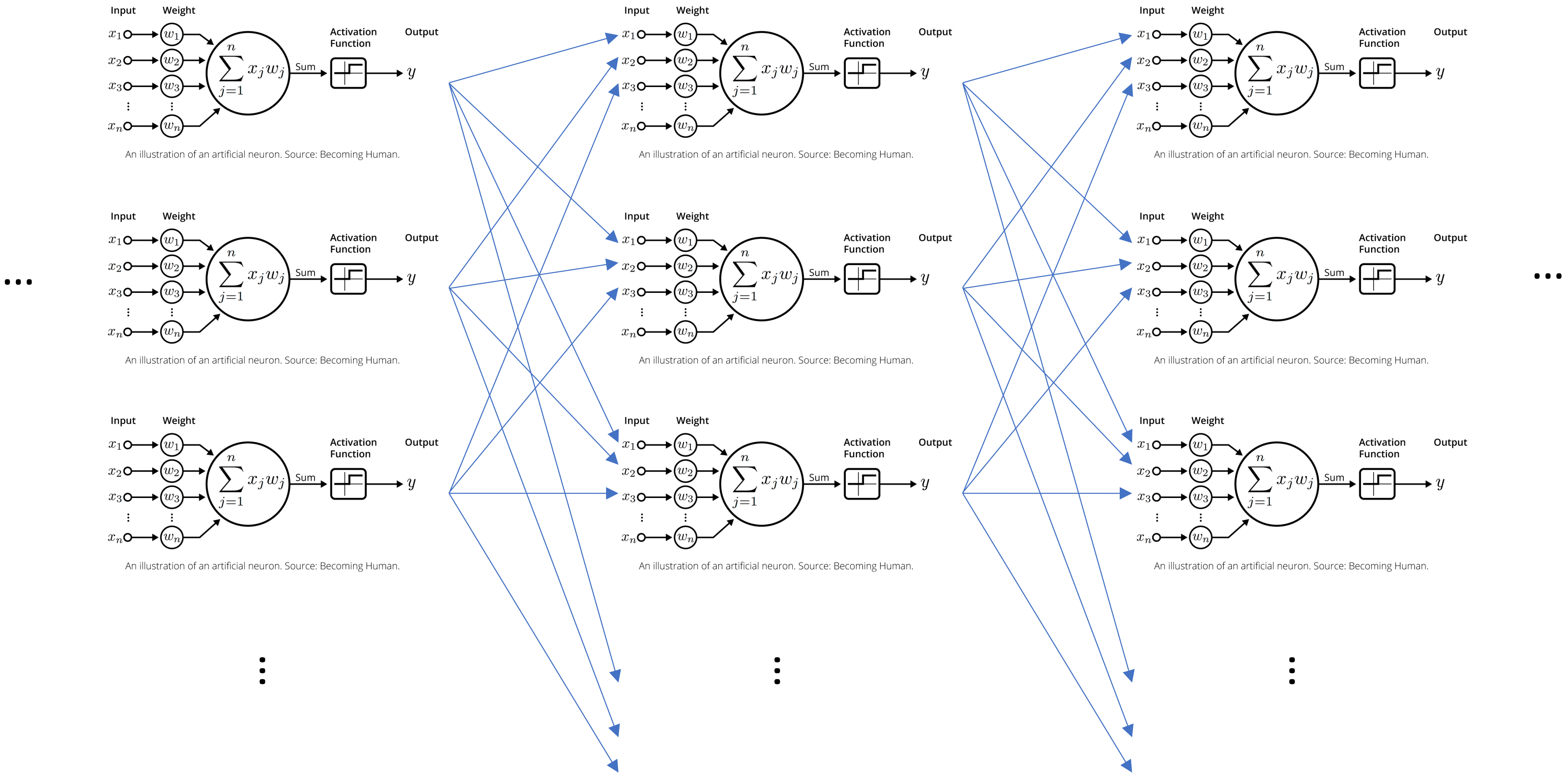
## What is Neural Network?



$$\begin{bmatrix} 0.05 \\ 0.95 \end{bmatrix}$$

# Neural Network

## How neural networks learn?



# Neural Network

## How neural networks learn?

What is Neural Netw

- Forward Propagation : The input data starts at the input layer and passes through each layer’s weighted computation, ultimately reaching the output layer.
- Compute Loss: The difference between the model’s predictions and the true labels is called the loss. A loss function measures this difference.
- Backpropagation : Based on the value of the loss function, each weight’s contribution to the loss is calculated. The backpropagation algorithm computes gradients to update weights, reducing the loss function gradually.
- Weight Update : Using the gradient descent algorithm, the weights are updated based on the computed gradients, making the model’s predictions more accurate.

# Neural Network

## How neural networks learn?

- Neurons are just weighted sum with activation function, so a neural network can be represented by a mathematical function. And, this function is differentiable.

