

2025



Introduction to Image Processing & Deep Learning

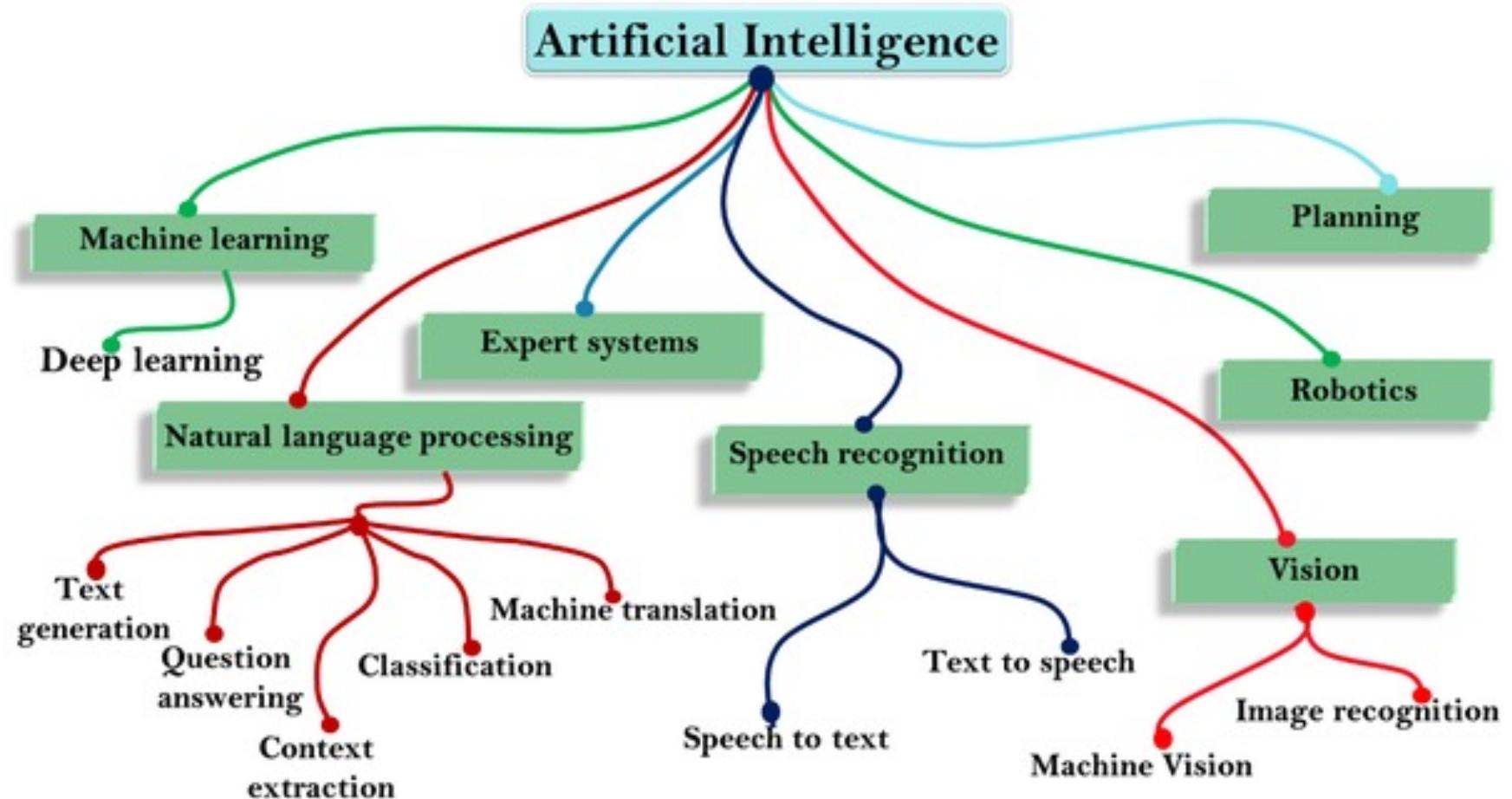
Ai Innovator 2025

รศ.ดร.กอบเกียรติ สารอุบล
Assoc.Prof.Kobkiat Saraubon, Ph.D.

kobkiat.s@sci.kmutnb.ac.th

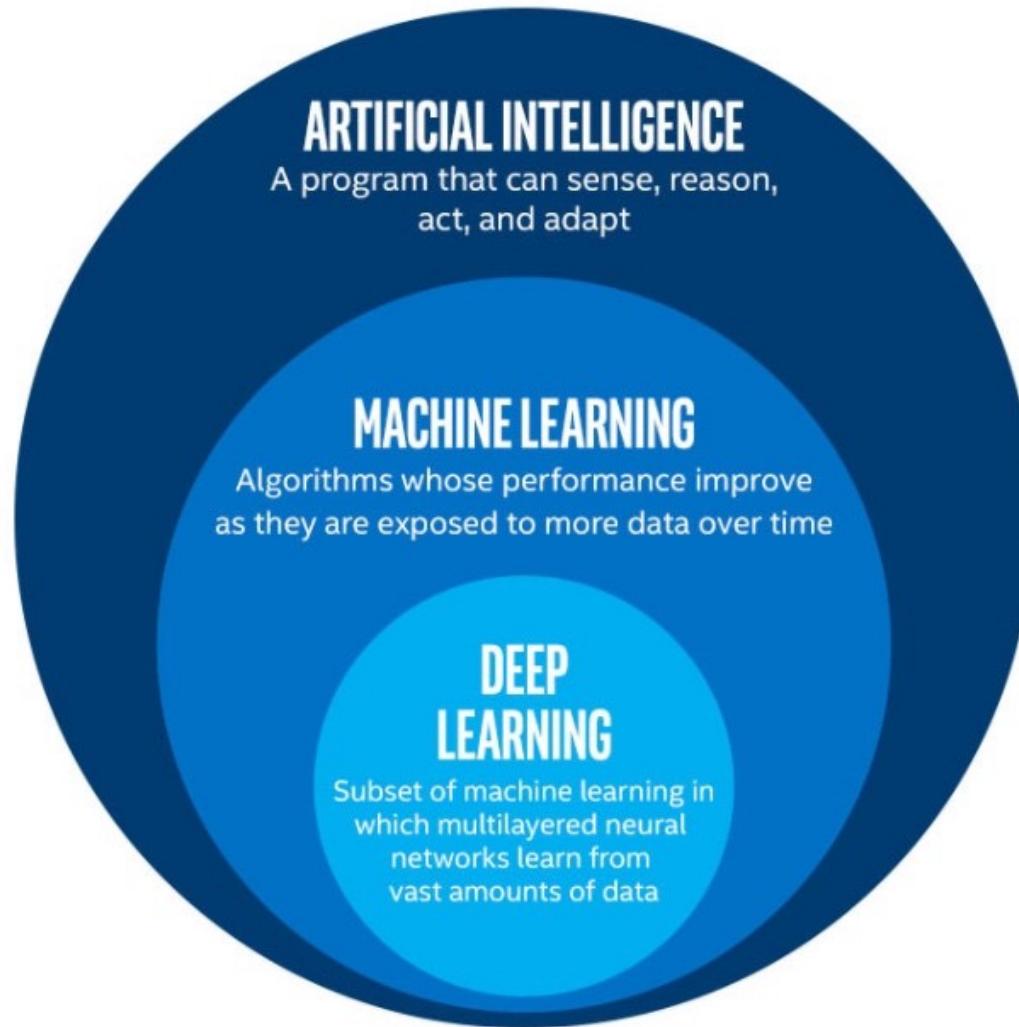


AI: ML, DL

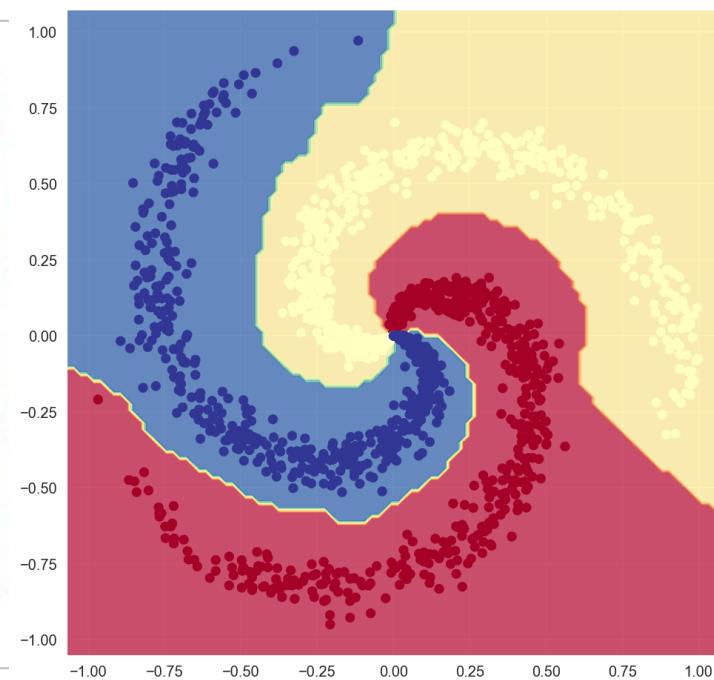
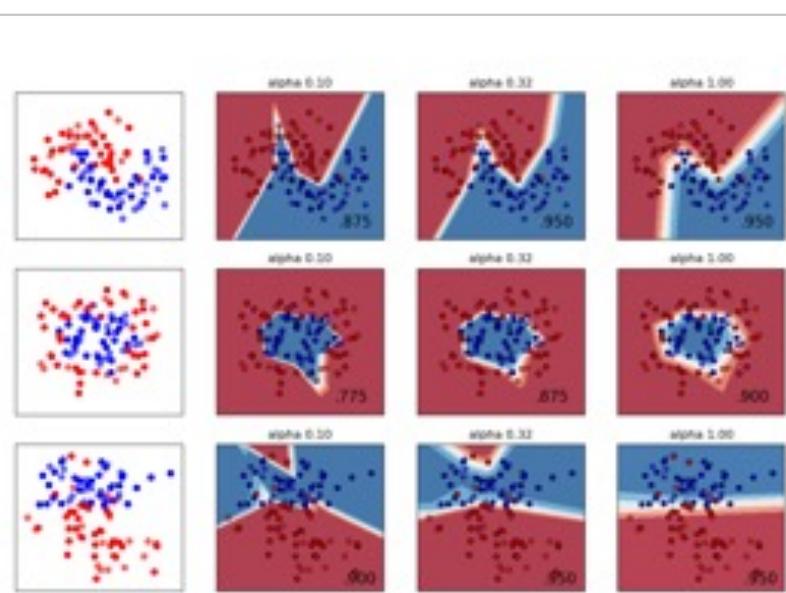
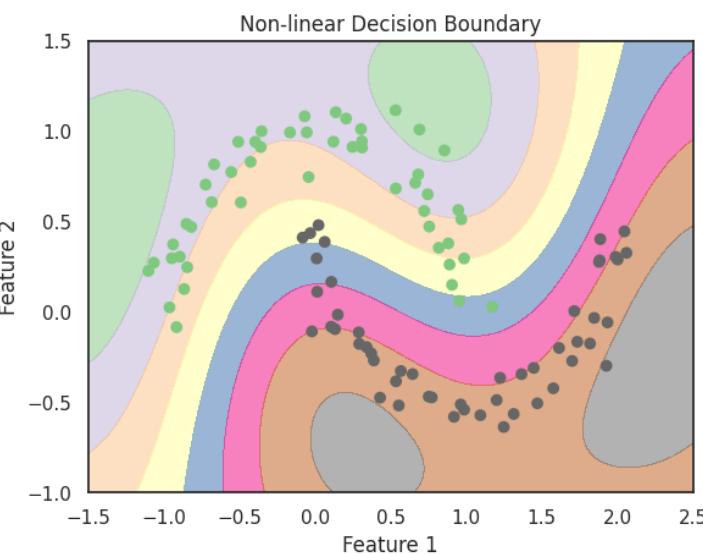
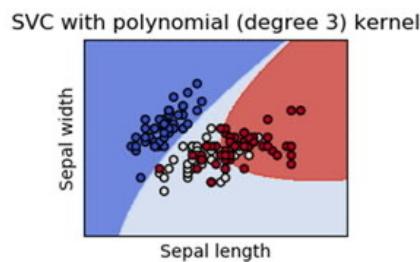
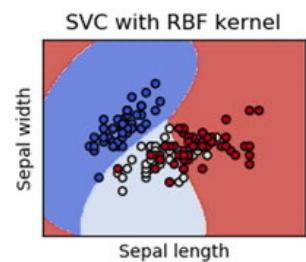
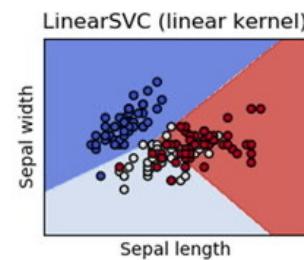
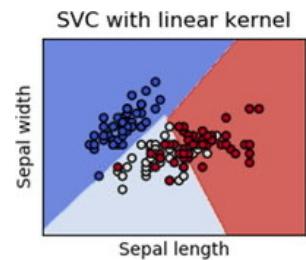


towardsdatascience.com

AI, ML, DL



towardsdatascience.com





MACHINE LEARNING



DE

towardsdatascience.com

DL Applications

Examples of Deep Learning Applications

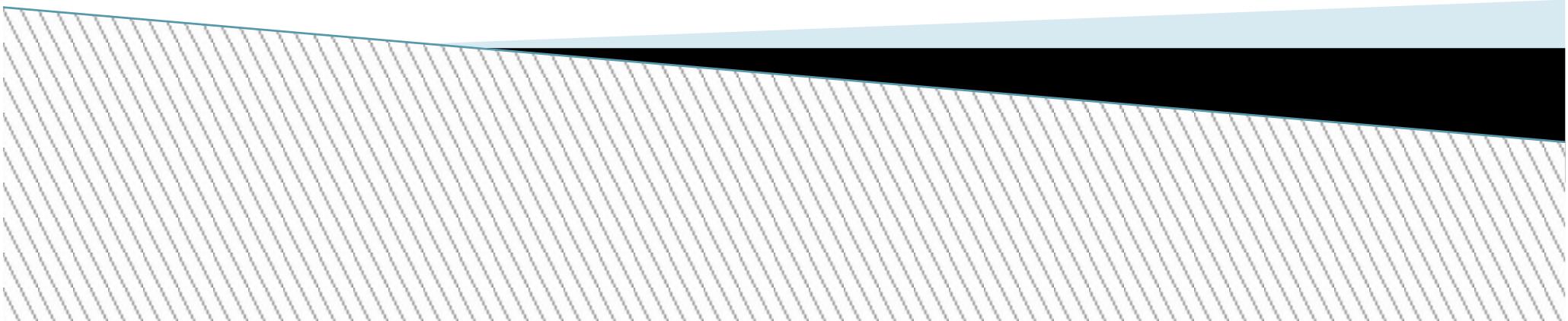




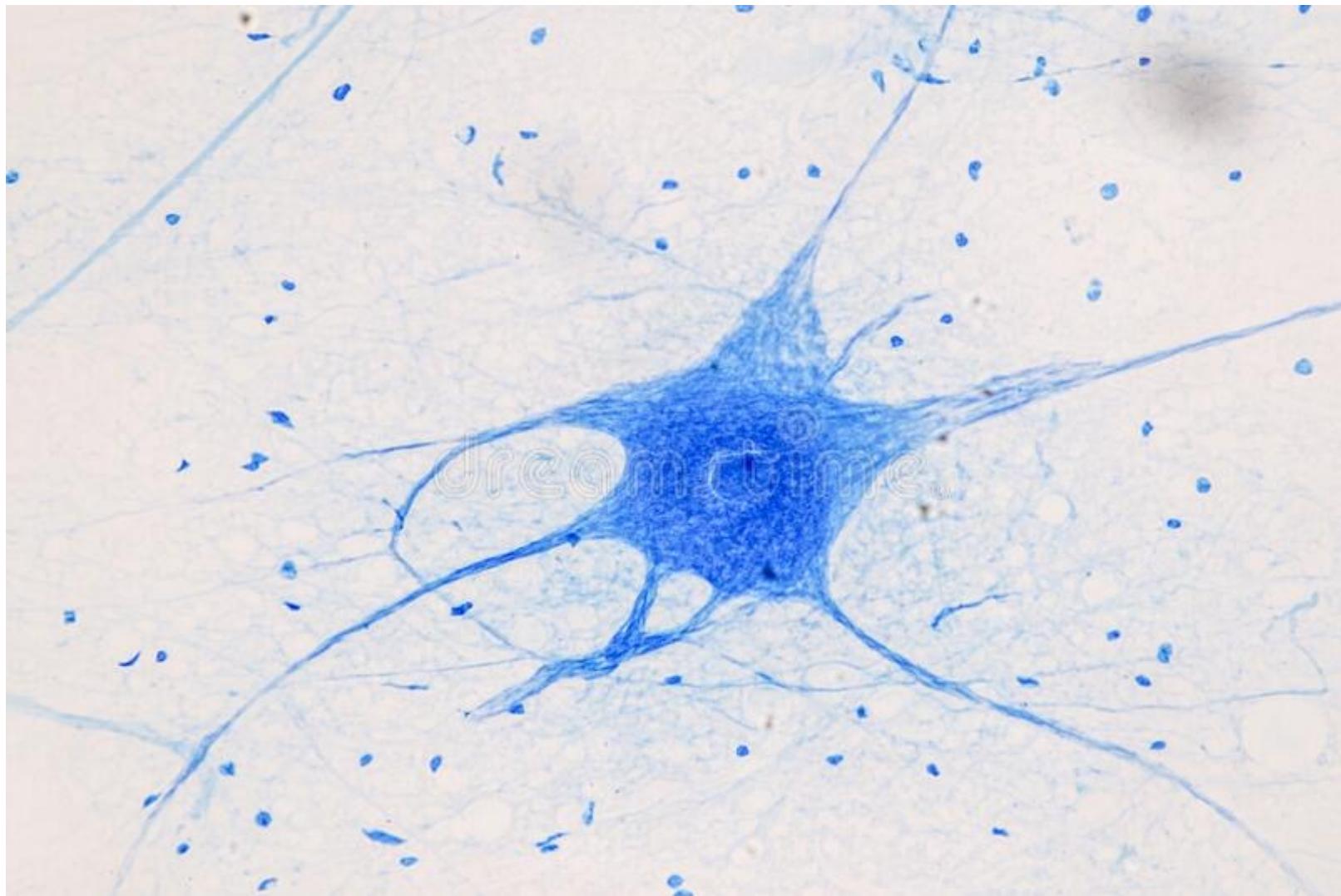
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระบรมราชูปถัมภ์

