

Image Processing with Python

Sampriti Chatterjee (Great Learning)

Agenda

- 1 What is Image Processing?
- 2 Traditional ML VS Deep learning
- 3 Human brain VS AI
- 4 How Image Input work in computer?
- 5 What is CNN?
- 6 Layers in CNN
- 7 Convolutional Layer
- 8 Relu layer
- 9 Pooling layer
- 10 Fully connected layer
- 11 Demo: Image Processing with Python

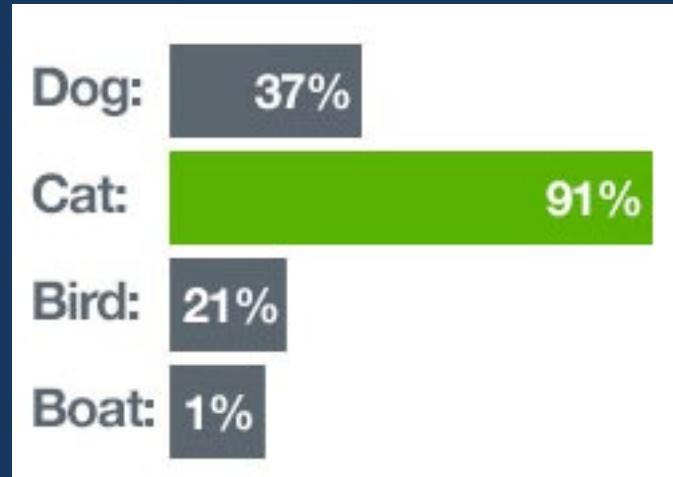
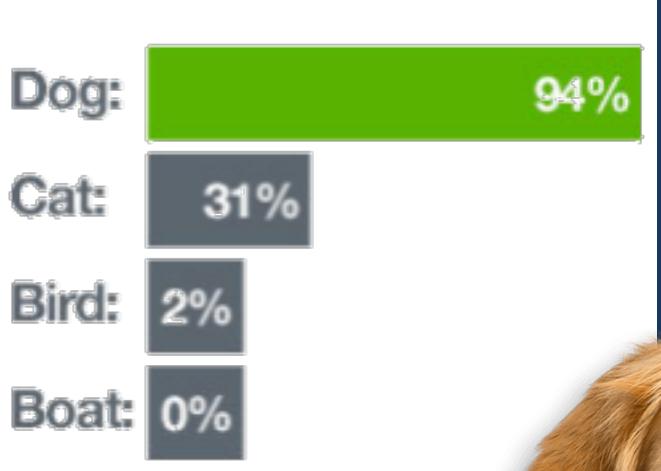
What is Image Processing

Image processing is a method to perform some important operations on an image. In order to get an enhanced high Quality image or to extract the most useful information from that

- It is a one type of signal processing
- In this processing input is an image and output may be image or characteristics/features associated with that image

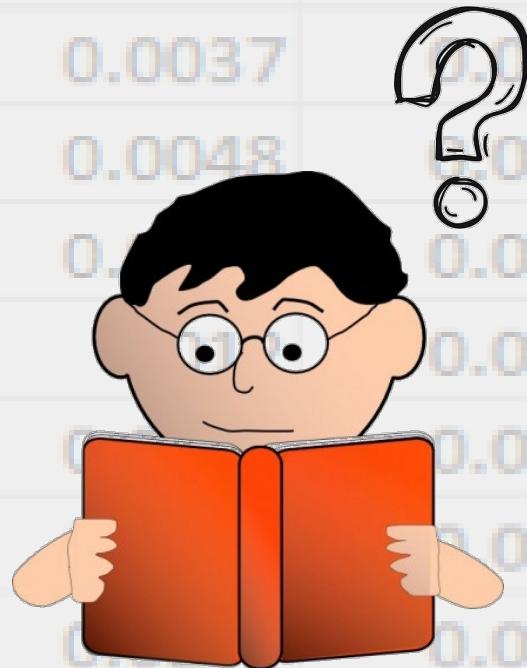


Is It A Cat Or A Dog?