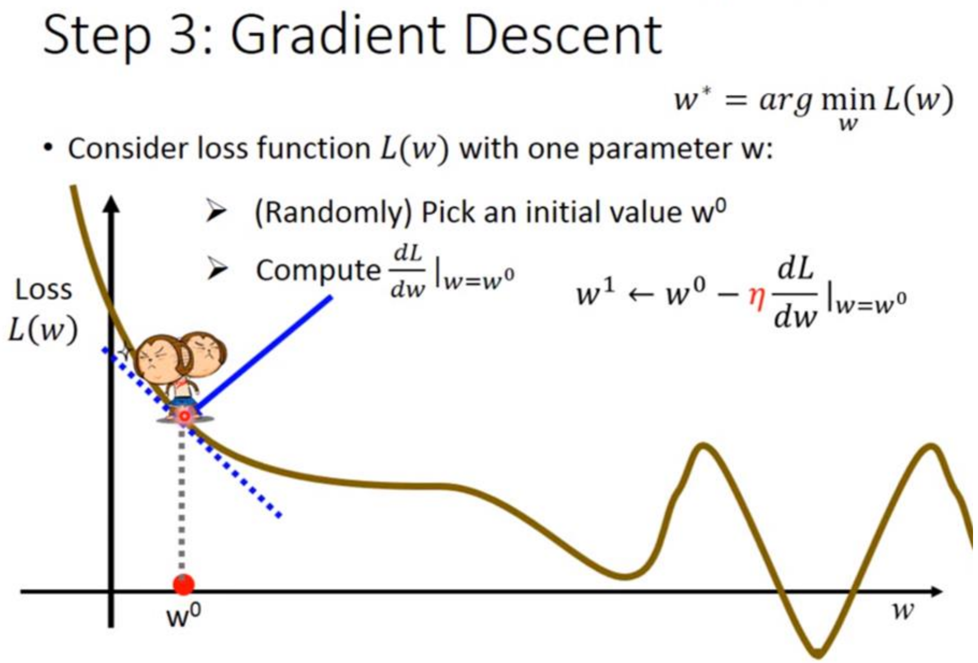
- Loss function: A way to evaluate the performance of neural networks.
  - Loss (Error): The distance between prediction and ground truth
  - Mean Absolute Error (MAE)
  - Mean Squared Error (MSE)
  - Cross Entropy (CE)
  - Focal Loss
  - Dice Loss

# Neural Network

## How neural networks learn?

- We can plot the parameter space with a simple curve.
- All we want is to find the global minimum of loss.
- Because neural networks are differentiable, we can calculate the gradient of current parameters with given loss.
- We modify the weights toward negative gradient direction, so the loss will decrease ideally.