Artificial Neural Network



Neuron/Nerve cell

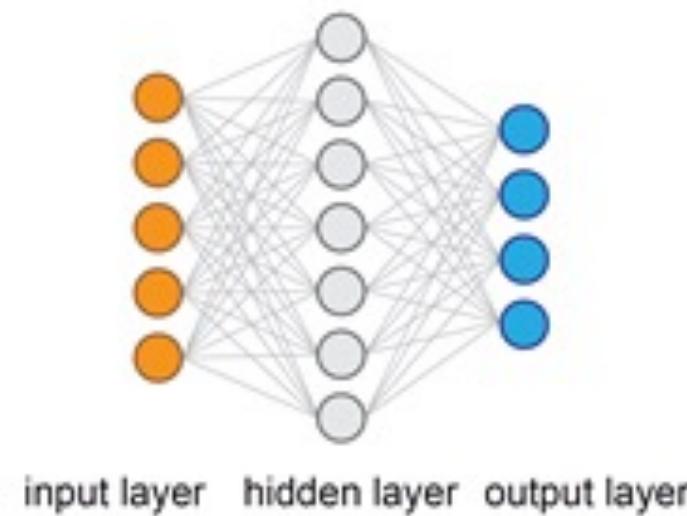


Neuron/Nerve cell

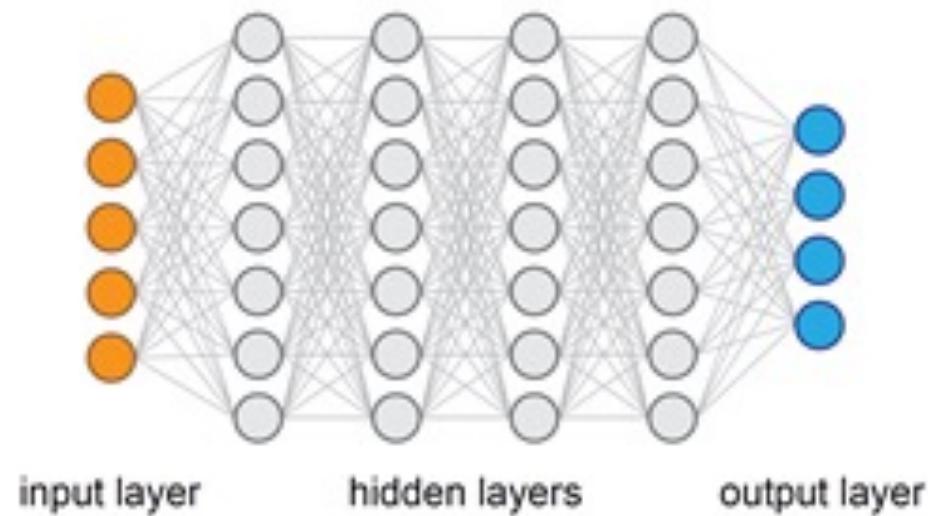


Shallow & Deep Neural Network

shallow neural network

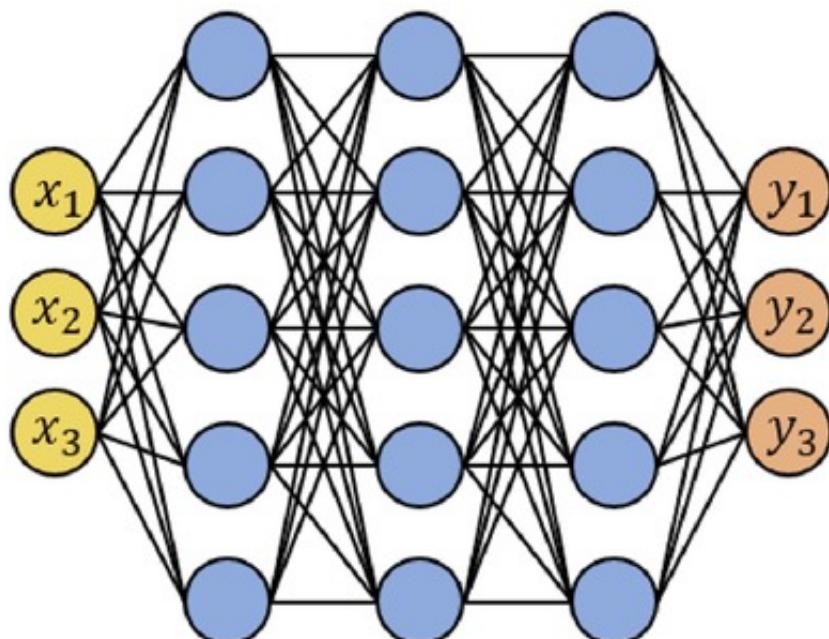


deep neural network

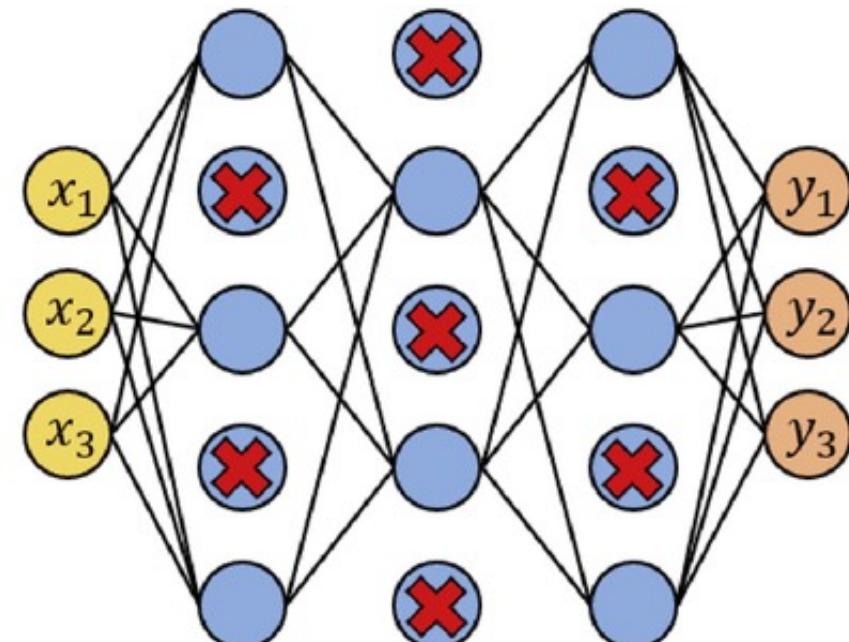


Dropout

- Dropout regularization is a technique used to prevent overfitting in neural networks by randomly deactivating a certain percentage of neurons during training.



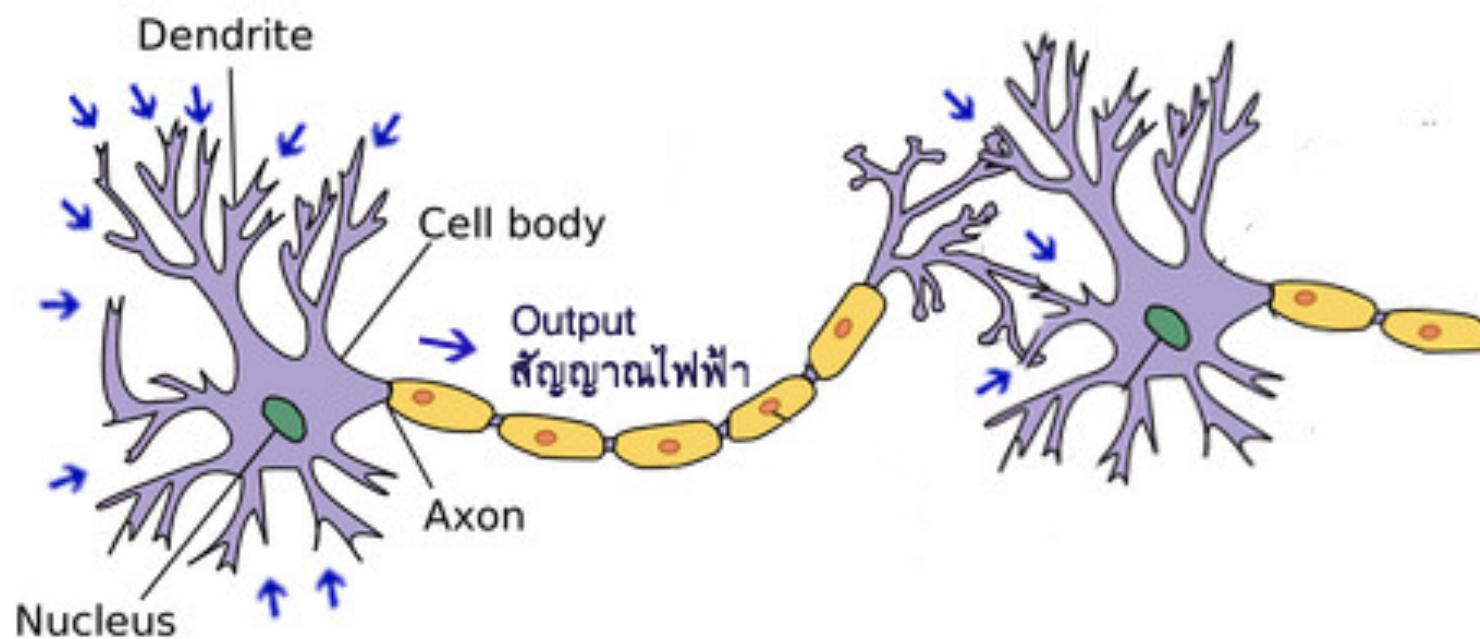
(a) Standard ANN



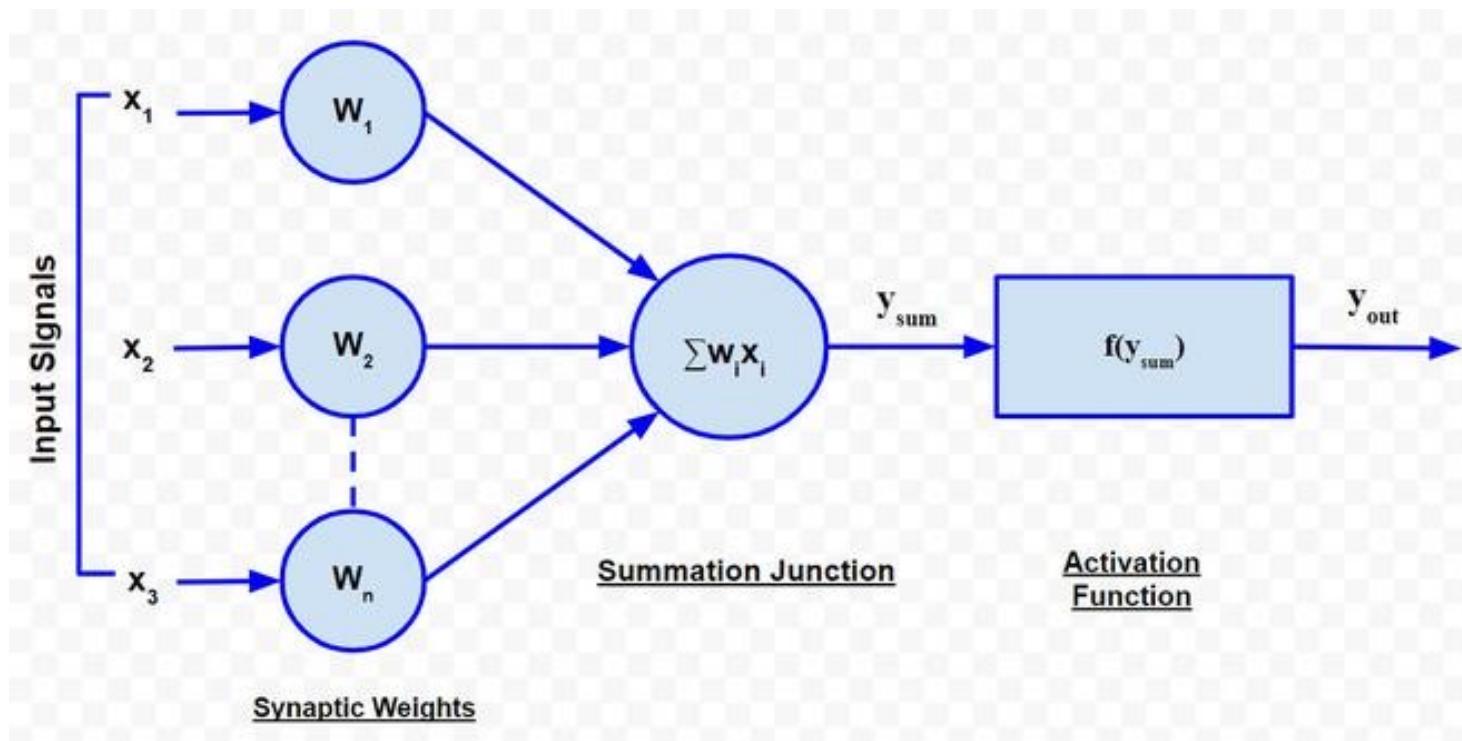
(b) Standard ANN with the dropout

Neural Network

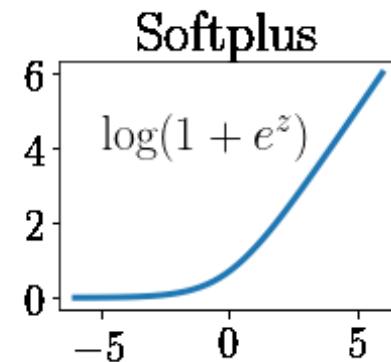
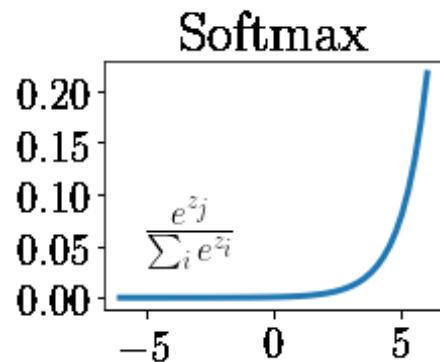
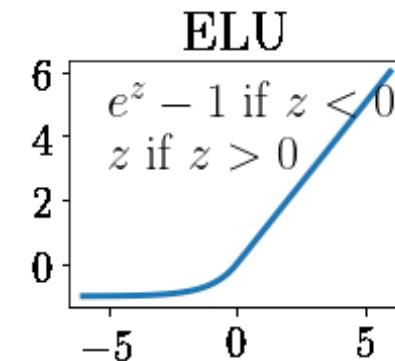
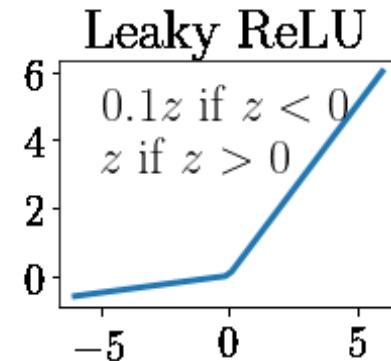
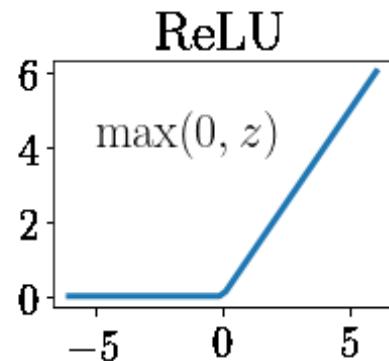
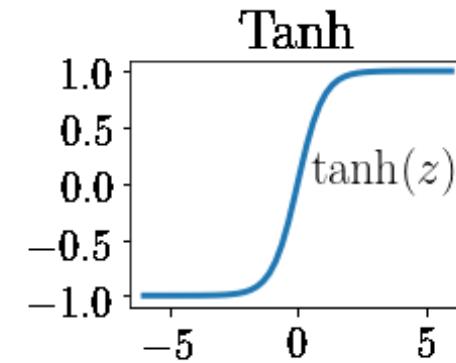
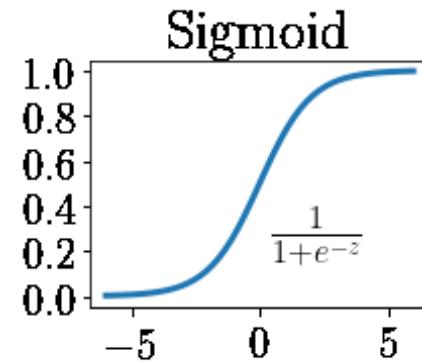
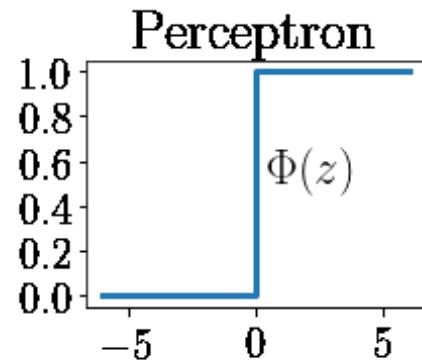
- Sum
- Activation