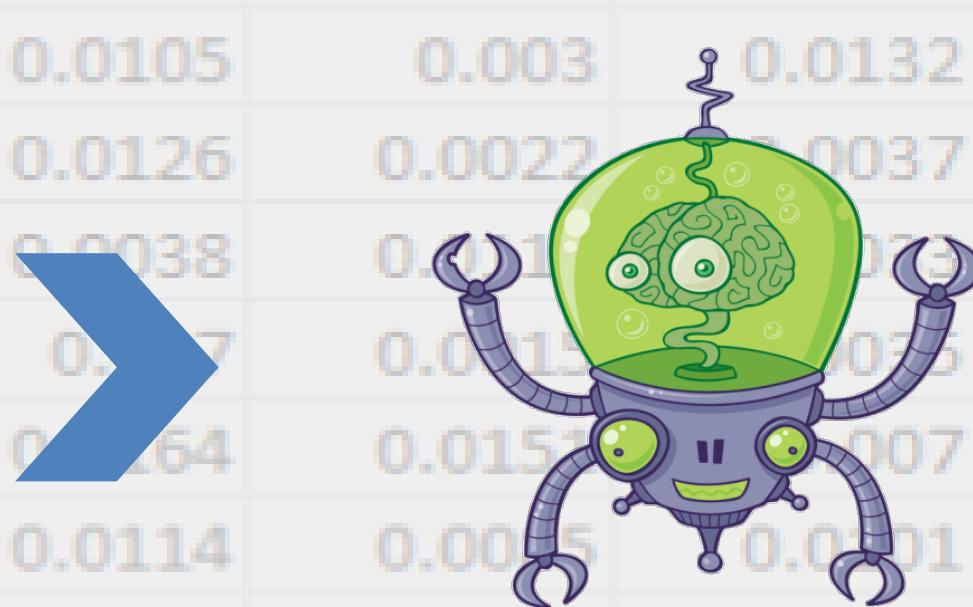
Is It A Cat Or A Dog?