http://chico386.pixnet.net/album/photo/171572850



Step 3: Gradient Descent

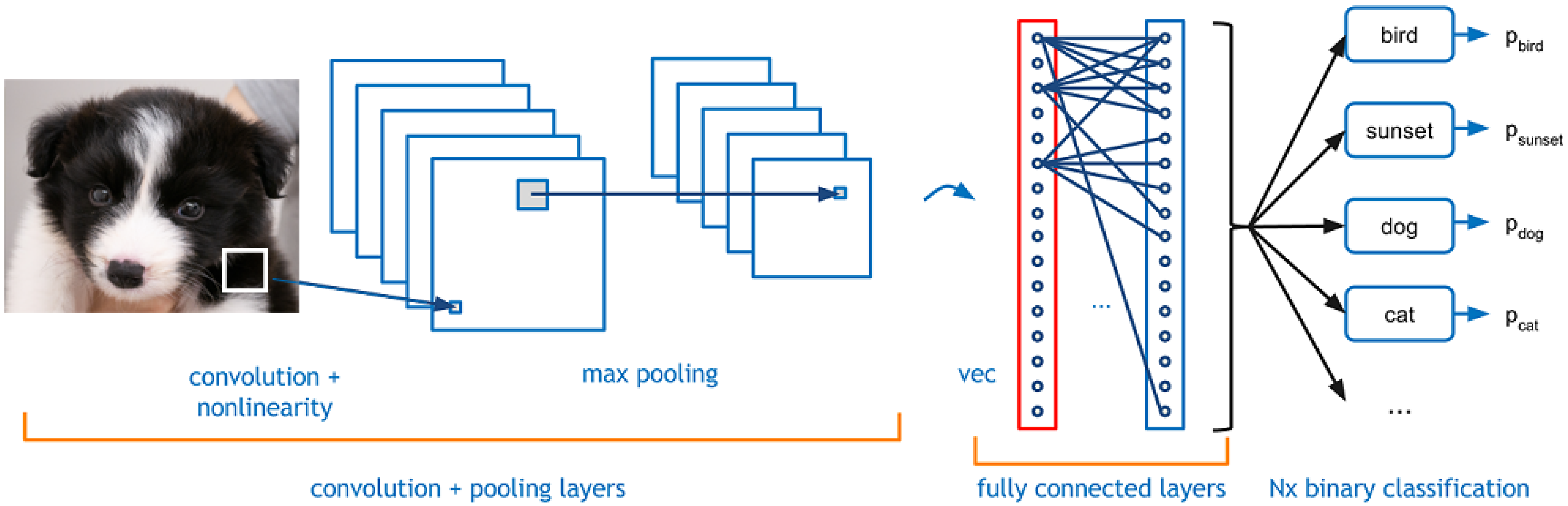
$$w^* = arg \min_w L(w)$$

- Consider loss function $L(w)$ with one parameter w:
  - ➢ (Randomly) Pick an initial value $w^0$
  - ➢ Compute $\frac{dL}{dw}|_{w=w^0}$

$$w^1 \leftarrow w^0 - \eta \frac{dL}{dw}|_{w=w^0}$$

# Neural Network

## Convolutional Neural Network (CNN)

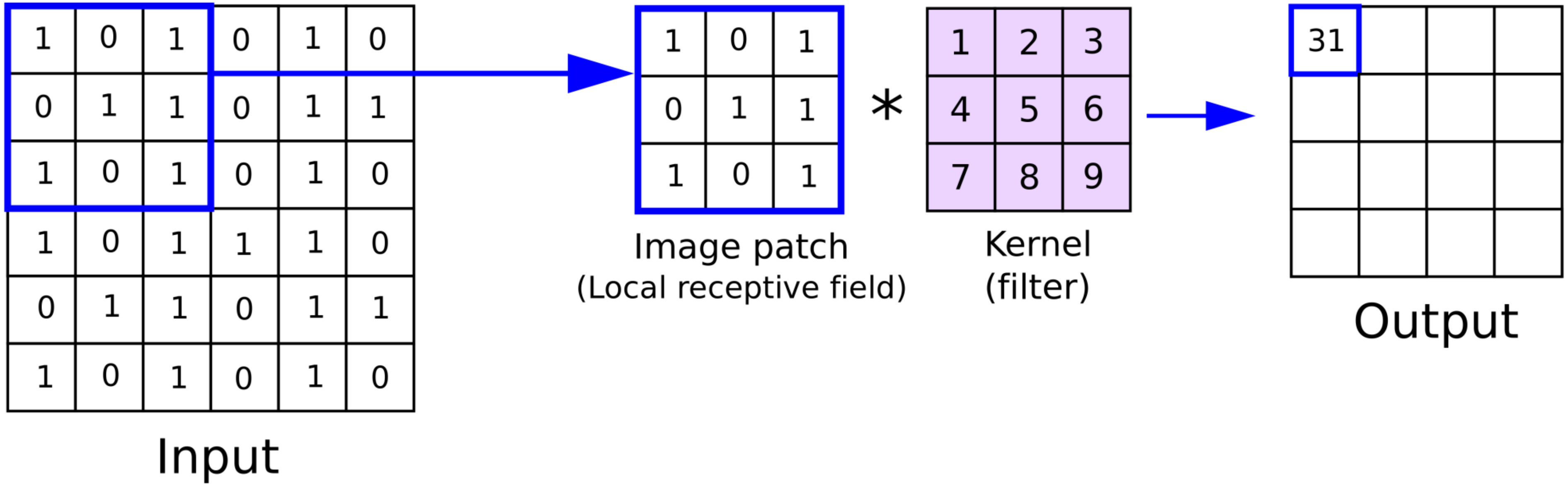


https://medium.com/%E9%9B%9E%E9%9B%9E%E8%88%87%E5%85%94%E5%85%94%E7%9A%84%E5%B7%A5%E7%A8%8B%E4%B8%96%E7%95%8C/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-ml-note-convolution-neural-network-%E5%8D%B7%E7%A9%8D%E7%A5%9E%E7%B6%93%E7%B6%B2%E8%B7%AF-bfa8566744e9

# Neural Network

## Convolutional Neural Network (CNN)

Convolutional Layer



|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |

Input

|  |  |  |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |

Image patch
(Local receptive field)

*

|  |  |  |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Kernel
(filter)

|  |  |  |  |
|---|---|---|---|
| 31 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Output