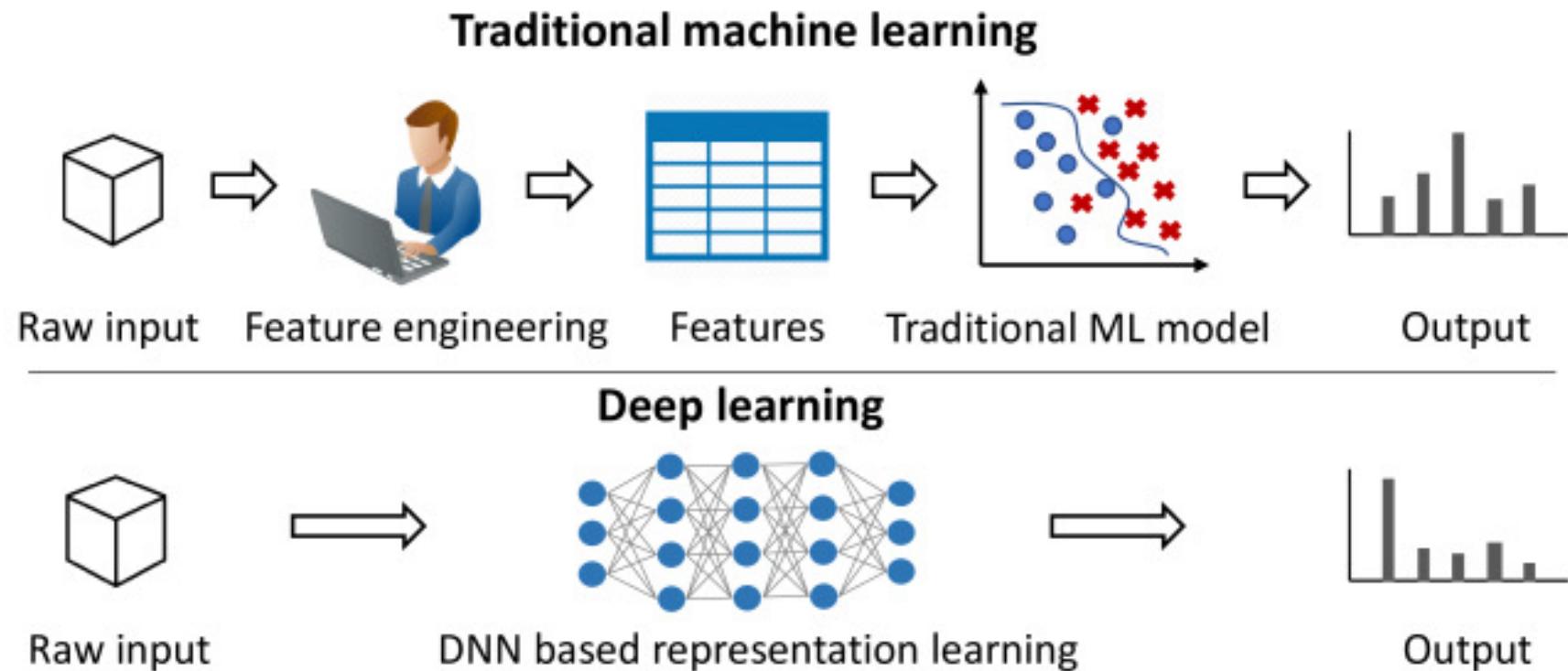
ANN



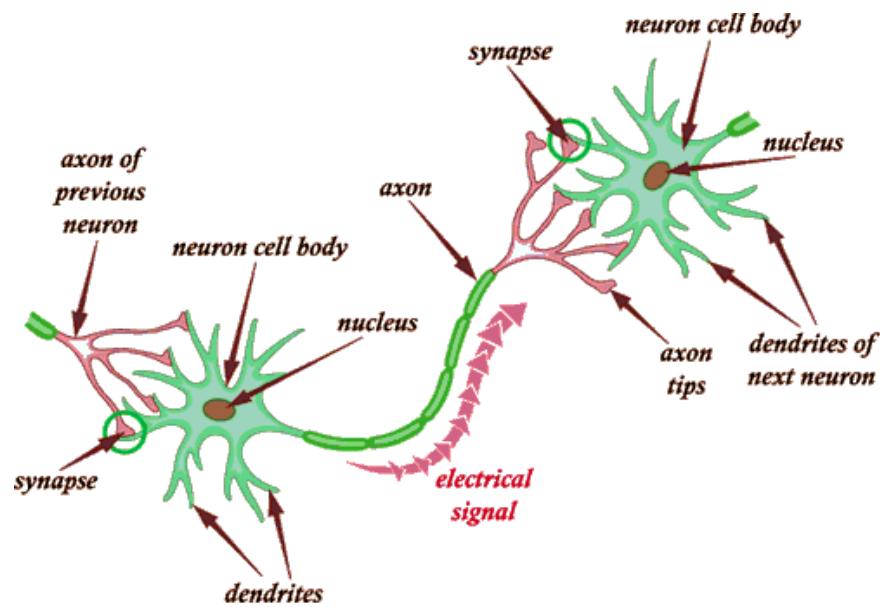
Activation Functions



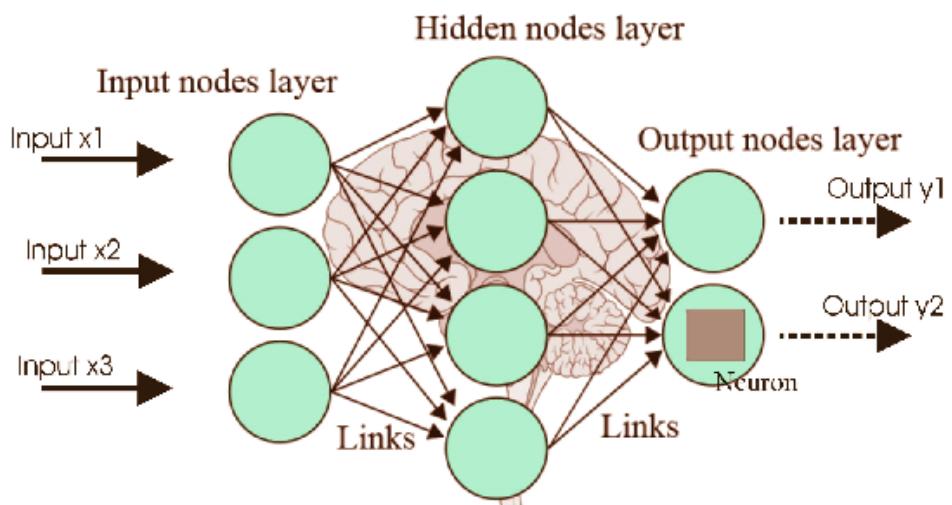
Machine Learning and Deep Learning



Neural Network and Deep Learning

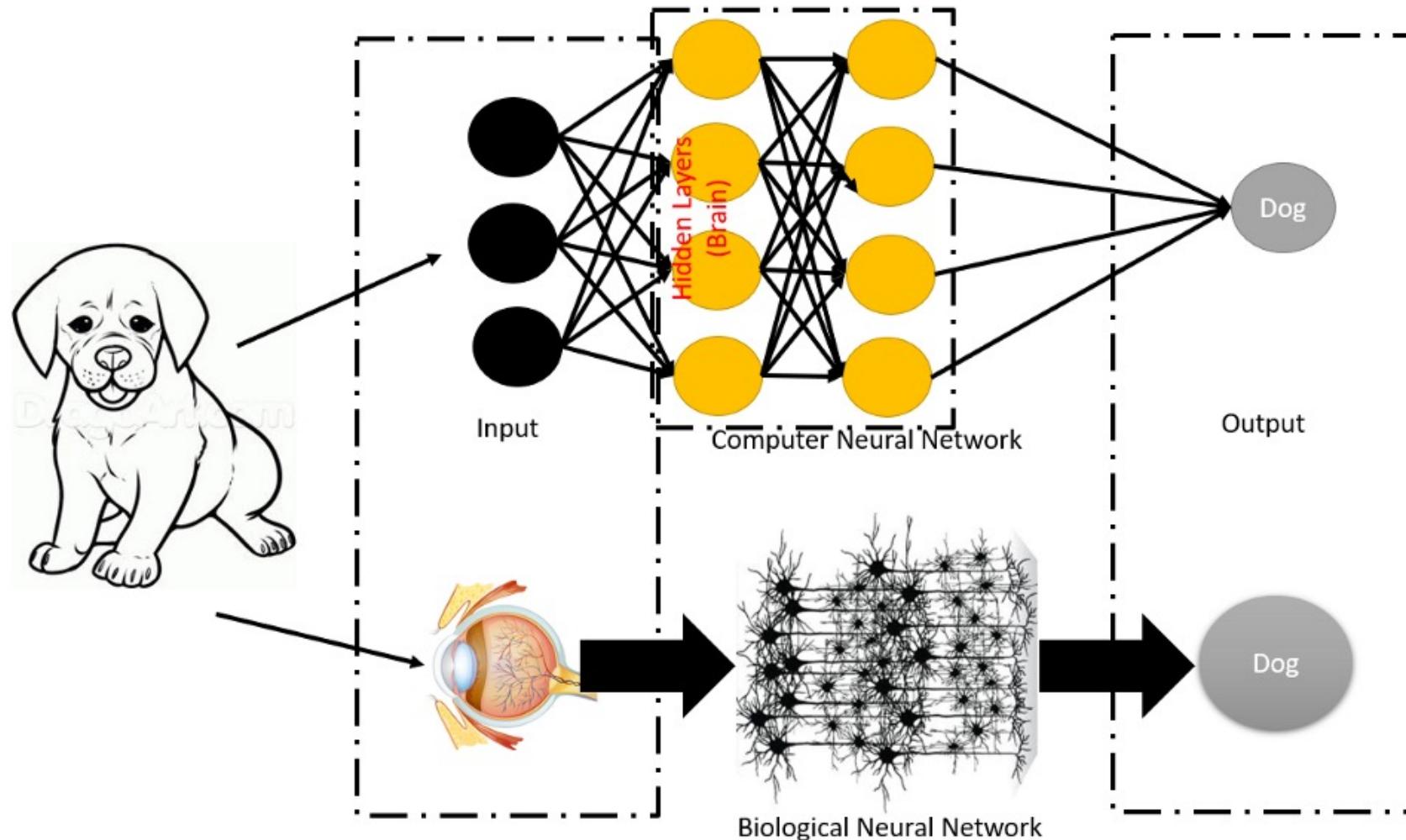


Neuron in our brain.

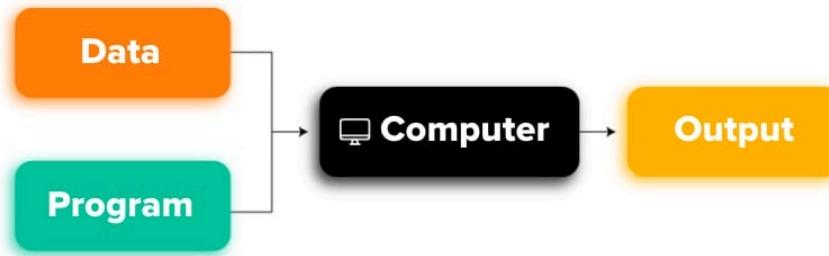


Neural Network develop using A.I.