Traditional Machine Learning



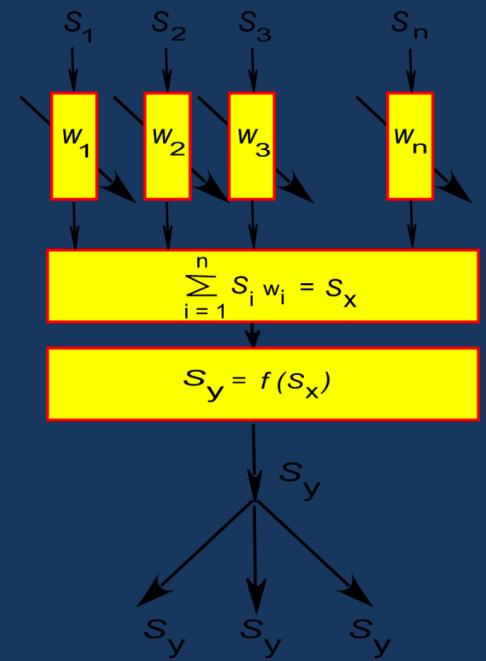
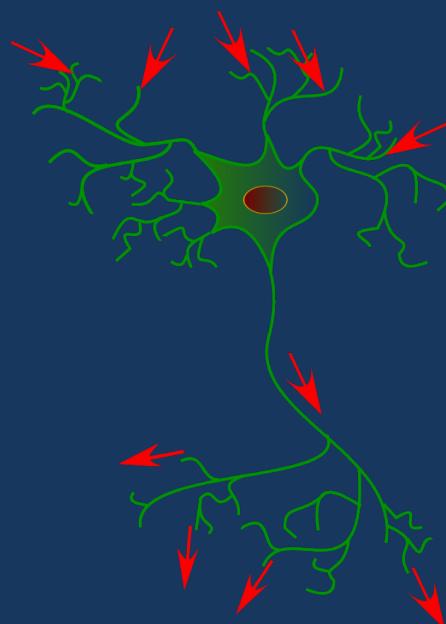
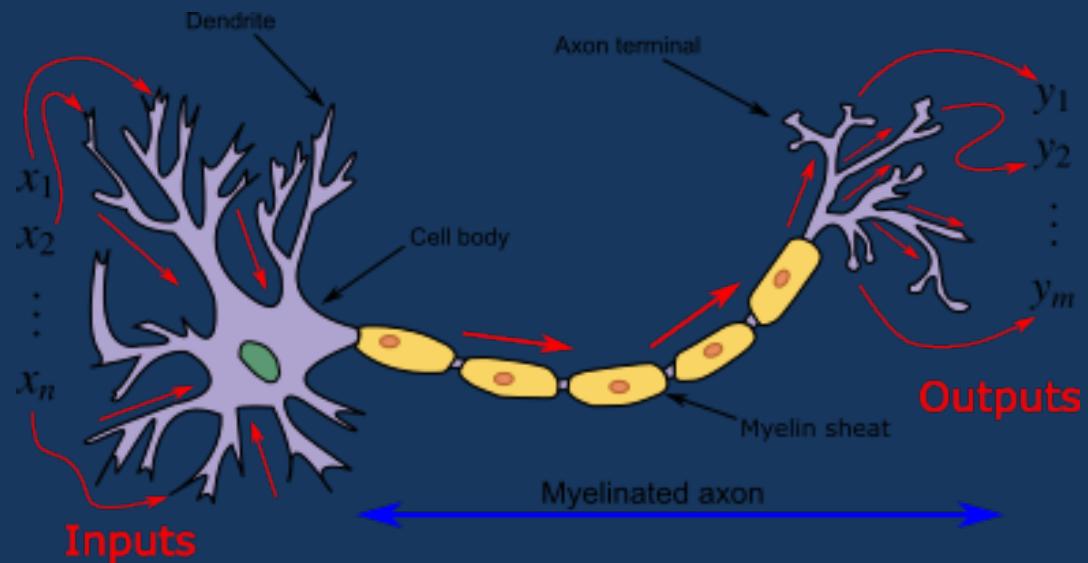
Manual selection from so many features is impossible

Deep Learning



This is where traditional machine learning fails and **deep learning** comes into picture

Human Brain VS Neural Network



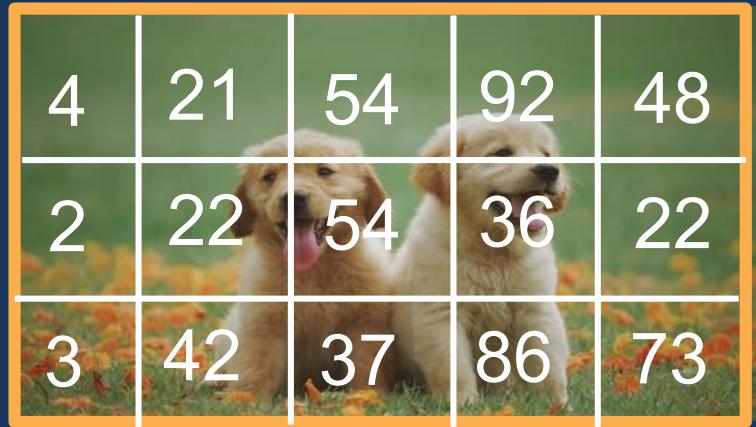
Human Brain VS Neural Network

Neuron	Artificial Neuron
• Cell Nucleus	• Node
• Dendrites	• Input: x_1, x_2
• Synapse	• Weights or interconnections
• Axon	• $f(x)$ (mapping, activation, learning)
• Terminal Axon	• Output: y_1

How Image Input Works In Computer?