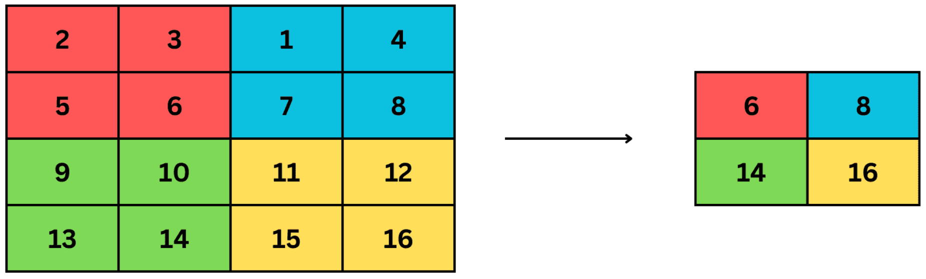
https://anhreynolds.com/blogs/cnn.html

# Neural Network

## Convolutional Neural Network (CNN)

### Pooling Layer

**Max Pooling**



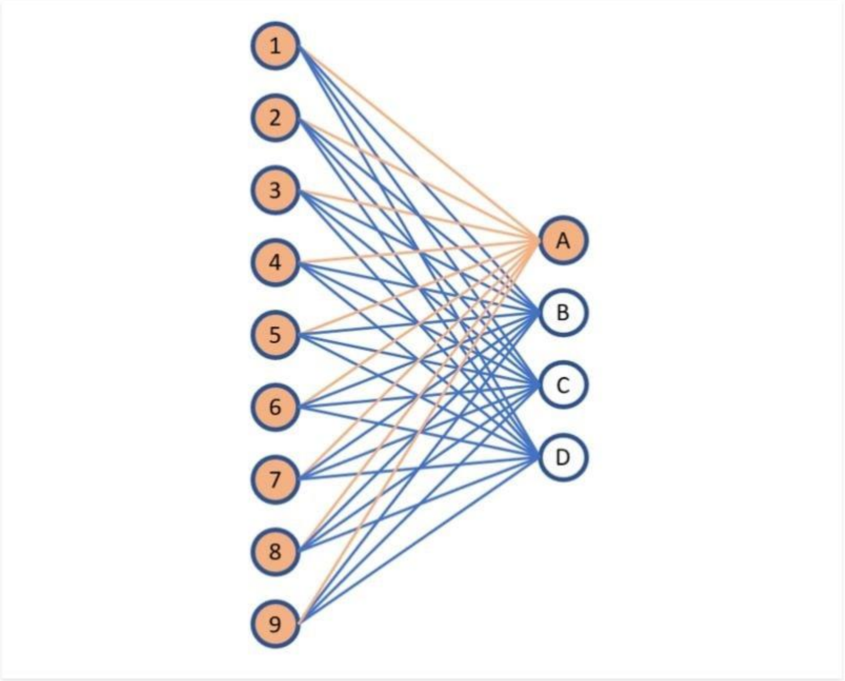
|  |  |  |  |
|---|---|---|---|
| 2 | 3 | 1 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