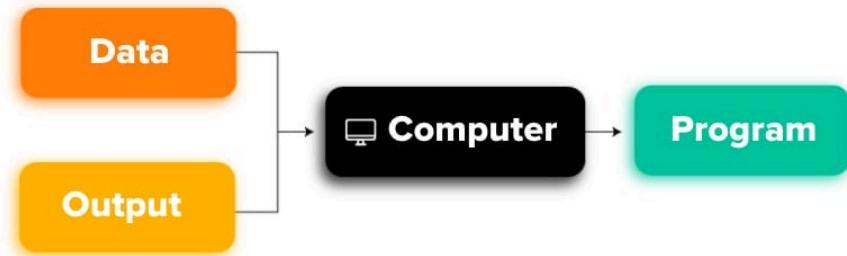
Neural Network and Deep Learning



TRADITIONAL PROGRAMMING

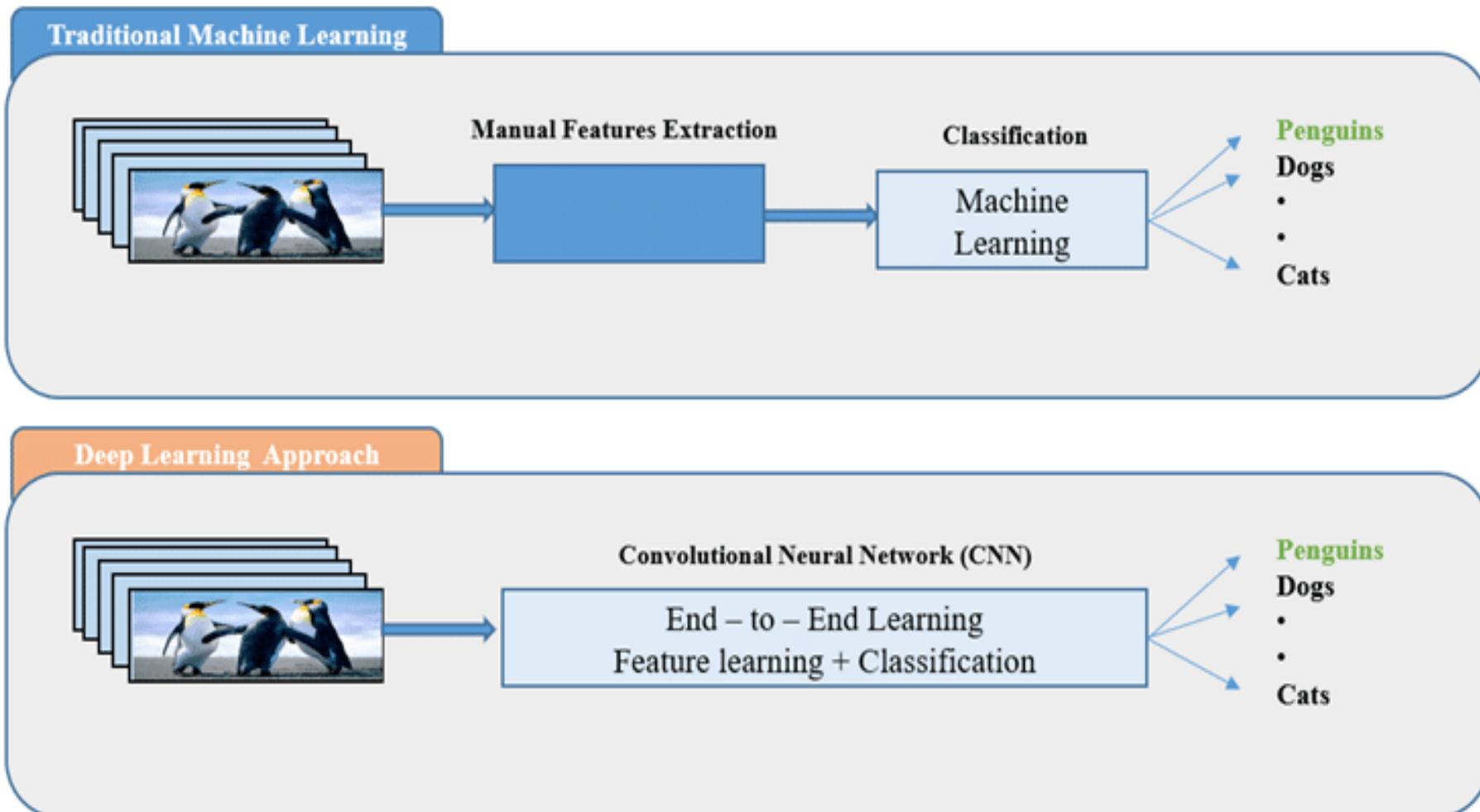


MACHINE LEARNING



towardsdatascience.com

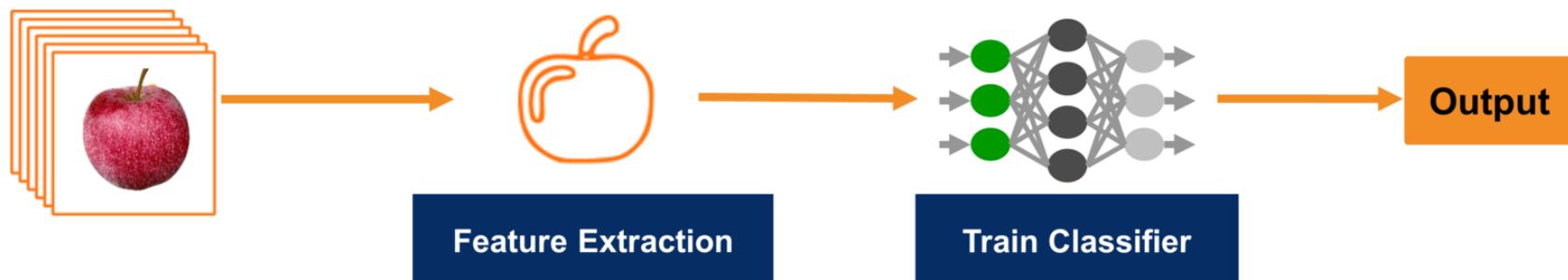
Machine Learning & Deep Learning



towardsdatascience.com

Machine Learning & Deep Learning

Classic Machine Learning



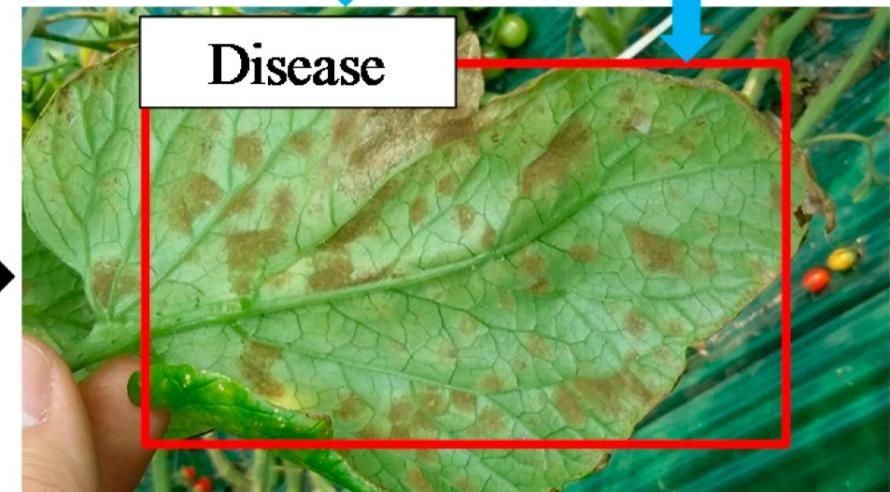
Deep Learning



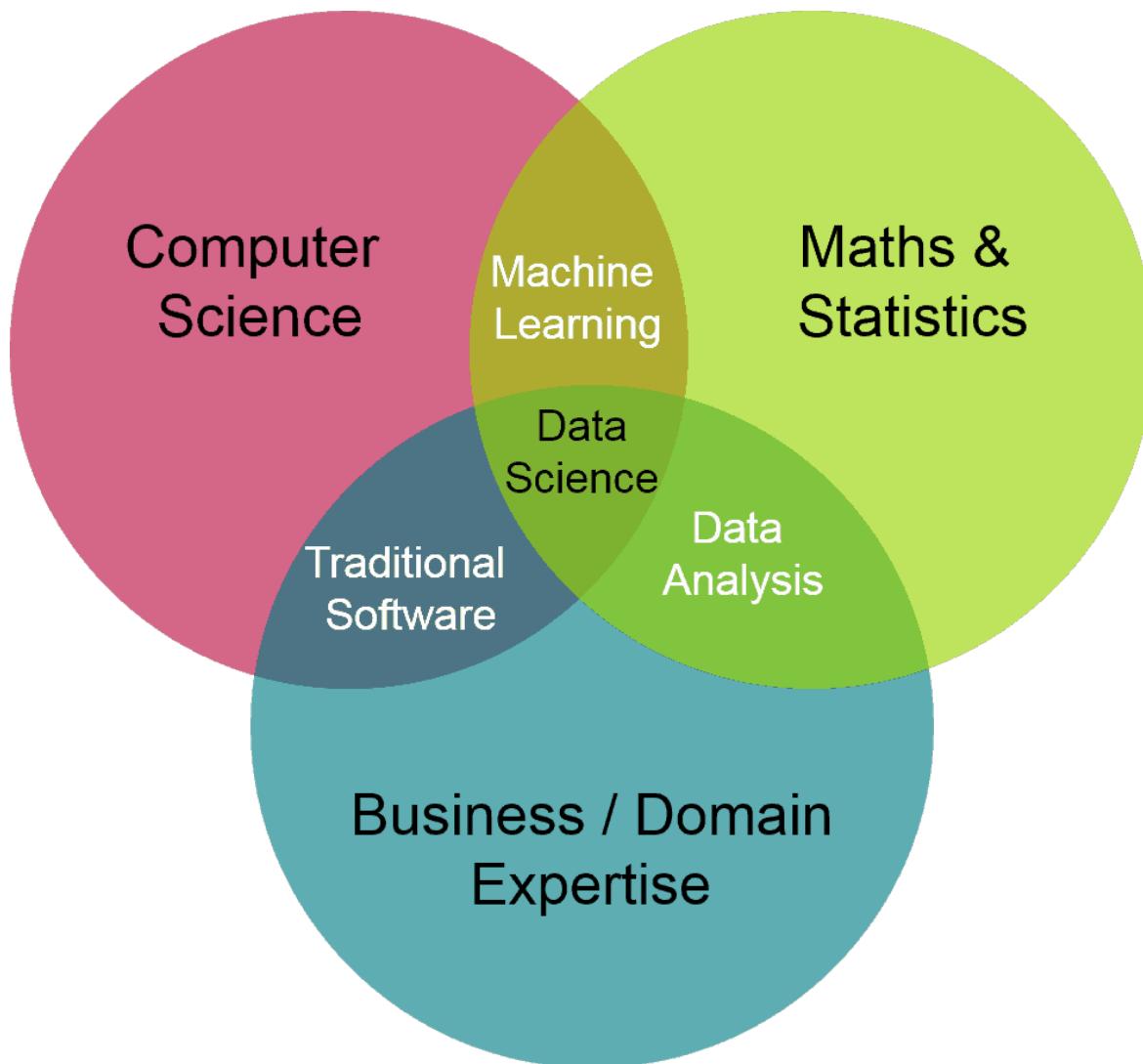
towardsdatascience.com



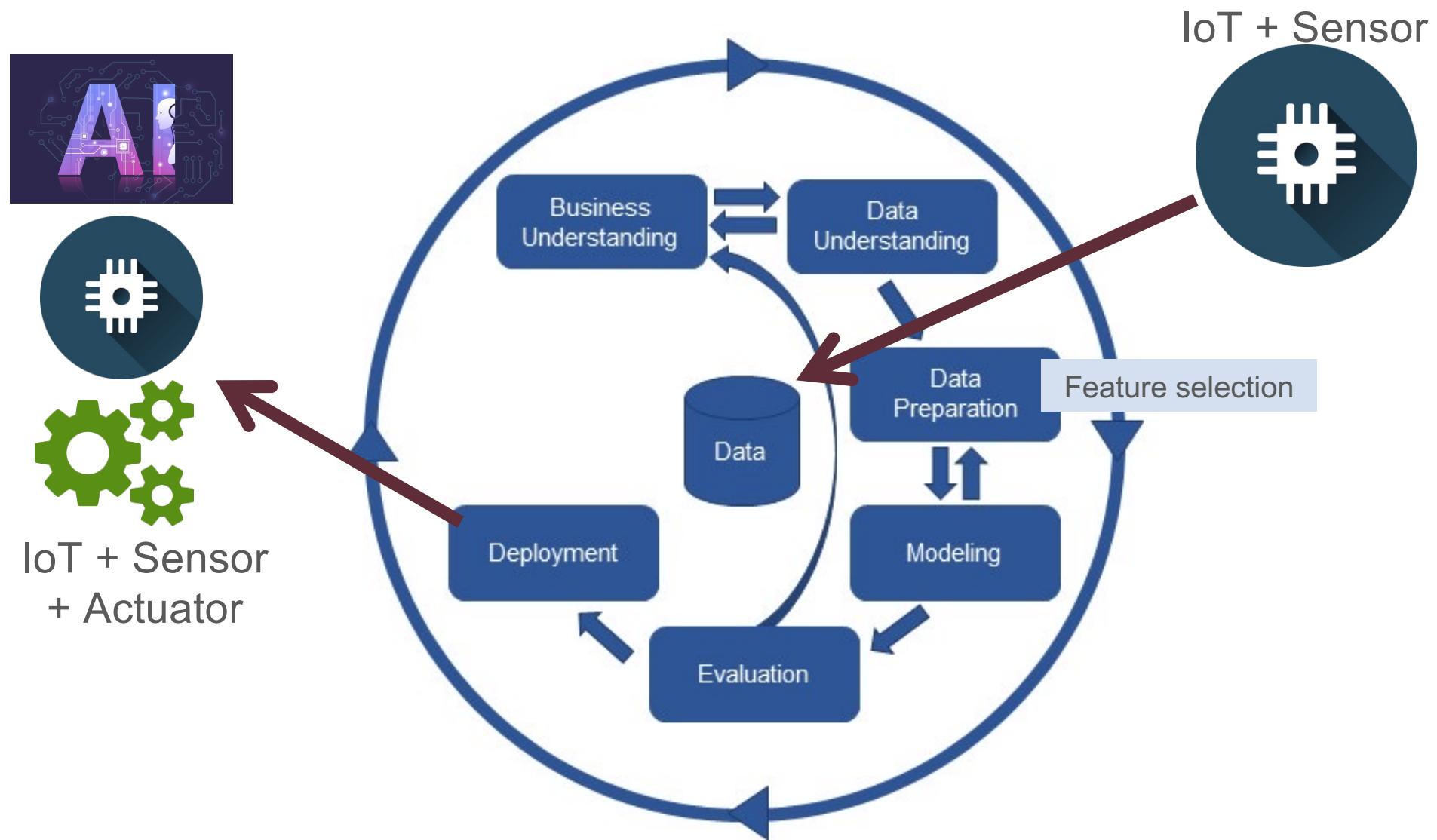
Input



Desired output



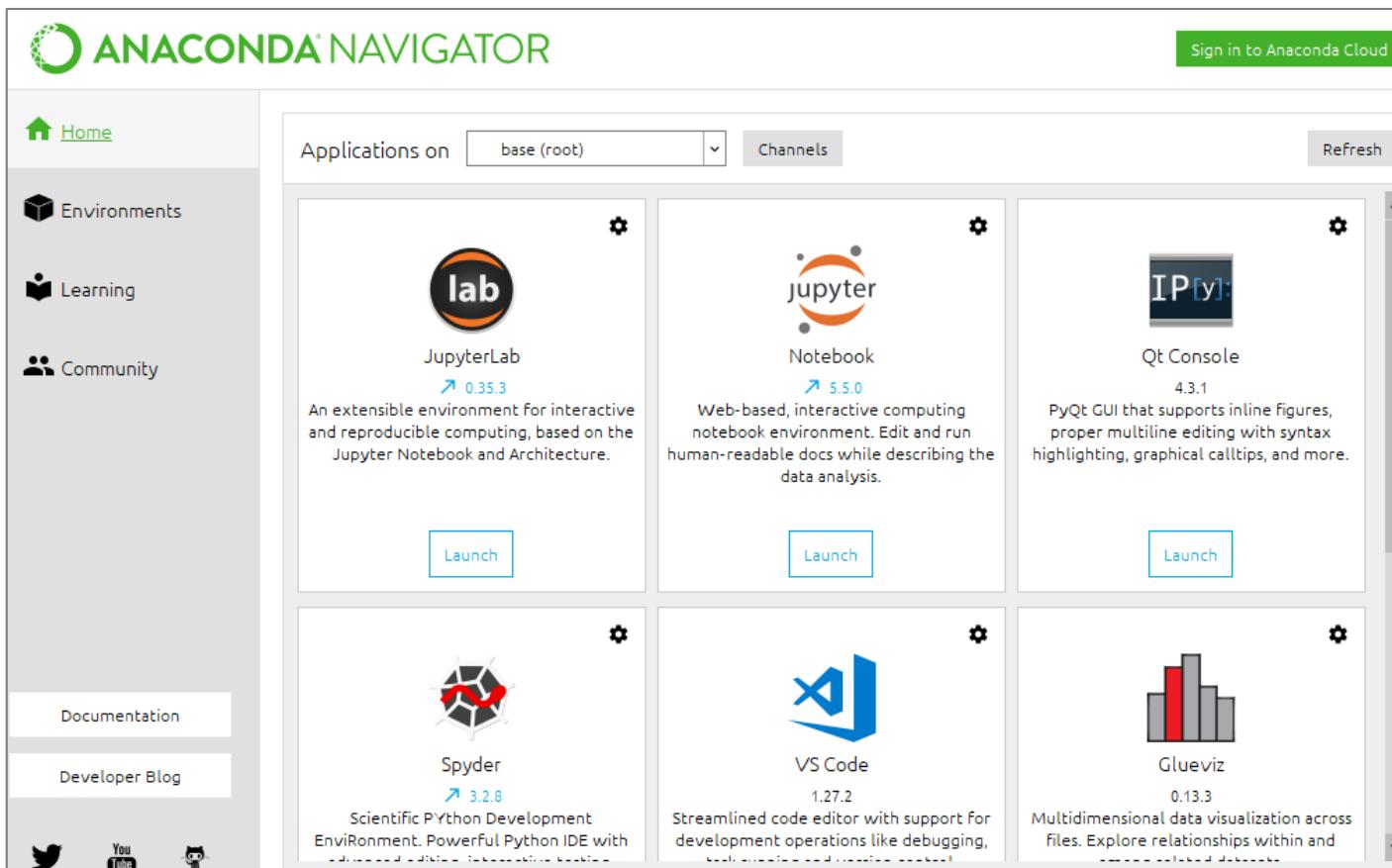
Data Science Life Cycle



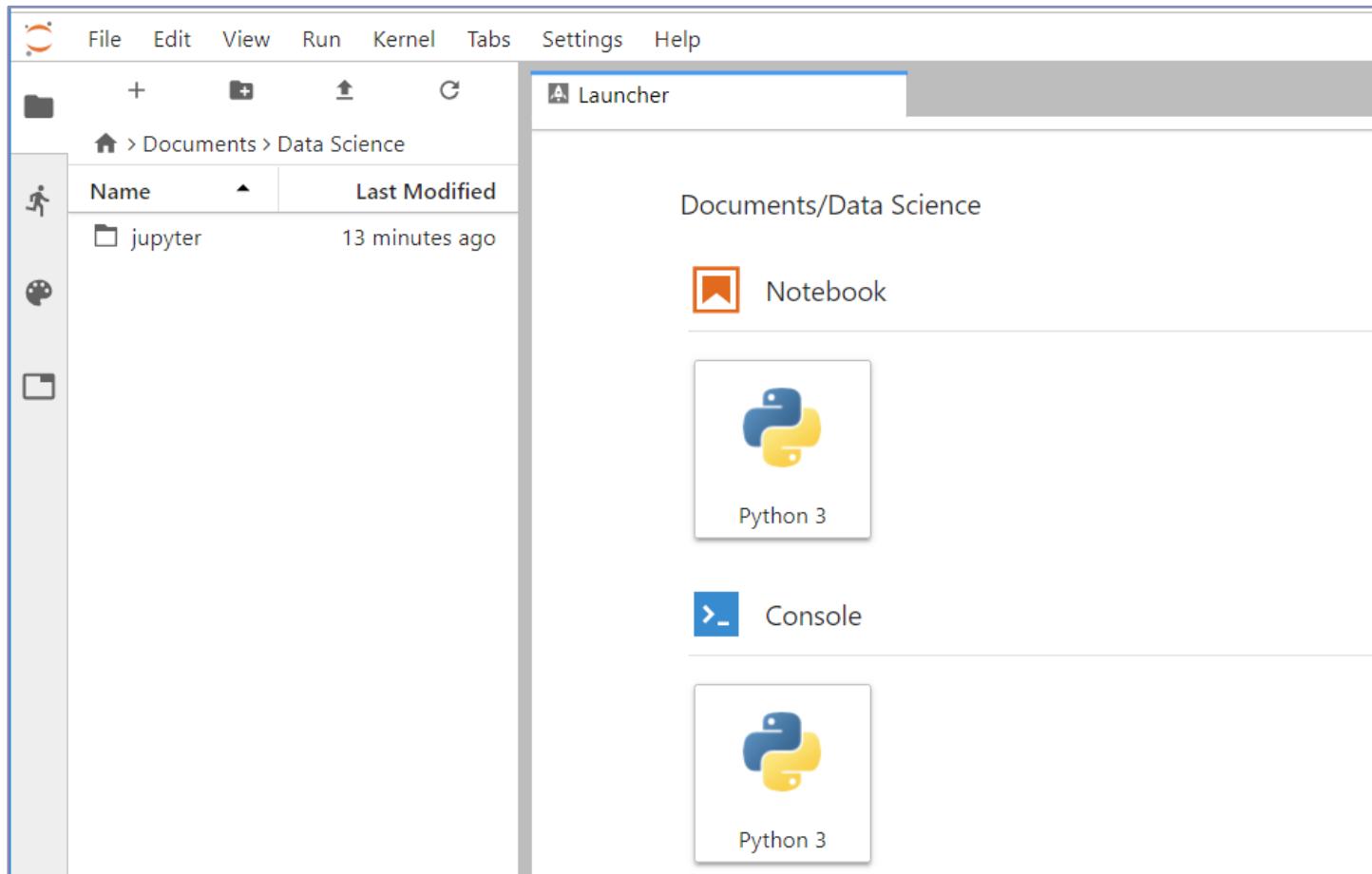
Installation



- Anaconda.com
- Download installer -> <https://www.anaconda.com/products/distribution>
- เลือกแพลตฟอร์มตามที่ใช้งาน เลือกเวอร์ชัน Python 3.x



Jupyter Lab/Notebook



Google Colab

- ❑ colab.research.google.com
- ❑ Log in (Use your Google account)



พื้นที่ทดลอง Matplotlib

แสดงยอดขายรายเดือนของสินค้า

```
[9]: %matplotlib inline
import matplotlib.pyplot as plt
y = [3,2,4,3,4]
x = ['Jan','Feb','Mar','Apr','May']
plt.bar(x, y)
```

```
[9]: <BarContainer object of 5 artists>
```

Month	Sales
Jan	3.0
Feb	2.0
Mar	4.0
Apr	3.0
May	4.0

Untitled1.ipynb

All changes saved

```
[1] import matplotlib.pyplot as plt
y = [3,2,4,3,4]
x = ['Jan','Feb','Mar','Apr','May']
plt.bar(x, y)
```

<BarContainer object of 5 artists>

Month	Sales
Jan	3.0
Feb	2.0
Mar	4.0
Apr	3.0
May	4.0

ប័ណ្ណលម្អិត



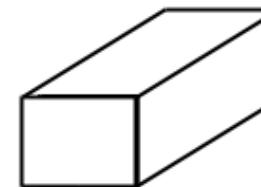
Vector
(1D Tensor)

-1
2
7
19
-5
0.5
1.9

Matrix
(2D Tensor)

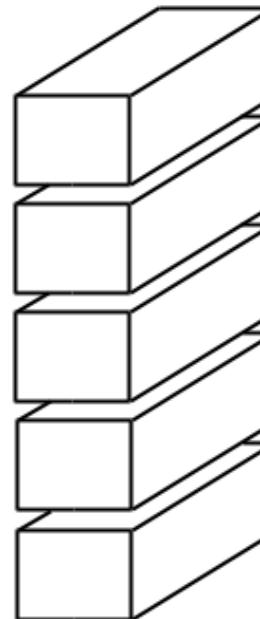
-1	-5	84	5
2	0.5	56	7
7	1.9	1	8.4
19	6	8	0.3

Matrixes
(3D Tensor)

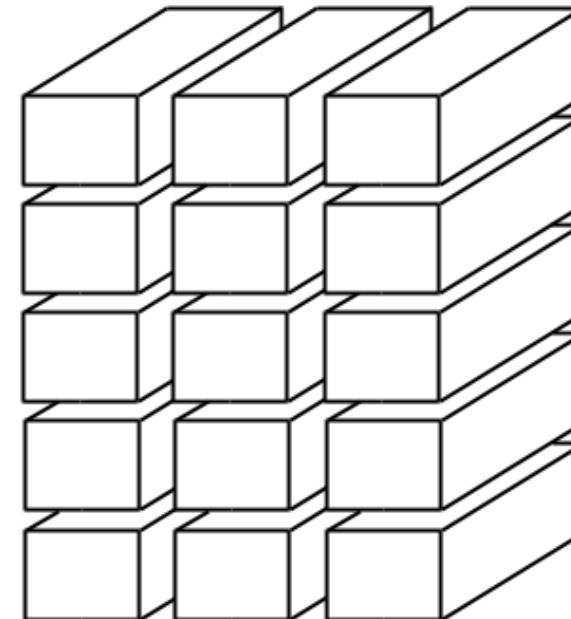


-1	-5	84	5
2	0.5	56	7
7	1.9	1	8.4
19	6	8	0.3

Vector of Matrixes
(4D Tensor)



Matrix of Matrixes
(5D Tensor)



Numpy Array

Scalars, Vectors and Matrices

And when we include **matrices** we get this interesting pattern:

- A **scalar** is a number, like **3, -5, 0.368, etc,**
- A **vector** is a **list** of numbers (can be in a row or column),
- A **matrix** is an **array** of numbers (one or more rows, one or more columns).