Converts the image into an array of pixel values where the dimension of array depends on the resolution of the image

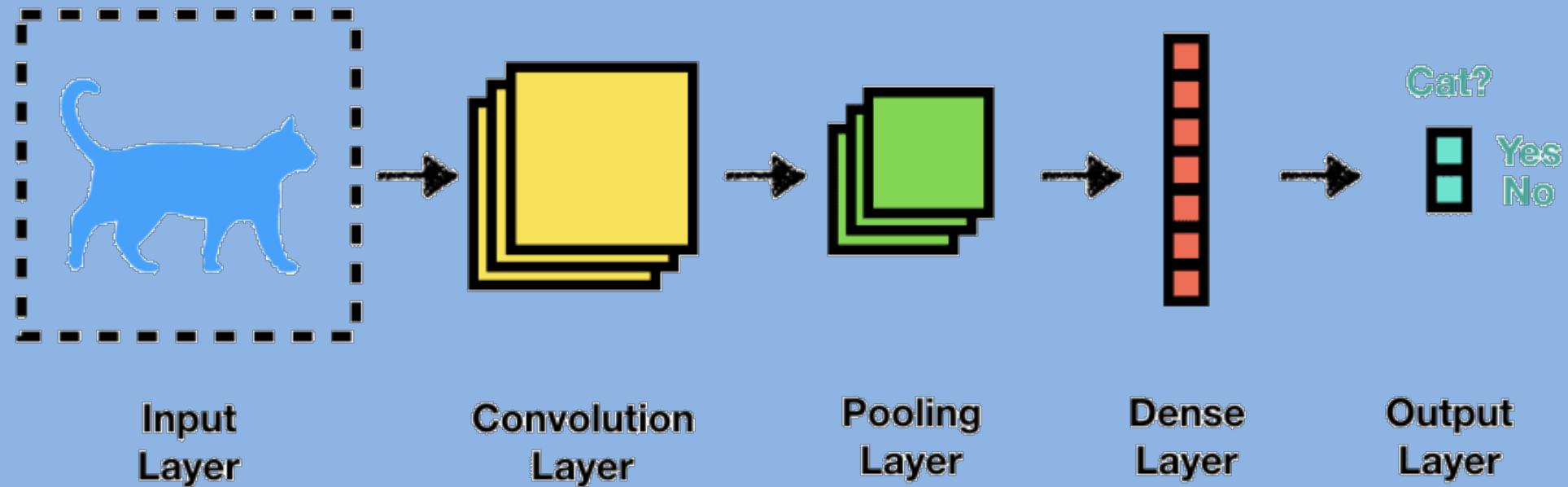
4	21	54	92	48
2	22	54	36	22
3	42	37	86	73



Array of dimension 32 X 32 X 3 (The 3 refers to RGB values)

What is CNN?

“A feed forward network to process and recognize image data with the grid version”



Fully Connected Network

Combines the extracted features
and represents a new model

ReLU Layer

Converts negative numbers into zero

Pooling Layer

Reduce the image size

Convolution Layer

Converts images into an array

Layers In CNN

Layers in CNN

Convolutional Layer

“Converts images into an array”

1

Convolutional Layer

2

ReLU

3

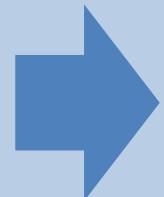
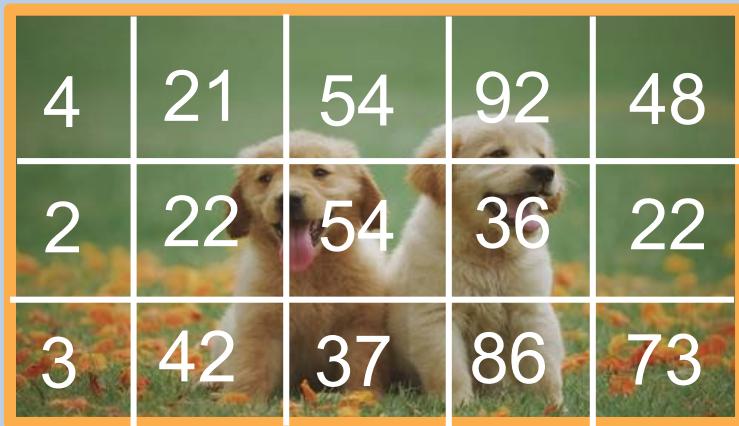
Pooling Layer