|  |  |
|---|---|
| 6 | 8 |
| 14 | 16 |

*Select the maximum value from each window.*

https://builtin.com/machine-learning/fully-connected-layer

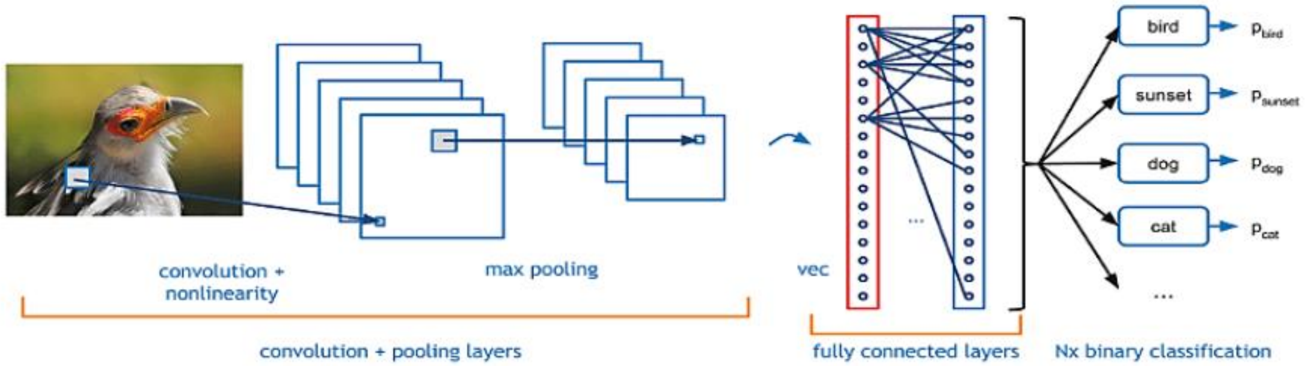
### Fully Connected Layer

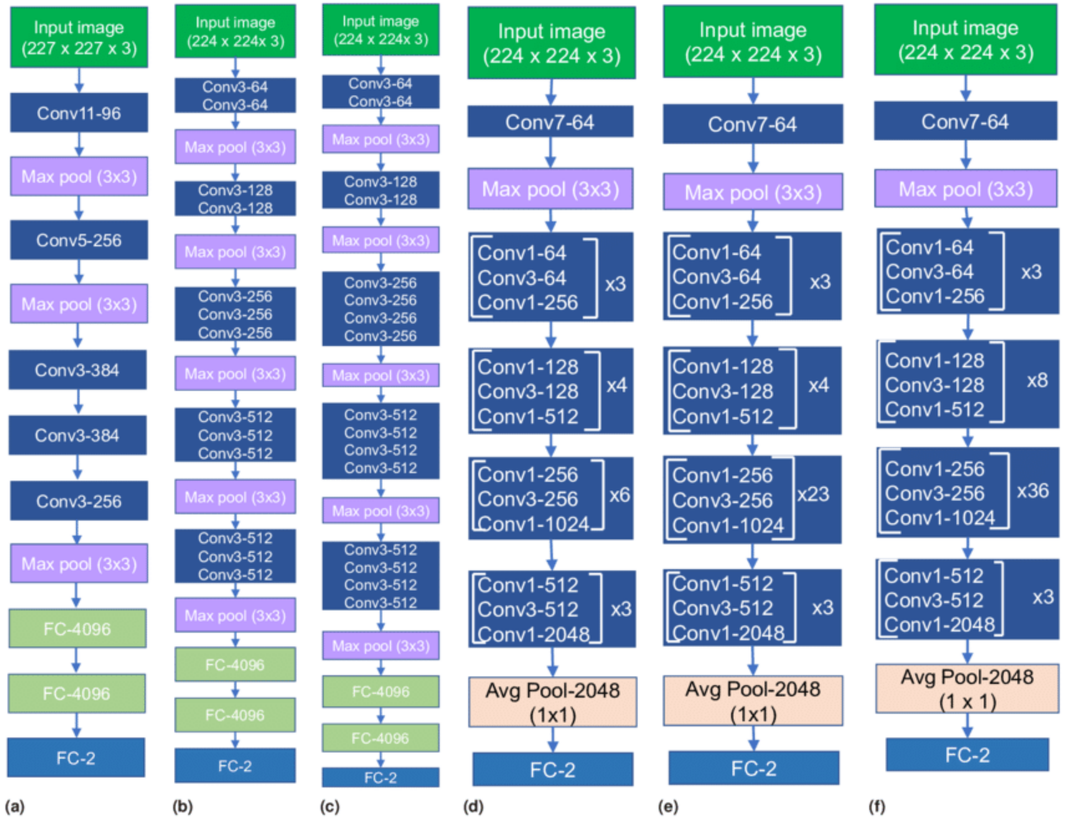


https://builtin.com/machine-learning/fully-connected-layer

# Object Recognition and Detection

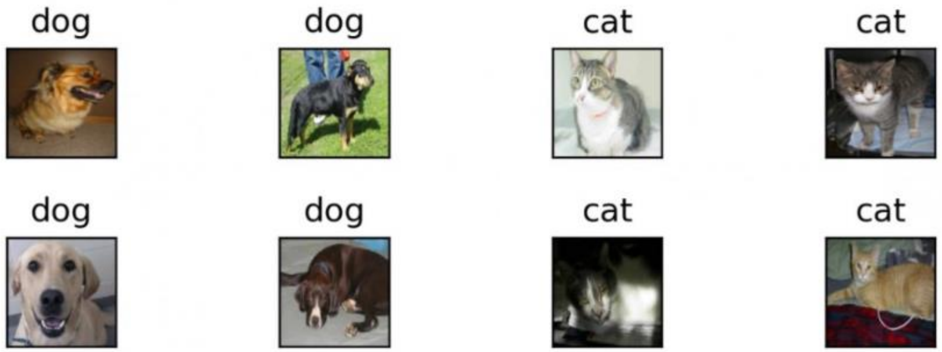
## images recognition

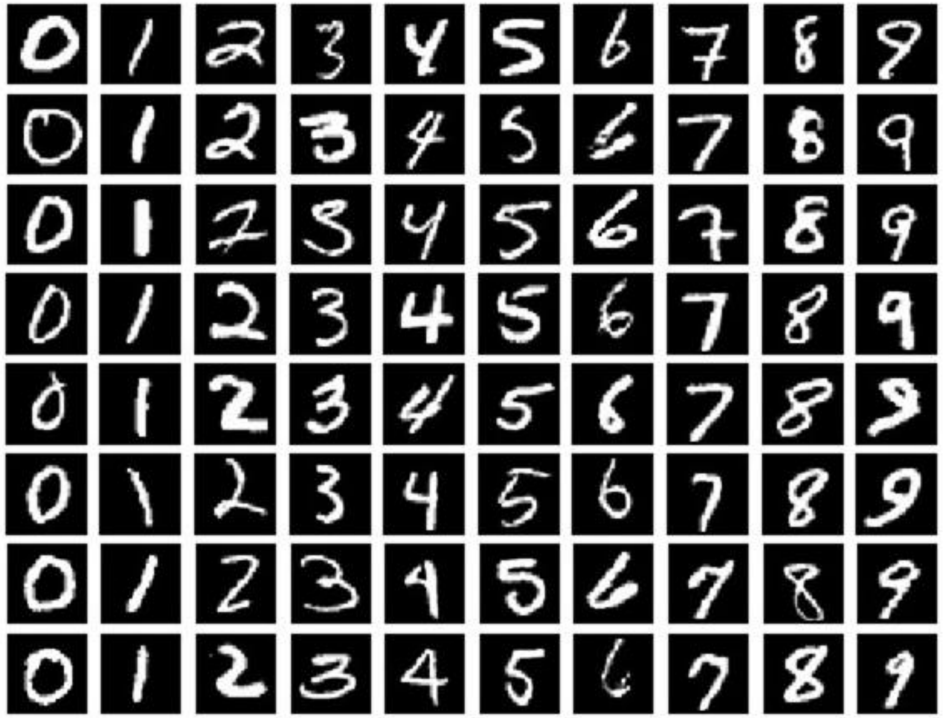