Scalar

24

Vector

$$\begin{bmatrix} 2 & -8 & 7 \end{bmatrix}$$

row

or
column

$$\begin{bmatrix} 2 \\ -8 \\ 7 \end{bmatrix}$$

Matrix

$$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}$$

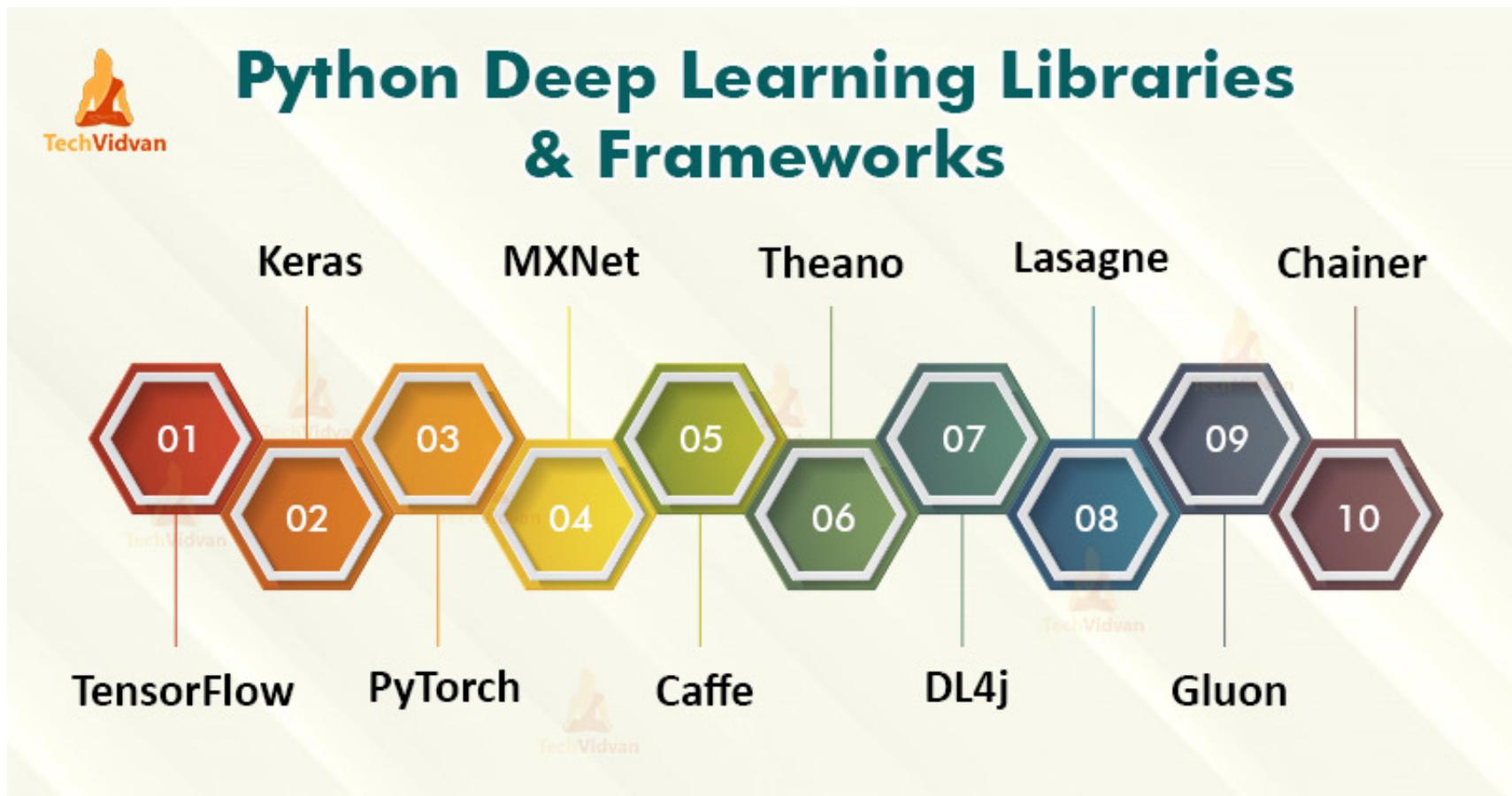
row(s) × column(s)

$$\begin{bmatrix} 6 & 7 & 4 & 5 & 1 \\ 2 & 8 & 3 & 6 & 4 \\ 1 & 3 & 2 & 9 & 6 \\ 8 & 9 & 1 & 7 & 2 \end{bmatrix}$$

In fact a **vector is also a matrix!** Because a matrix can have just one row or one column.

So the rules that work for matrices also work for vectors.

Deep Learning Libraries & Tools



AI Deep Learning Frameworks/Platform

theano



TensorFlow



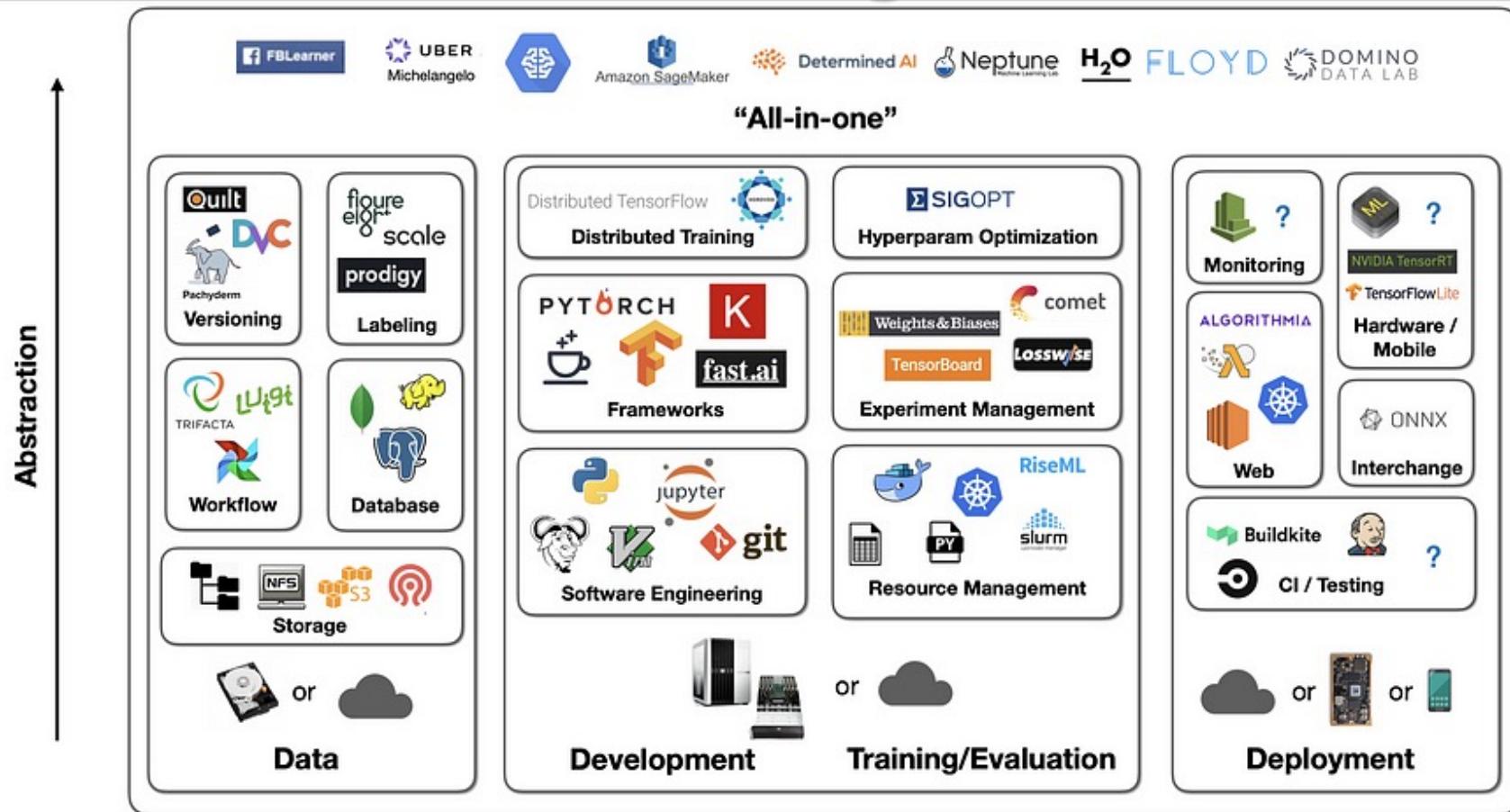
Keras

Caffe



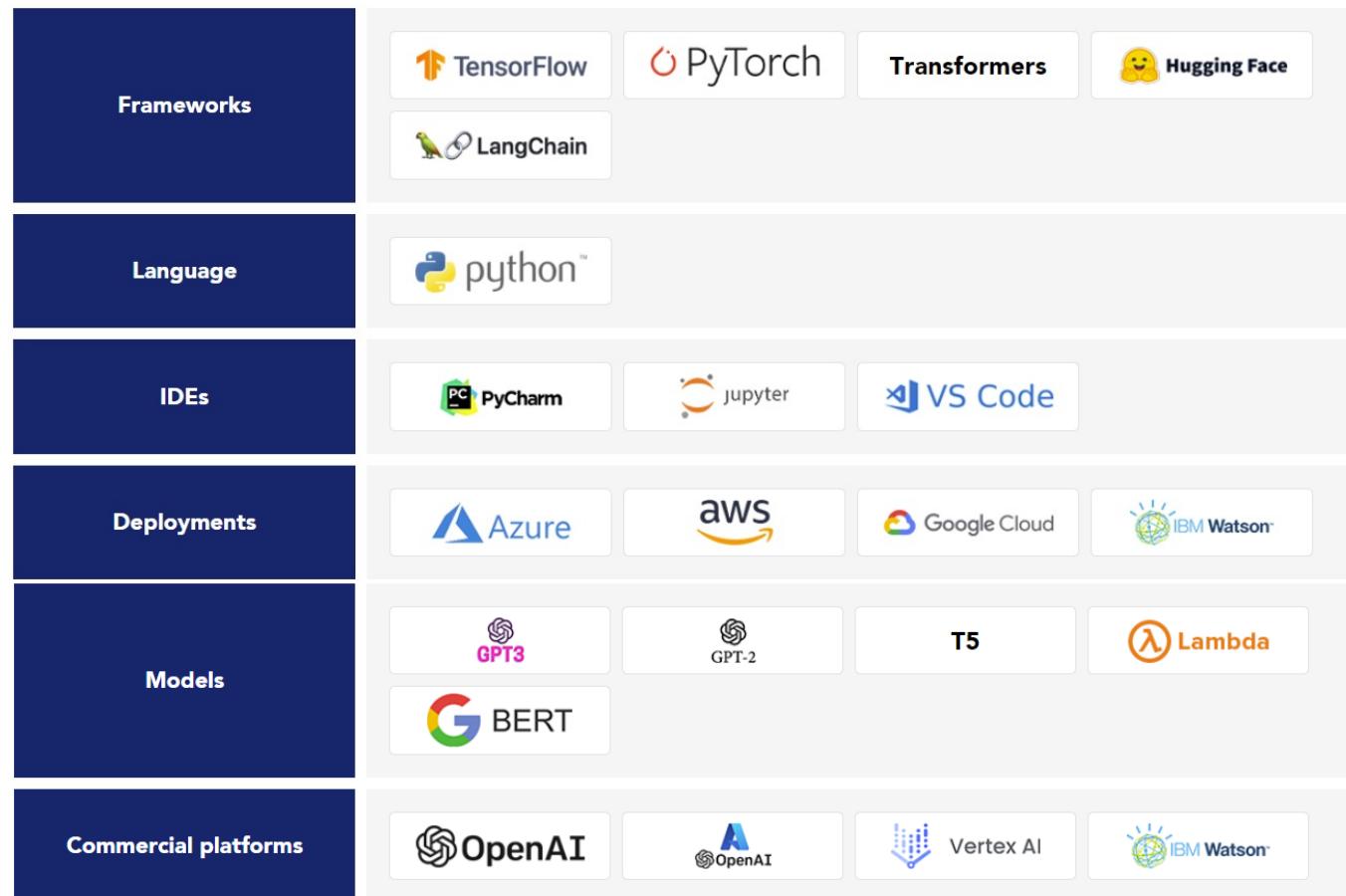
PyTorch

Useful Machine Learning Tools

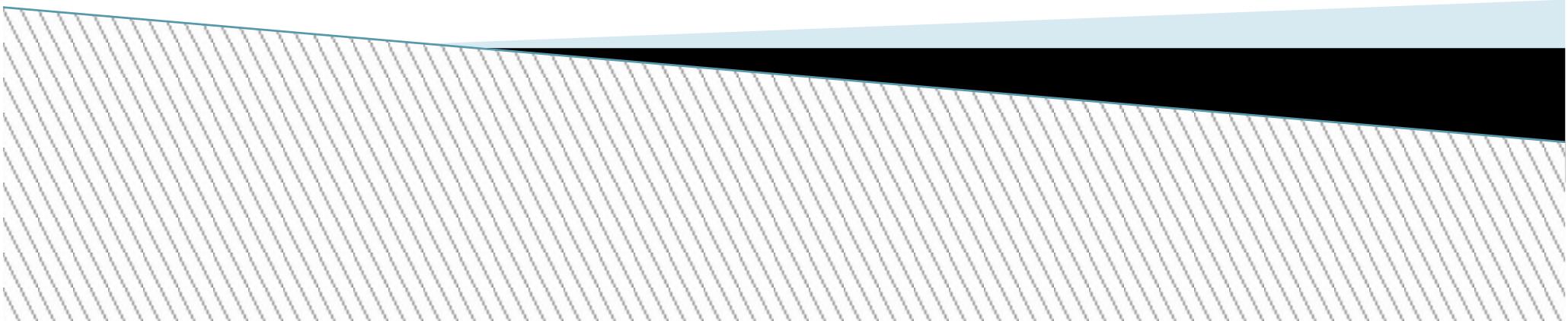


kdnuggets.com

Generative AI Tech Stack



Convolutional Neural Network (CNN)





(a)



(b)



(c)



(d)



Input



Desired output

Recognition
What?

Localization
Where?

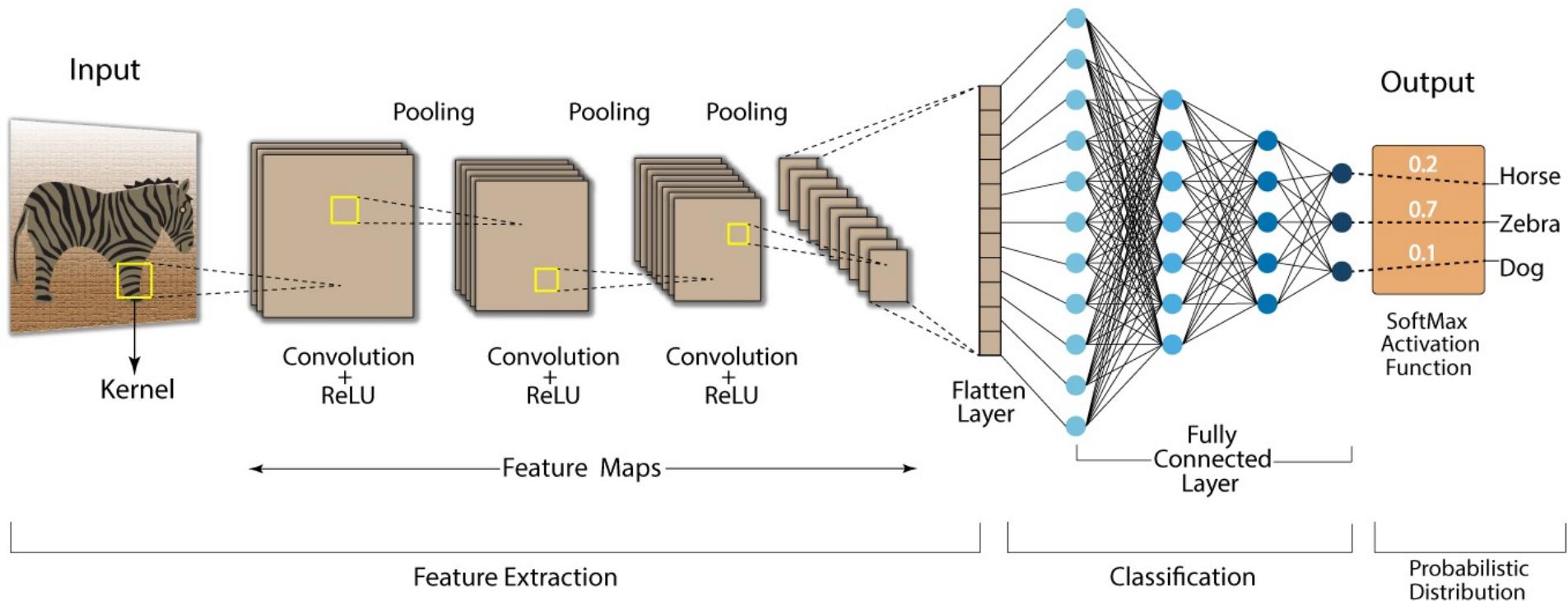
Disease



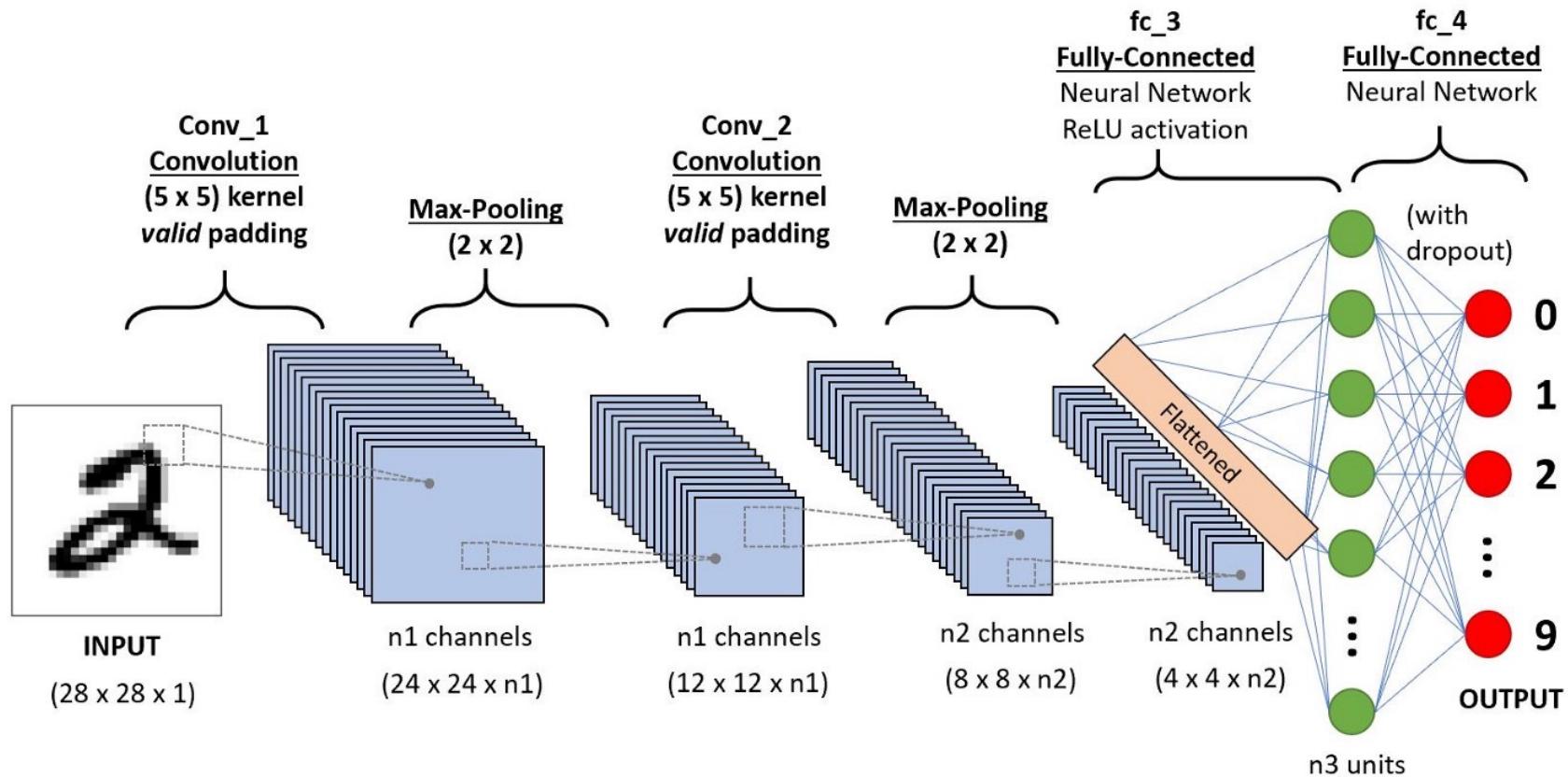
Convolutional Neuron Network (CNN)