4

Fully Connected Layer

- First layer of CNN
- Stores the pixelated values of image into an array
- Used for extracting the features of the image and reducing its dimensionality

Working Of Convolutional Layer



4	21	54	92	48
2	22	54	36	22
3	42	37	86	73

*

0	-1
1	1

Filter/Feature detector

Terminology Alert!!

3	22	-2	10
23	25	87	139



- Extracted features from the image
- Dimension reduced

Layers In CNN

Layers in CNN

1

Convolutional Layer

2

ReLU

3

Pooling Layer

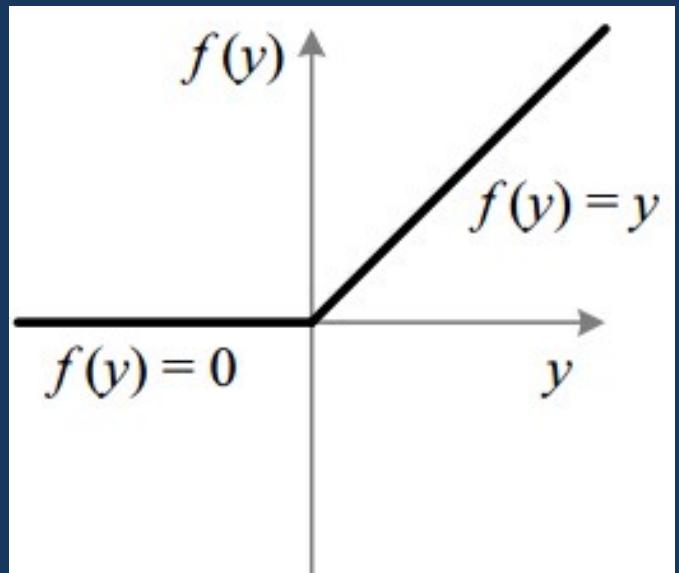
4

Fully Connected Layer

Activation function: ReLu

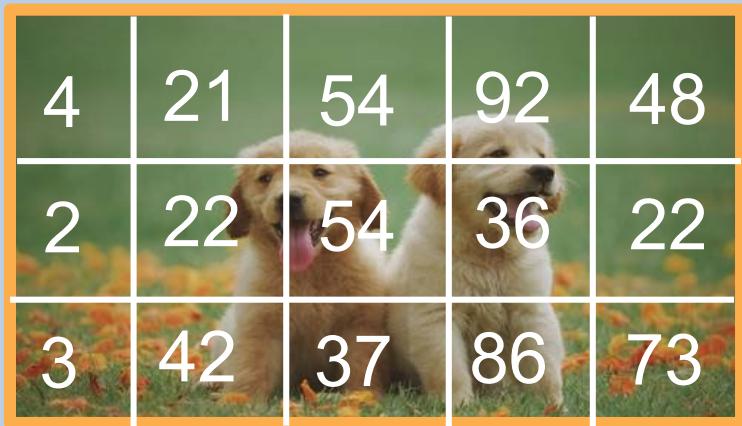
“Converts negative values into zero”

- ReLU is a half rectifier
 - $f(y) = 0$ when $y < 0$
 - $f(y) = y$ when $y \geq 0$
- Range of ReLU : [0 to infinity]



ReLu

Working Of Relu Layer



3	22	-2	10
23	25	87	139

Applying ReLu layer

3	22	0	10
23	25	87	139



After removing the negative values

Layers In CNN

Pooling Layer

Layers in CNN

“Reduces the spatial size and the numbers of parameters”