# Object Recognition and Detection

## images recognition





# Object Recognition and Detection

## images recognition

ballplayer 69.22%

anemone_fish 92.48%

African_elephant 89.94%

forklift 98.95%

ice_cream 99.60%

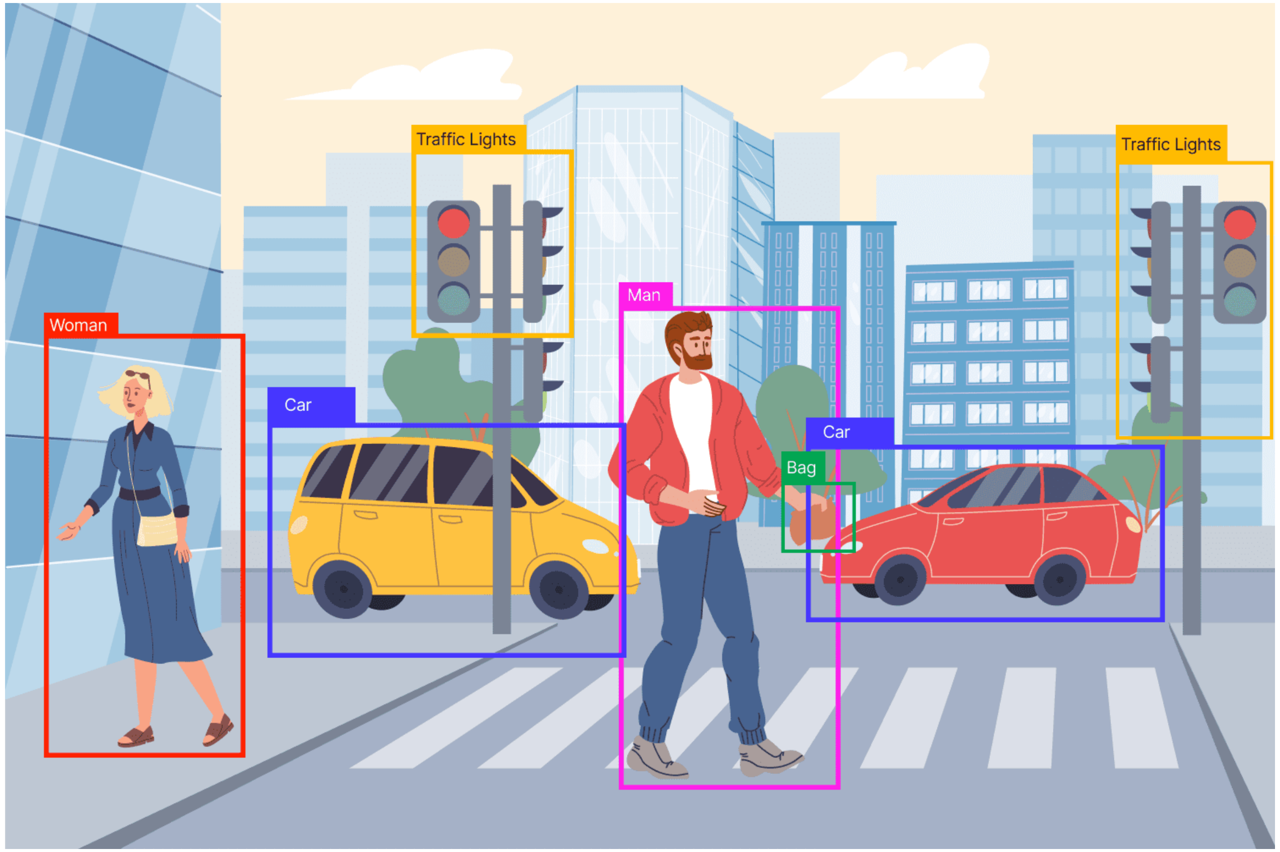
lemon 97.06%

magnetic_compass 97.08%

ice_bear 84.80%

# Object Recognition and Detection

## Object detection



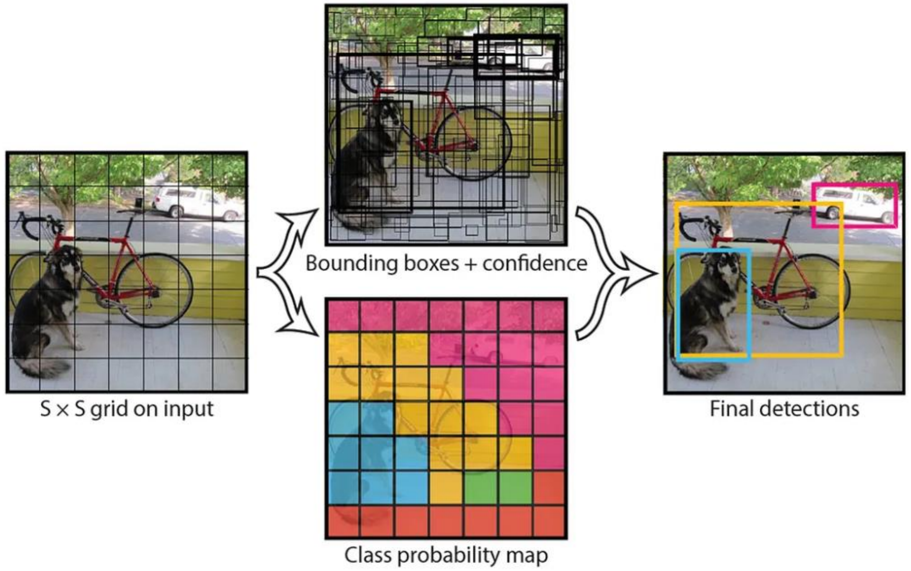