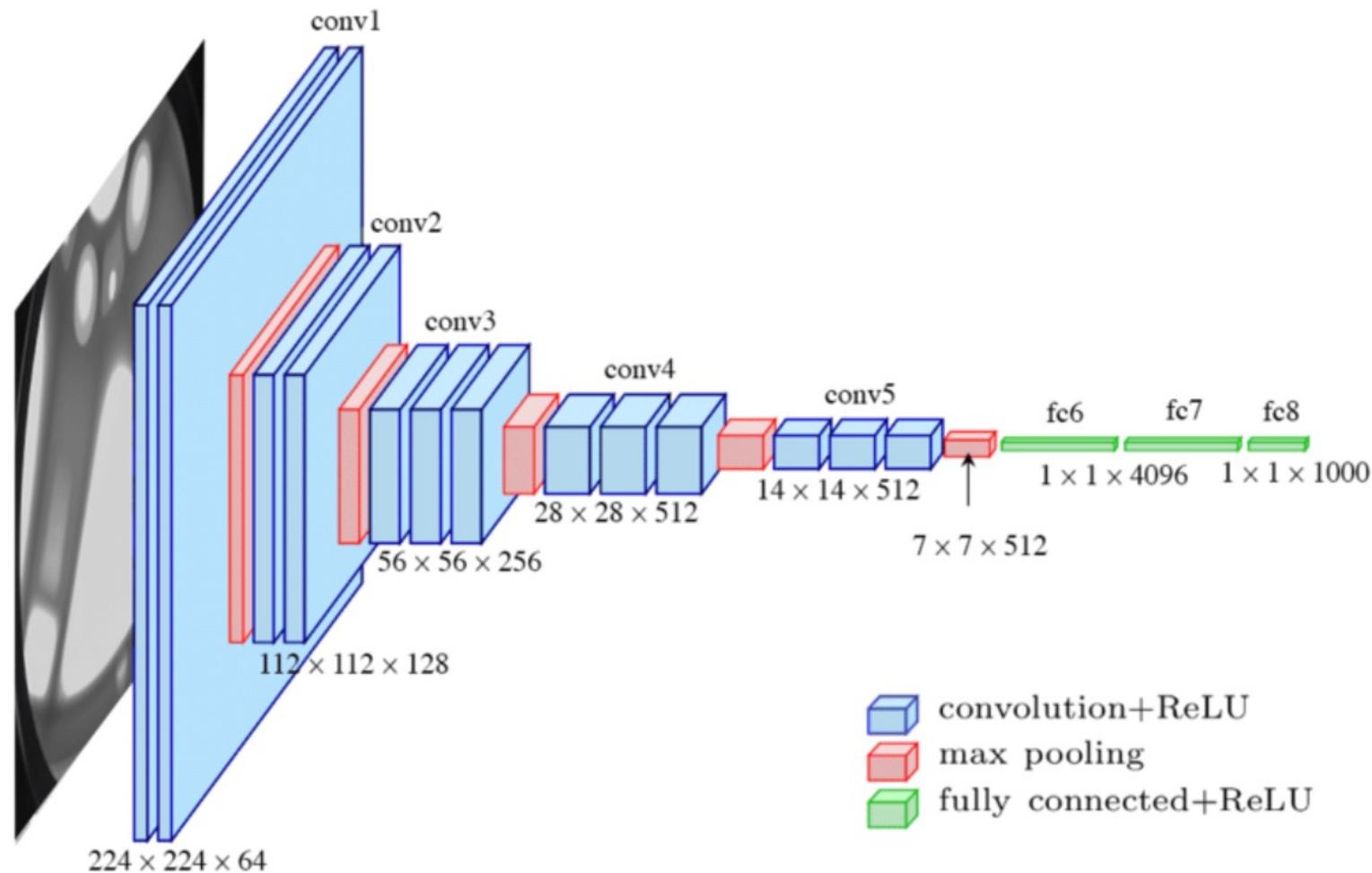
Convolution Neural Network (CNN)



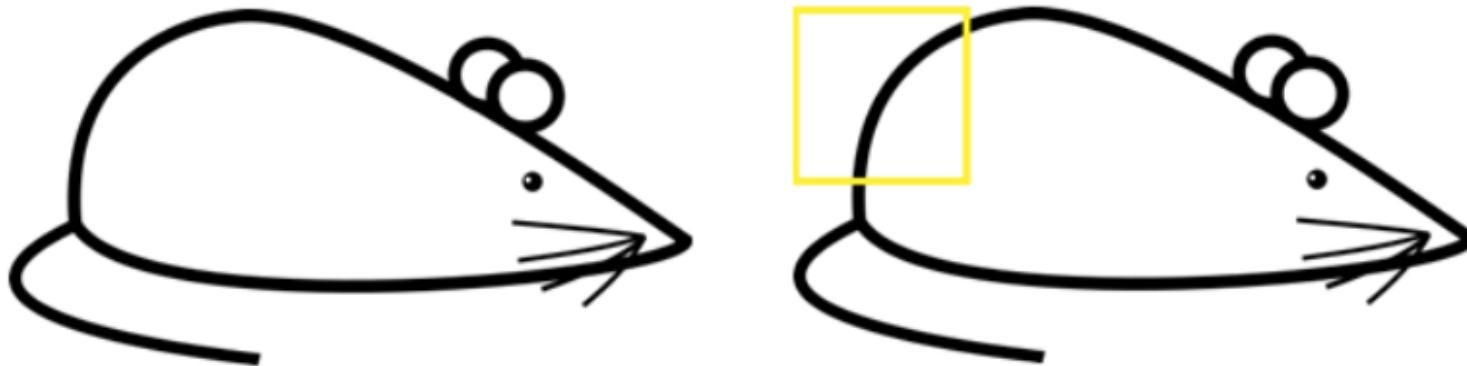
CNN



CNN architecture



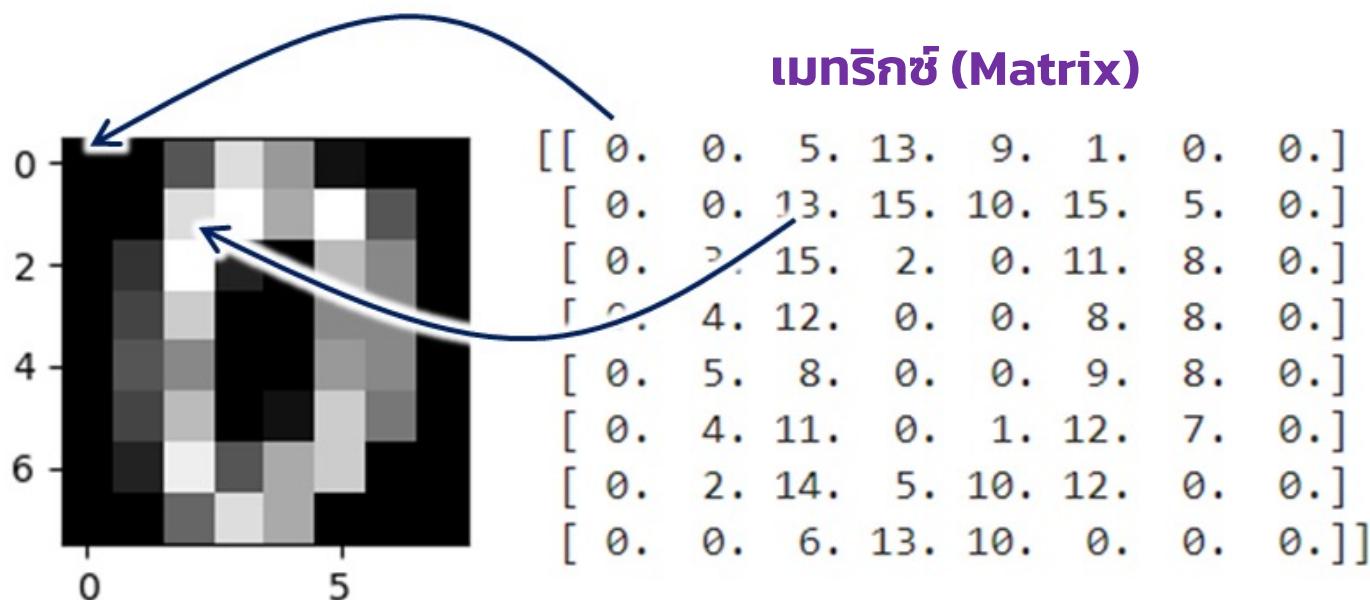
Feature Extraction



Source: <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>

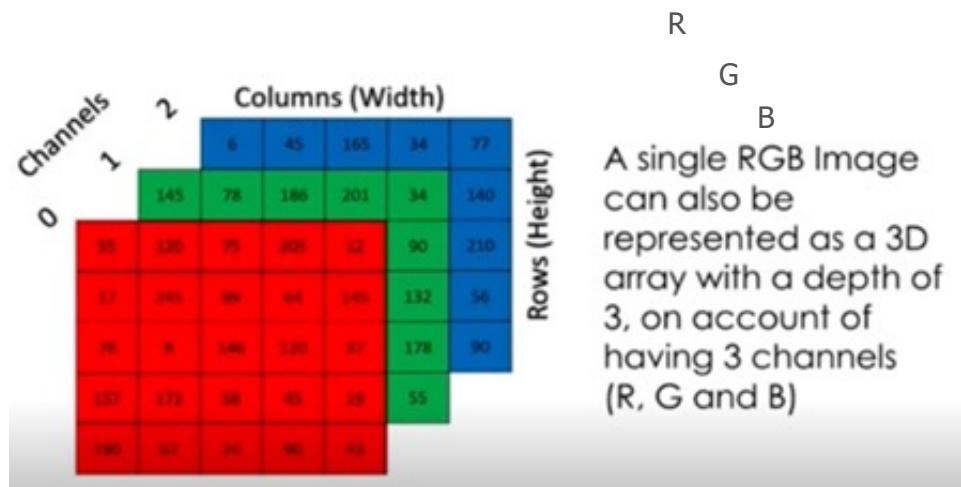
ข้อมูลภาพ

Pixel โภนเทา (Grayscale)



ข้อมูลภาพสี

ข้อมูลชุด: คล้ายเมทริกซ์ (3D Tensor)



Color Image

Convolution

Source layer

5	2	6	8	2	0	1	2
4	3	4	5	1	9	6	3
3	9	2	4	7	7	6	9
1	3	4	6	8	2	2	1
8	4	6	2	3	1	8	8
5	8	9	0	1	0	2	3
9	2	6	6	3	6	2	1
9	8	8	2	6	3	4	5

Sample kernels: <https://www.mdpi.com/1099-4300/25/5/738>

Convolutional kernel

-1	0	1
2	1	2
1	-2	0

Destination layer

Feature map

$$\begin{aligned} & (-1 \times 5) + (0 \times 2) + (1 \times 6) + \\ & (2 \times 4) + (1 \times 3) + (2 \times 4) + \\ & (1 \times 3) + (-2 \times 9) + (0 \times 2) = 5 \end{aligned}$$

Padding

0	0	0	0	0	0	0	0
0	3	3	4	4	7	0	0
0	9	7	6	5	8	2	0
0	6	5	5	6	9	2	0
0	7	1	3	2	7	8	0
0	0	3	7	1	8	3	0
0	4	0	4	3	2	2	0
0	0	0	0	0	0	0	0

$6 \times 6 \rightarrow 8 \times 8$

*

1	0	-1
1	0	-1
1	0	-1

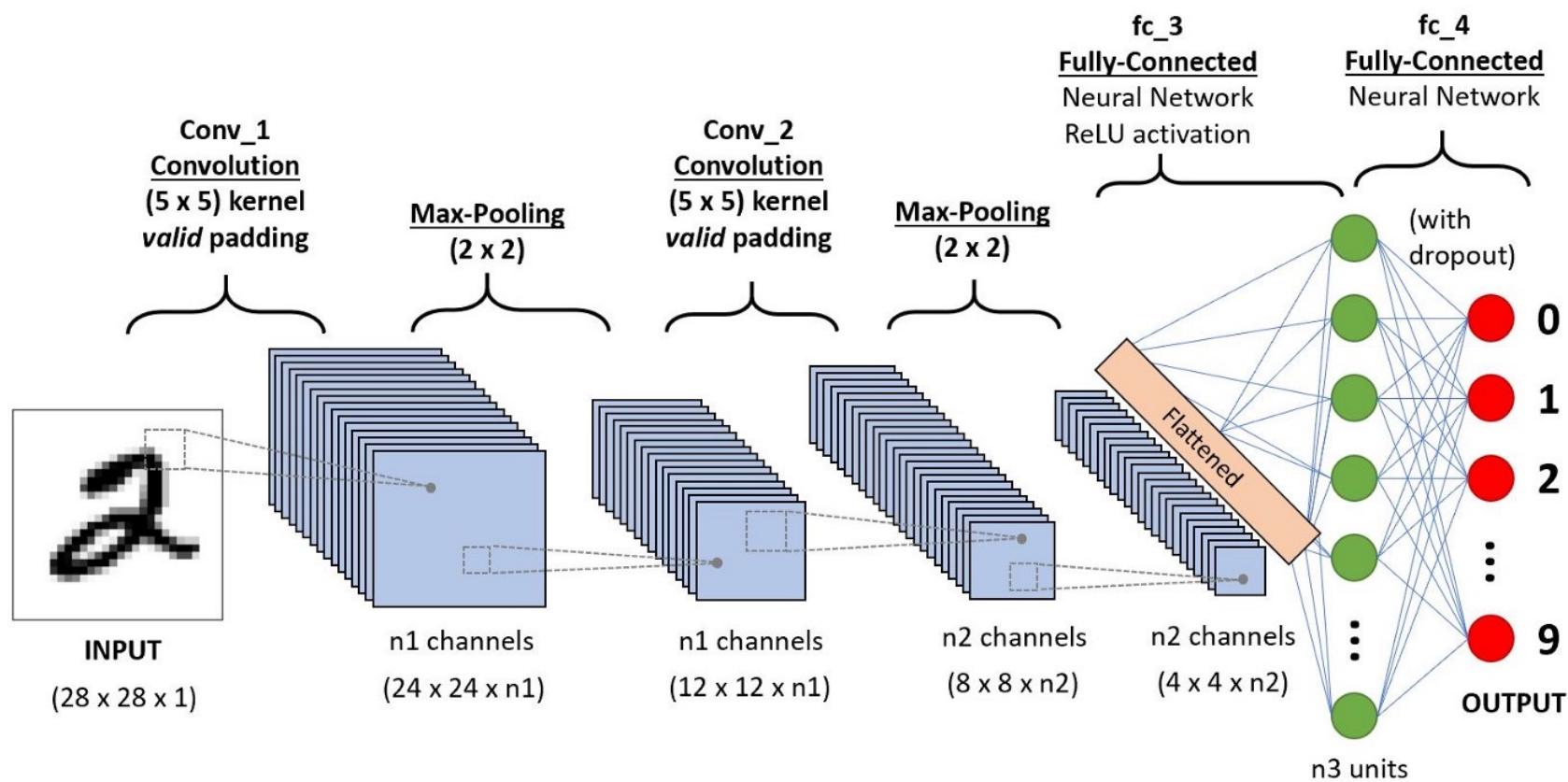
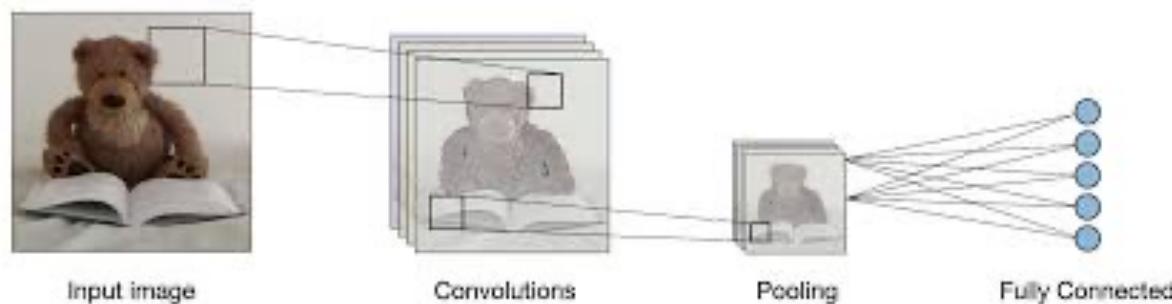
3×3