1

Convolutional Layer

- Used to reduce dimensionality
- Helps to control overfitting
- Filters of size 2×2 are commonly used in it

2

ReLU

3

Pooling Layer

4

Fully Connected Layer

Working Of Pooling Layer

3	22	0	10
23	25	87	139



Max Pooling

3	22	0	10
23	25	87	139

25			
----	--	--	--

3	22	0	10
23	25	87	139

3	22	0	10
23	25	87	139

3	22	0	10
23	25	87	139

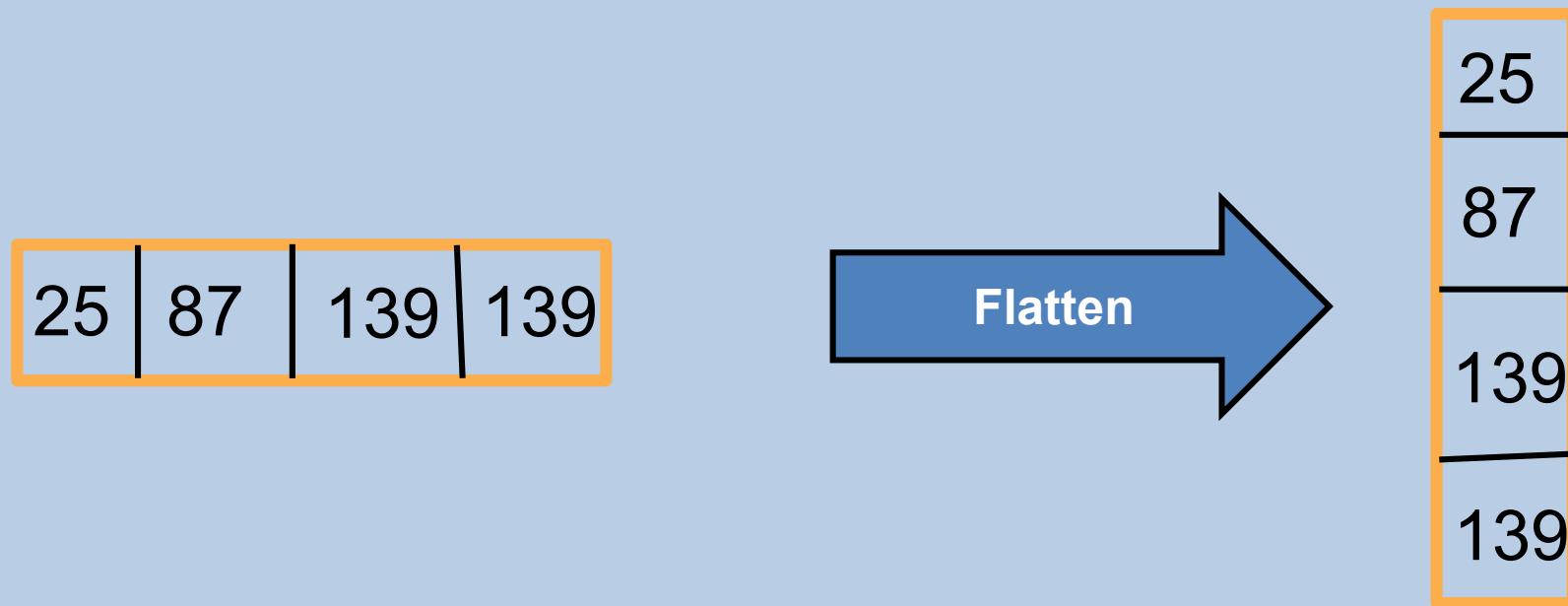
25	87		
----	----	--	--

25	87	139	
----	----	-----	--

25	87	139	139
----	----	-----	-----

Flatten The Data

“Converting the Pooled feature map into an array is known as data flattening”