- Bounding boxes: Bounding boxes are a commonly used annotation type for object detection. They are used to mark the rectangular area of an object within an image.
- Points: Points can be used to mark a specific location within an image. This annotation type is often used for facial recognition or keypoint detection.

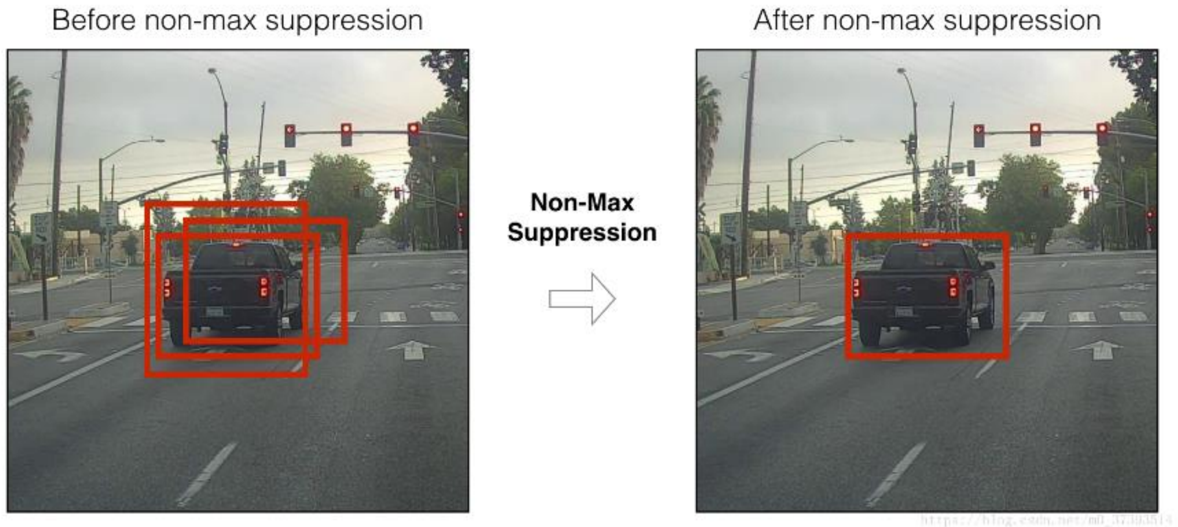
https://deeplobe.ai/exploring-object-detection-applications-and-benefits/