=

-10	-13	1			
-9	3	0			

6×6

Pooling



Pooling

Max Pooling			
29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2
pool size

100	184
12	45

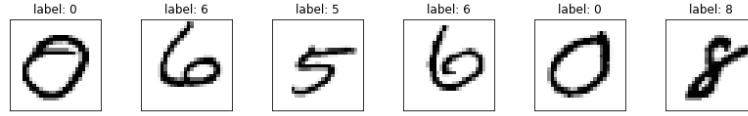
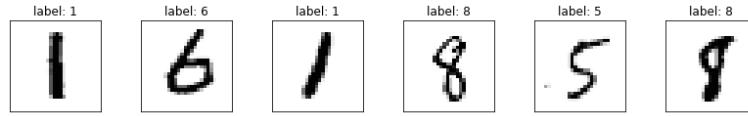
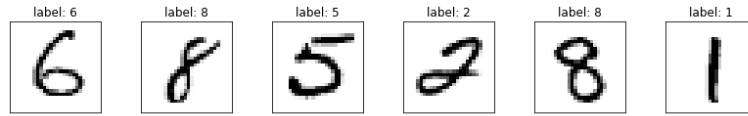
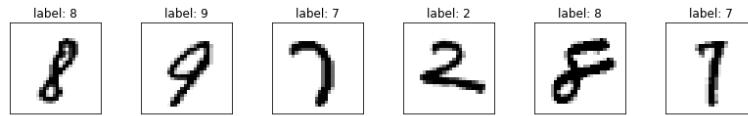
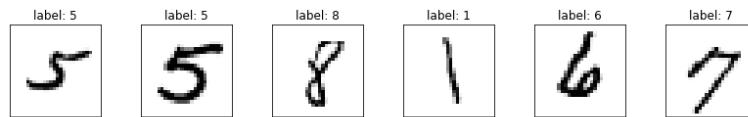
Average Pooling

Average Pooling			
31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

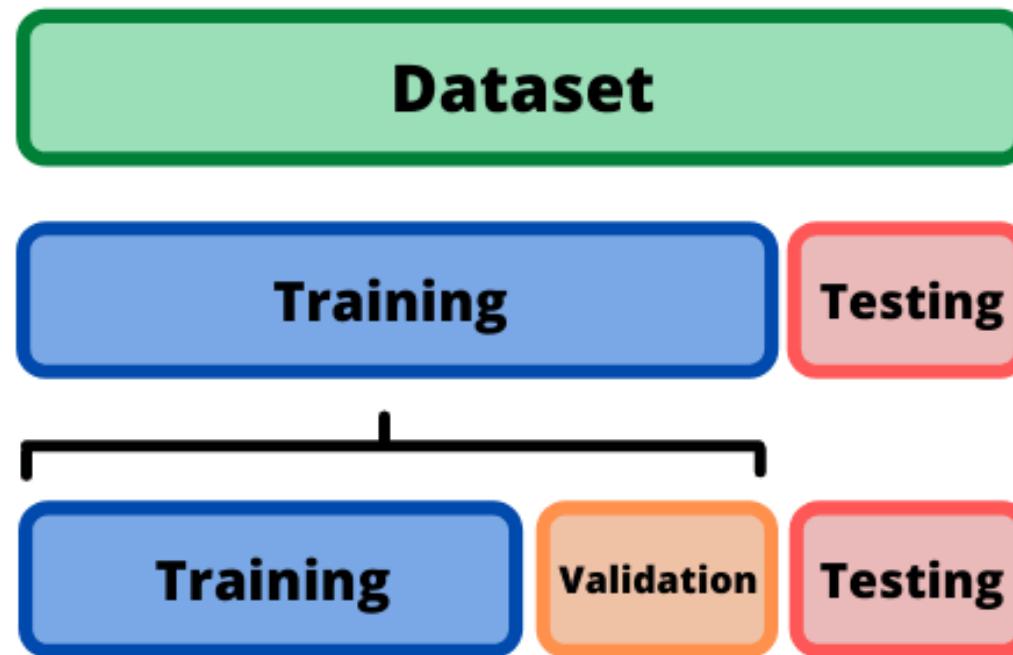
2 x 2
pool size

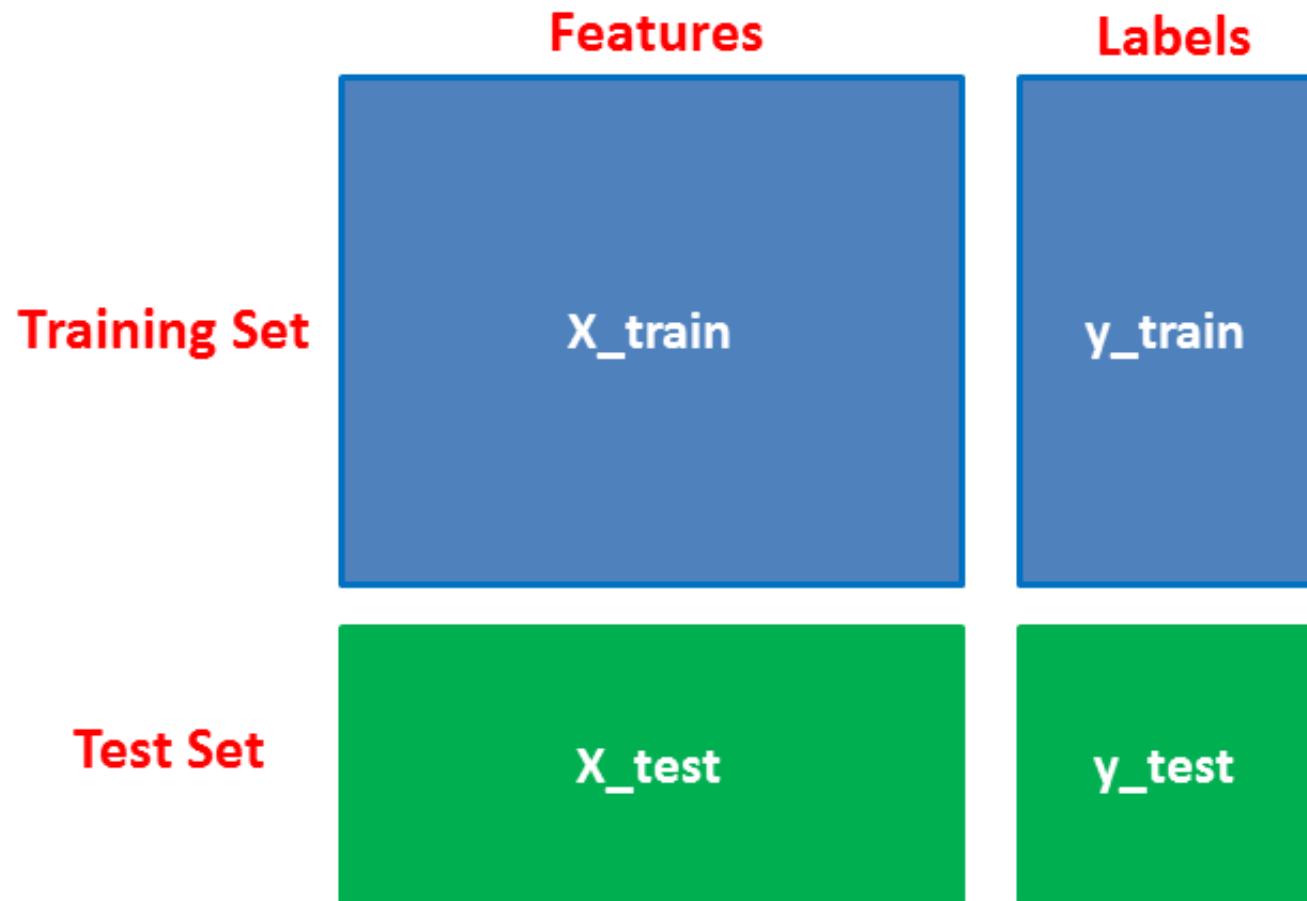
36	80
12	15

CNN Digit Recognition



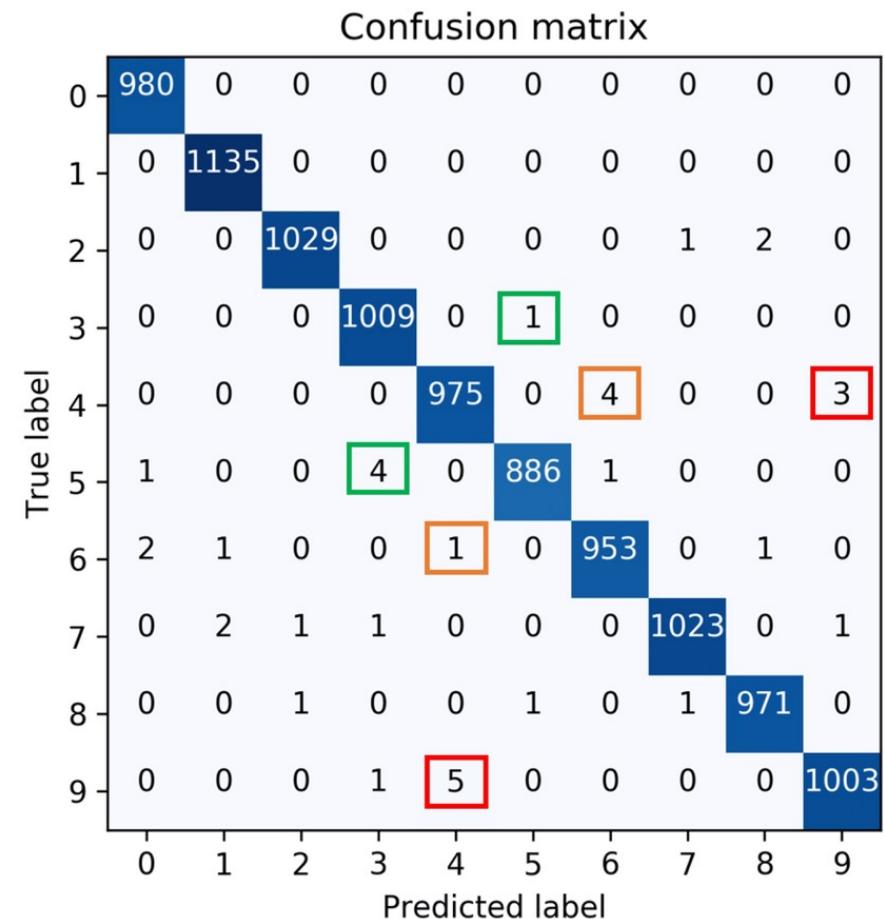
Train & Test





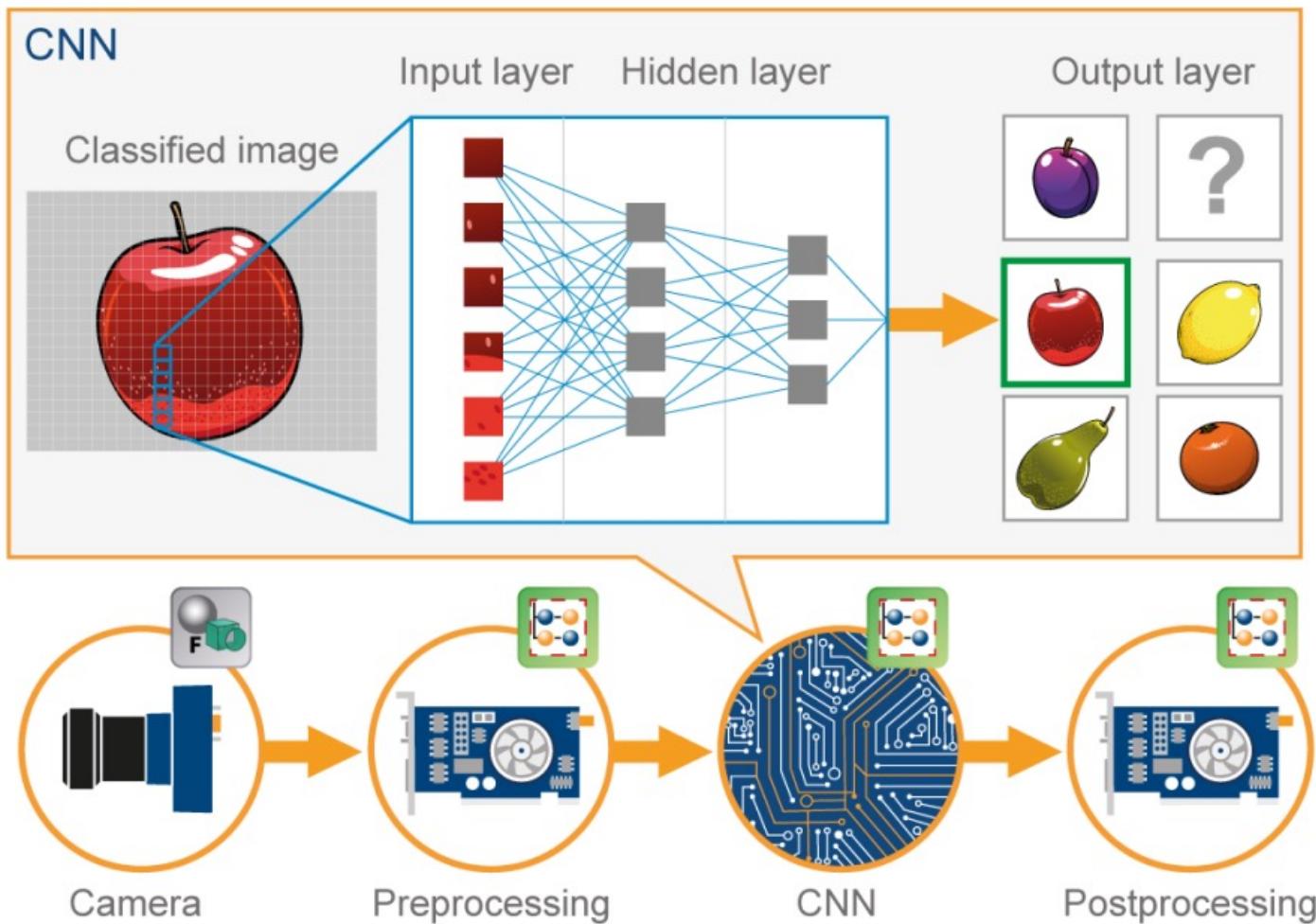
Model Evaluation

		ກຳນາຍ (Prediction)	
		Negative (0)	Positive (1)
ຂອງງວດ (Actual)	Negative (0)	True Negative (TN)	False Positive (FP)
	Positive (1)	False Negative (FN)	True Positive (TP)