Layers In CNN

Pooling Layer

Layers in CNN

“Combines all the features together to create a final model”

1

Convolutional Layer

- Used to reduce dimensionality
- Helps to control overfitting
- Filters of size 2×2 are commonly used in it

2

ReLU

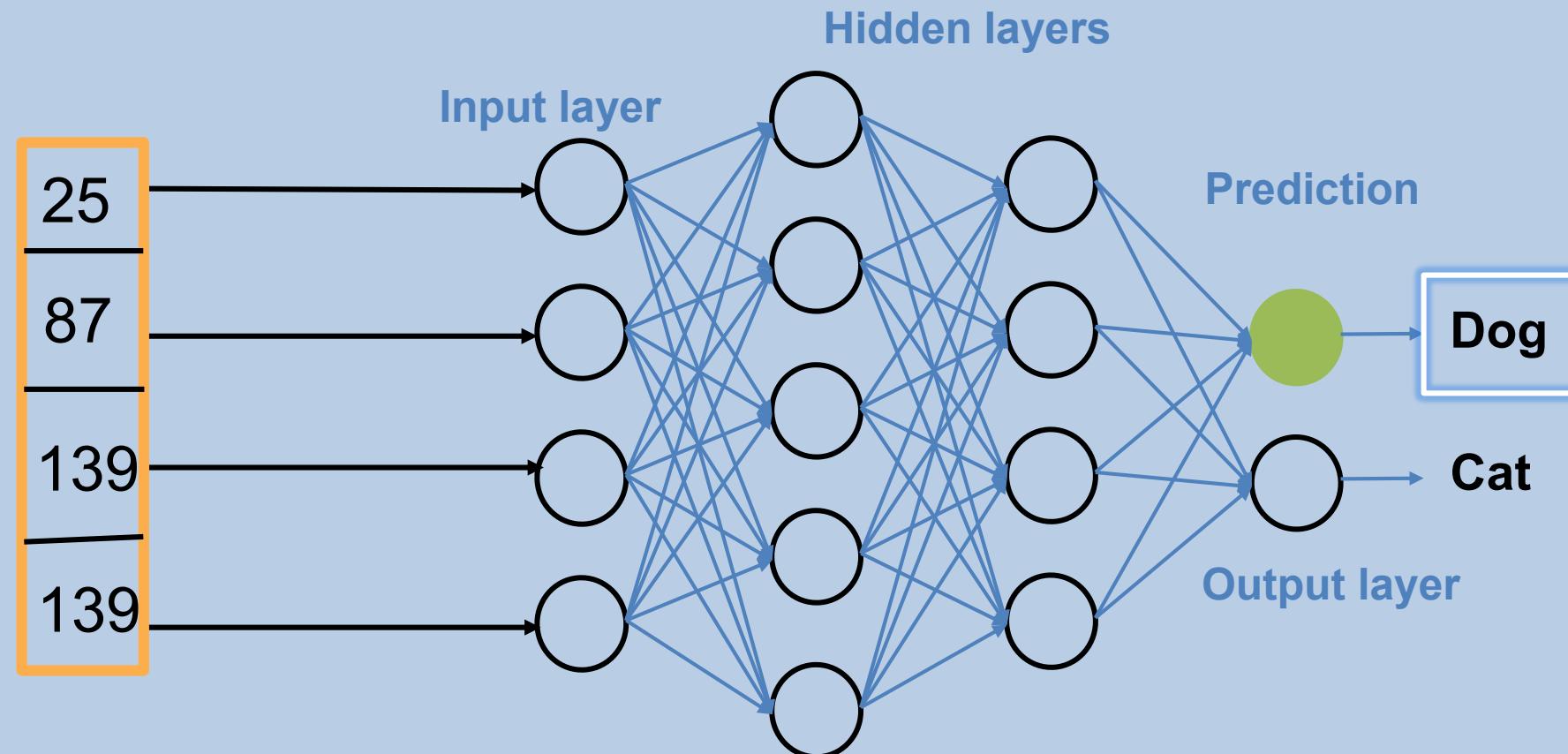
3

Pooling Layer

4

Fully Connected Layer

Working Of Fully Connected Layer



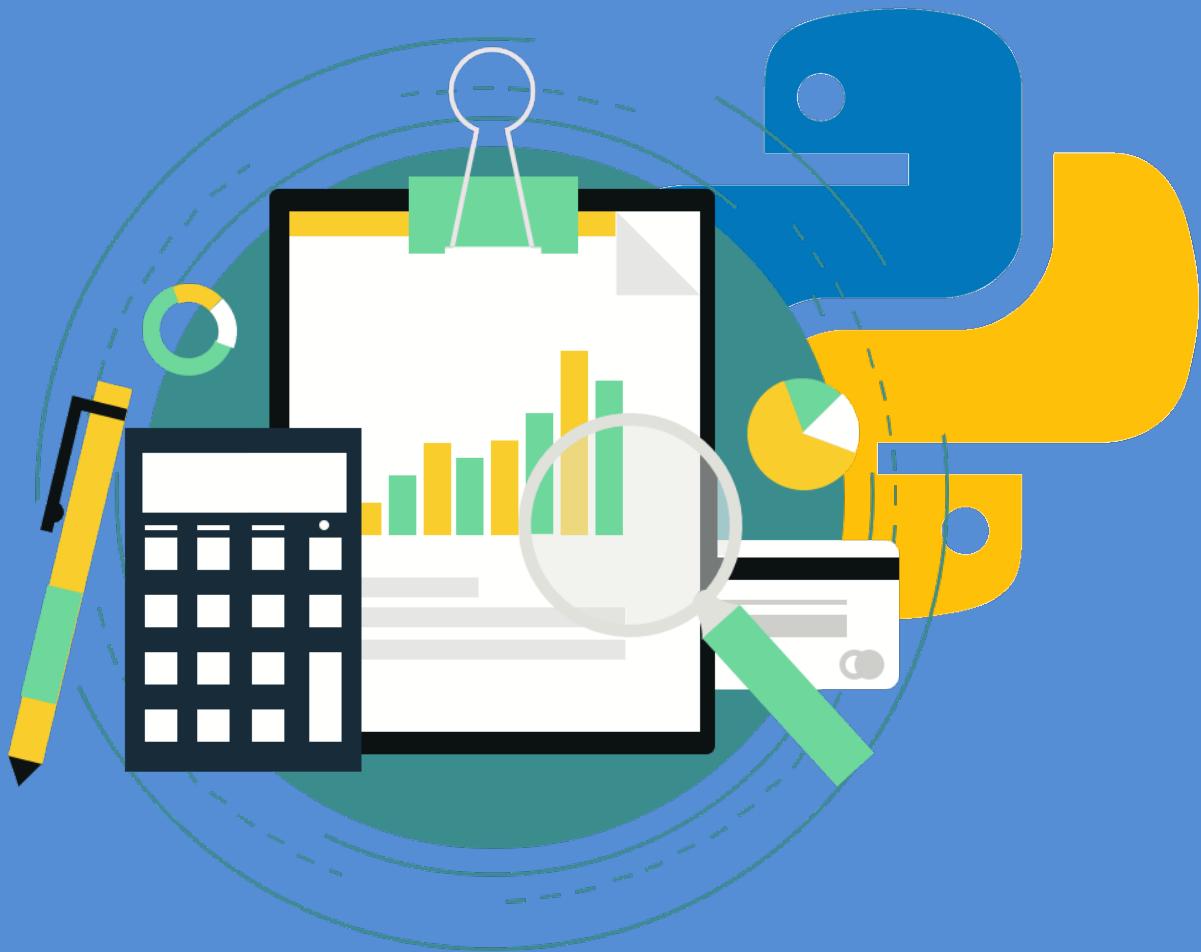


Image Processing with Python

Thank You