# Object Recognition and Detection

## Object detection

- One stage object detection : ssd 、 yolo

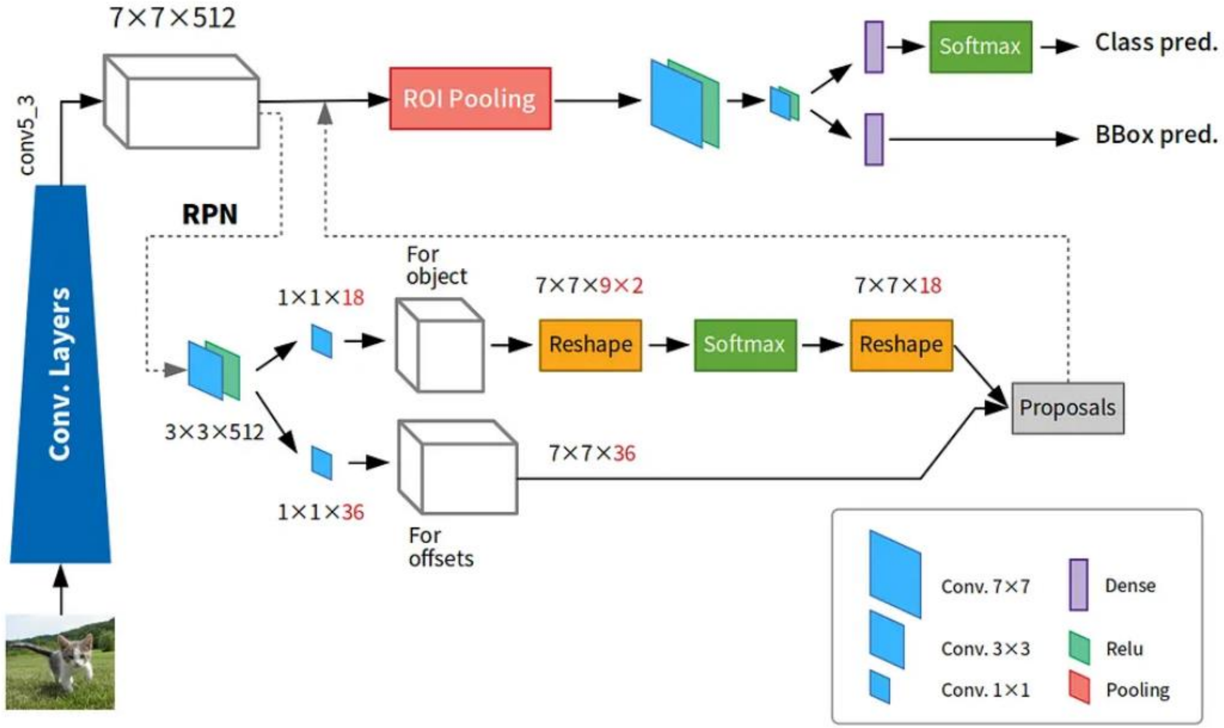


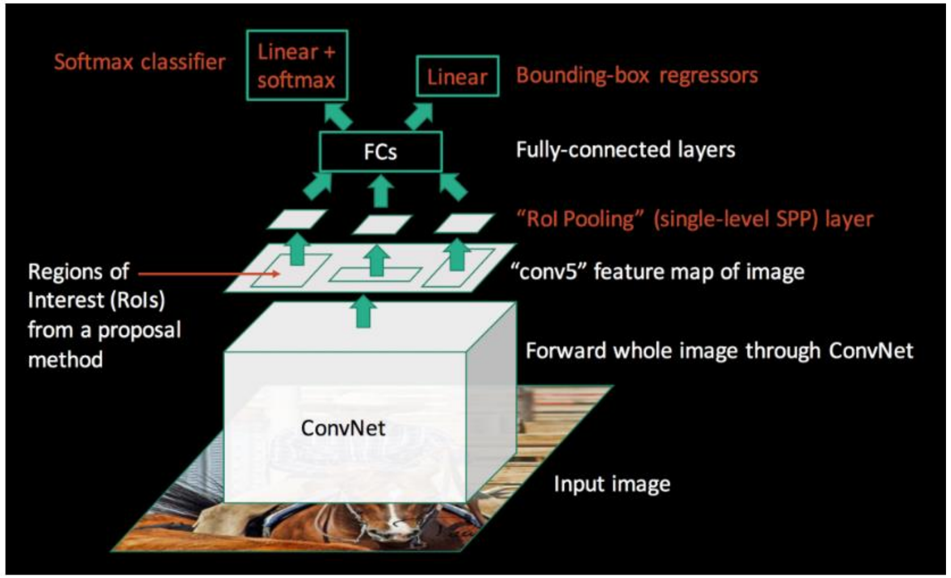


# Object Recognition and Detection

## Object detection

- Two stage object detection : RCNN 、faster-RCNN 、fpn 、mask-RCNN




Softmax classifier
Linear + softmax
Linear
Bounding-box regressors
FCs
Fully-connected layers
"RoI Pooling" (single-level SPP) layer
Regions of Interest (RoIs) from a proposal method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image

# THANK YOU :))))