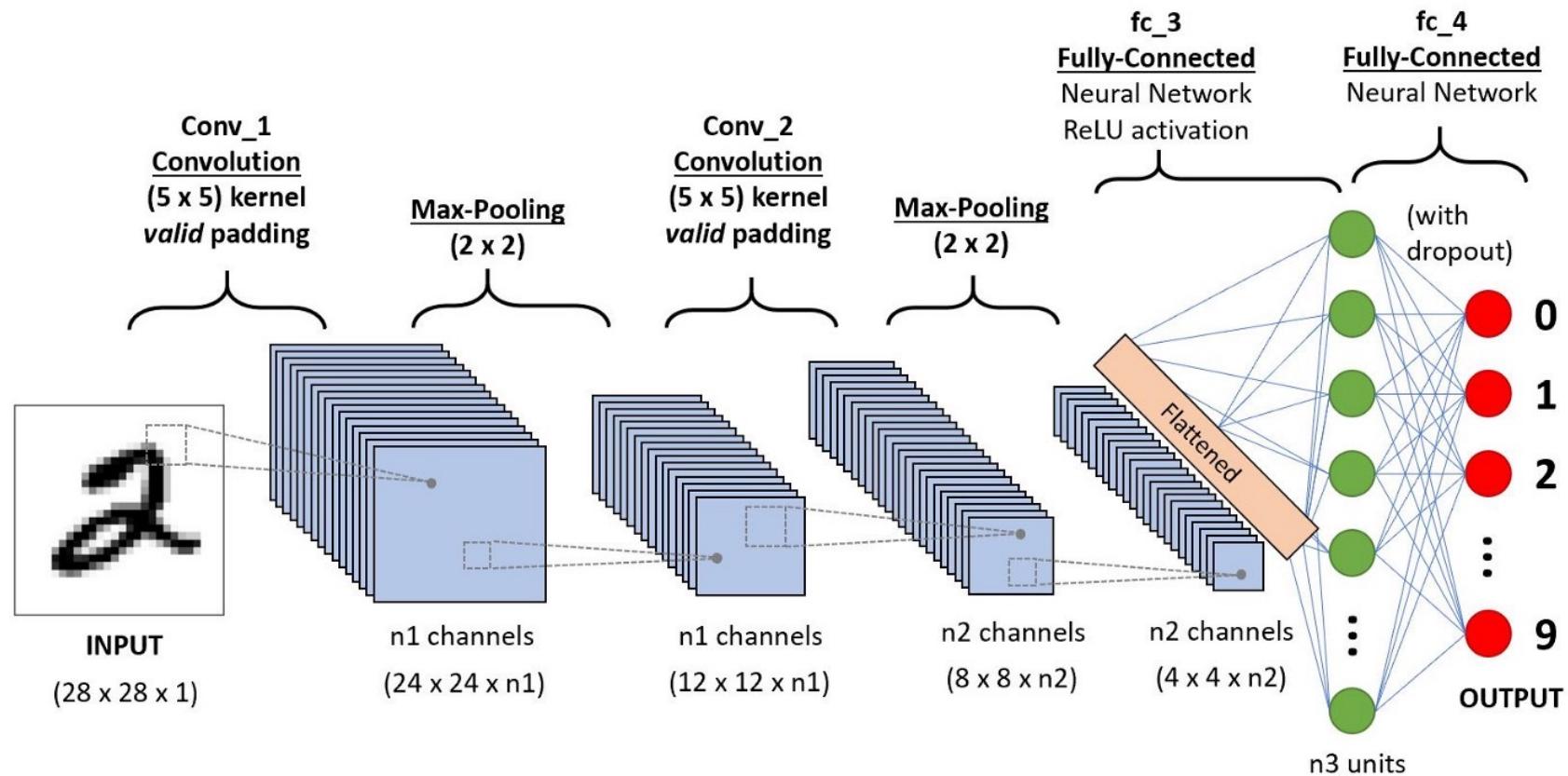


Confusion Matrix

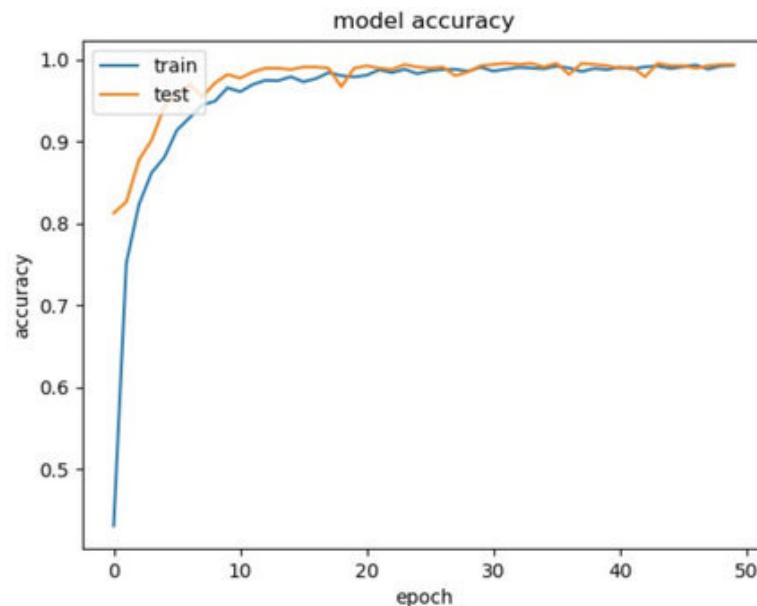
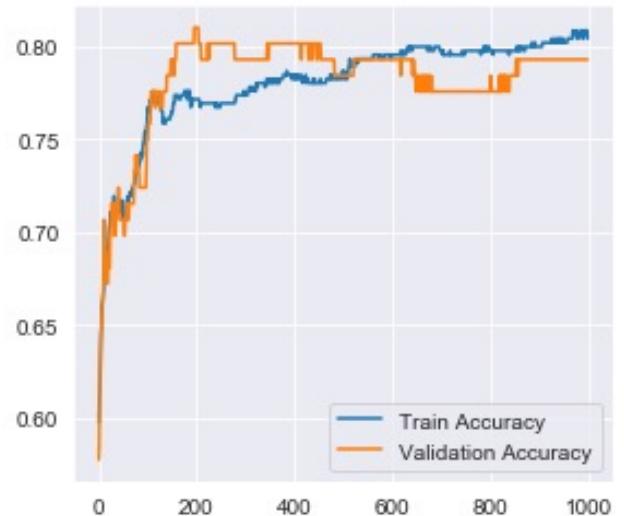
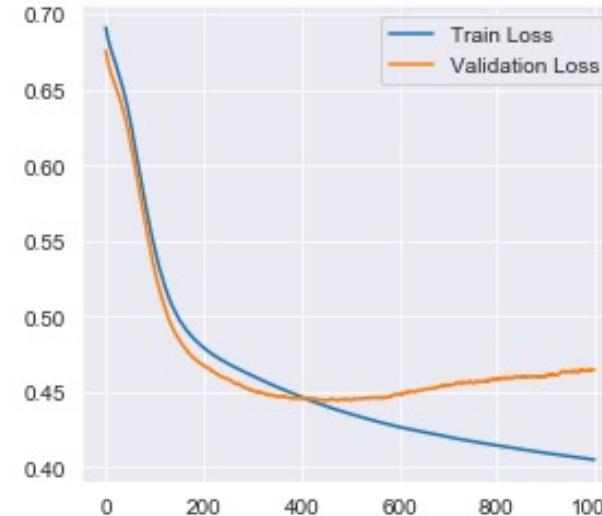
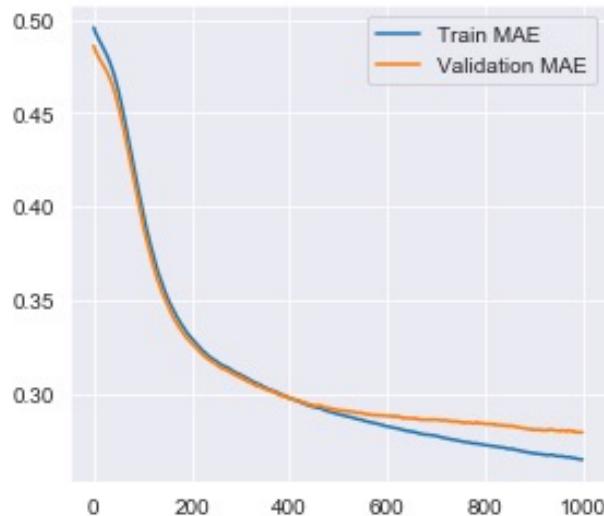
CNN Class



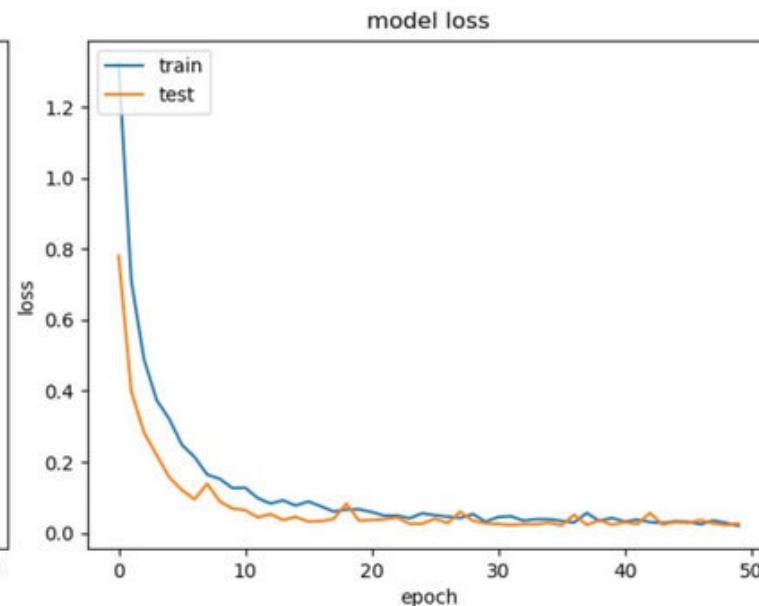
CNN Class



Accuracy and loss



(a)



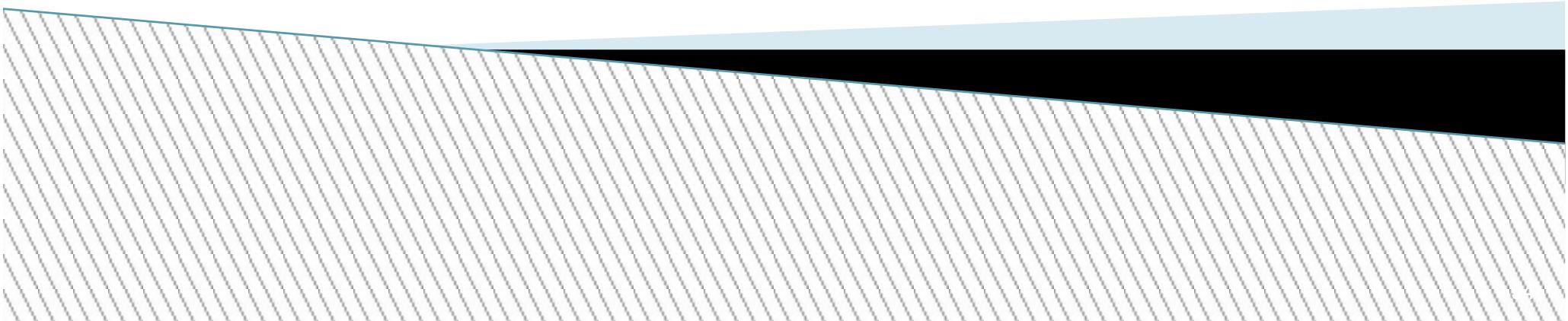
(b)



KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระบรมราชูปถัมภ์

Overfitting



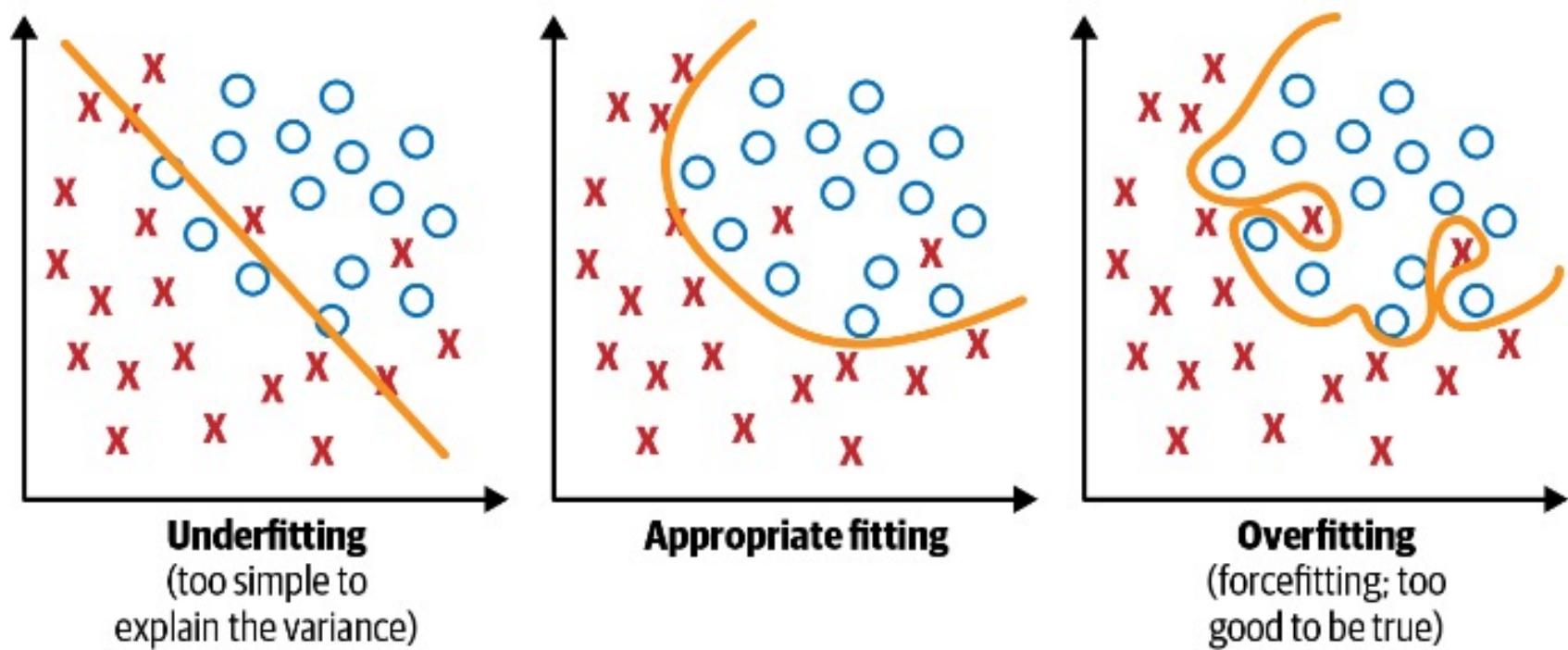
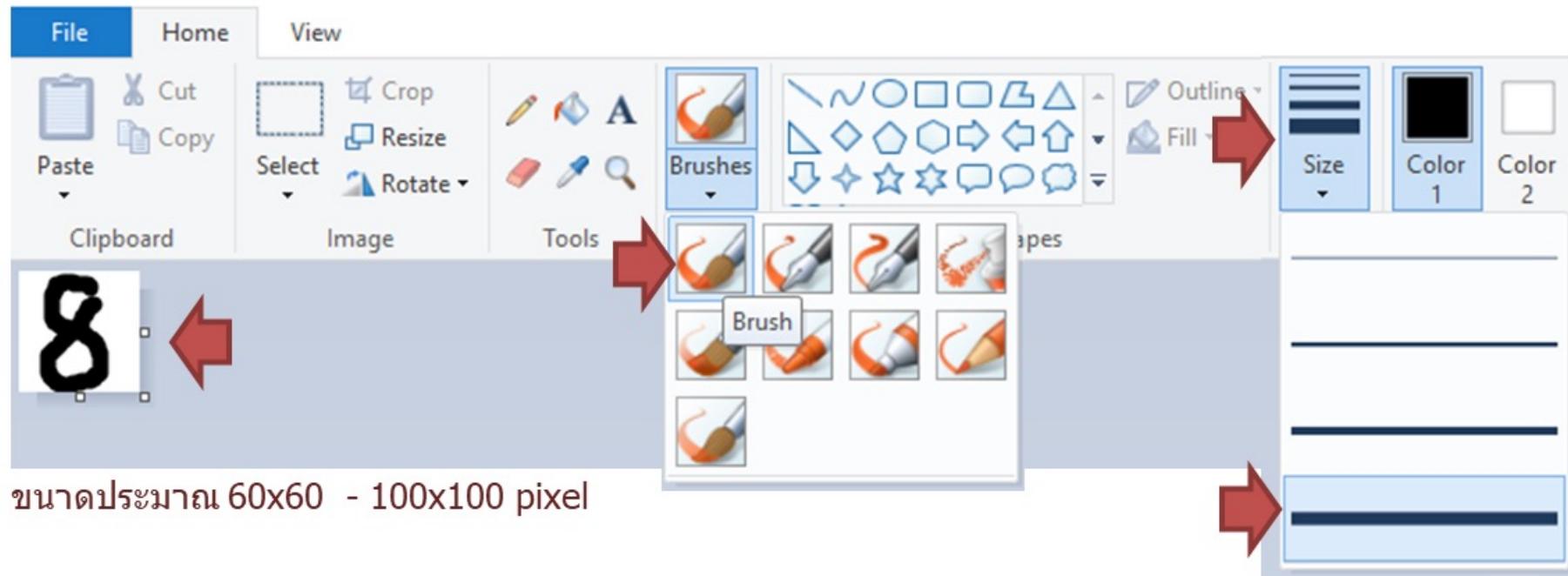


Figure 2-1. Overfitting and underfitting

Digits Prediction



Exercise..

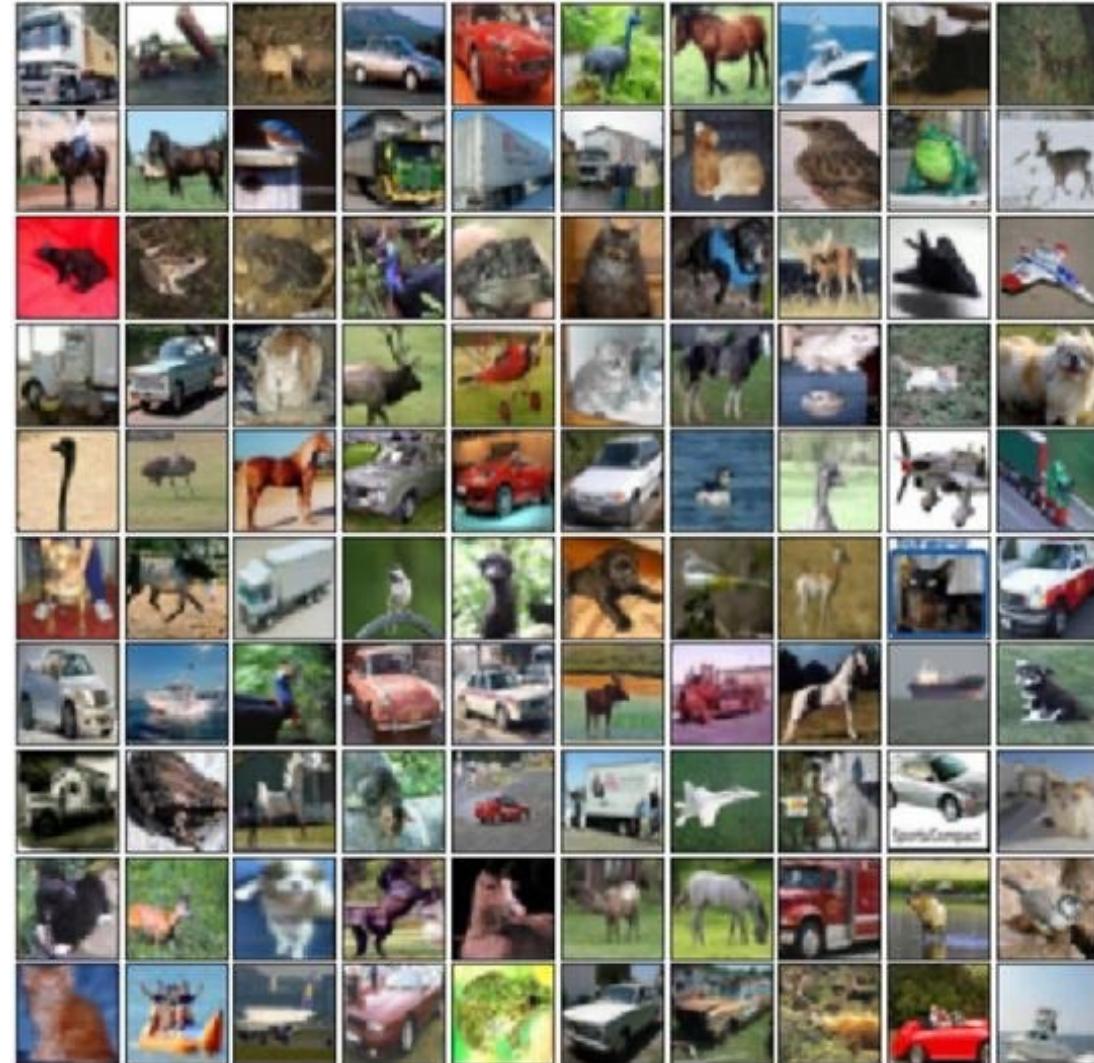
❑ Fashion mnist



```
from tensorflow.keras.datasets import fashion_mnist
(X_train, y_train), (X_test, y_test) = fashion_mnist.load_data()
```

Challenge..

❑ cifar10 mnist



```
from tensorflow.keras.datasets import cifar10  
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
```