HANDS ON PRACTICE IN

MACHINE LEARNING

<DEEP LEARNING>

By
Piyali Karmakar

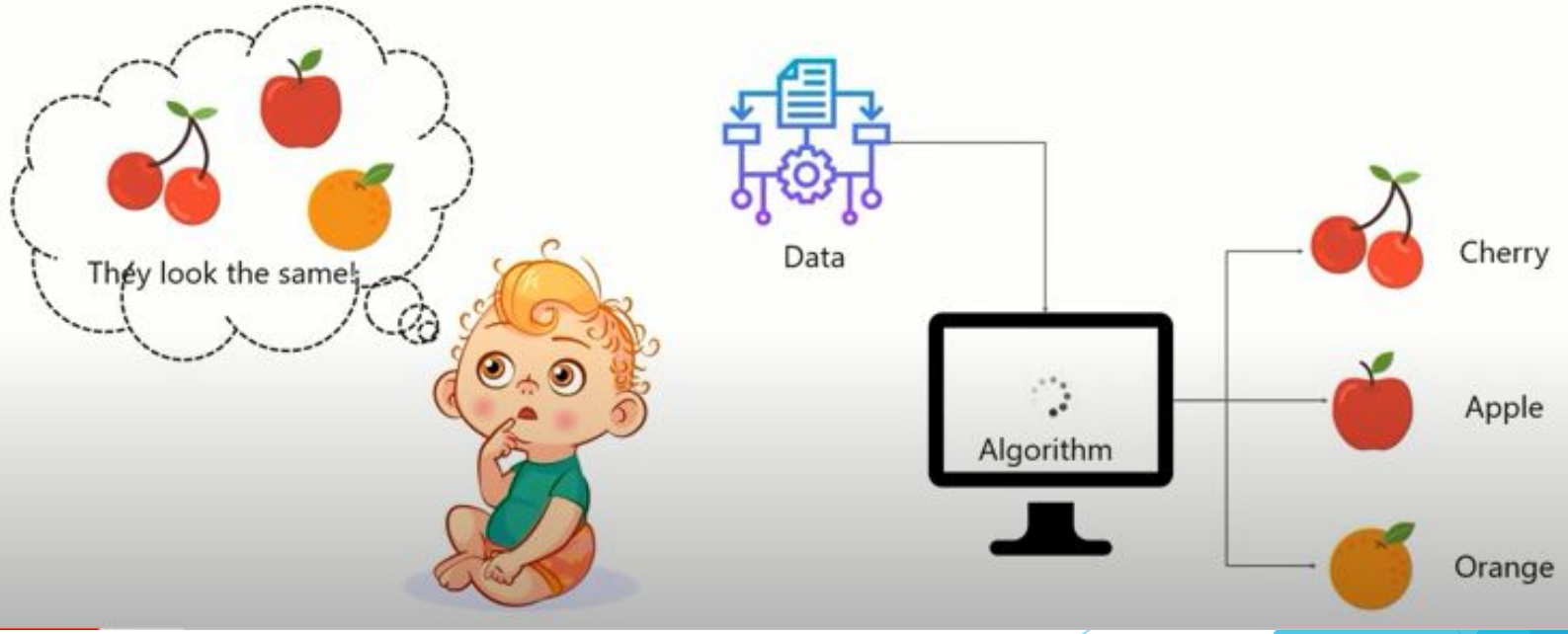
Highlights

- Definition of Machine Learning
- Workflow of ML
- Types of ML
  - Supervised
  - Unsupervised
  - Reinforcement
- Short definition of the types of learning
  - Working Process
  - Examples
  - Training
  - Approach
- Supervised Learning
  - Types
    - Regression
    - Classification
  - Applications

Sources: Google, Edureka, Simplilearn, Great learning, Codebasics, 3Blue1Brown, DeepLearningAI, CodeWithHarry, Krish Naik, CampusX etc

What Is Machine Learning?

Machine learning is a subset of artificial intelligence (AI) which provides machines the ability to learn automatically & improve from experience without being explicitly programmed.



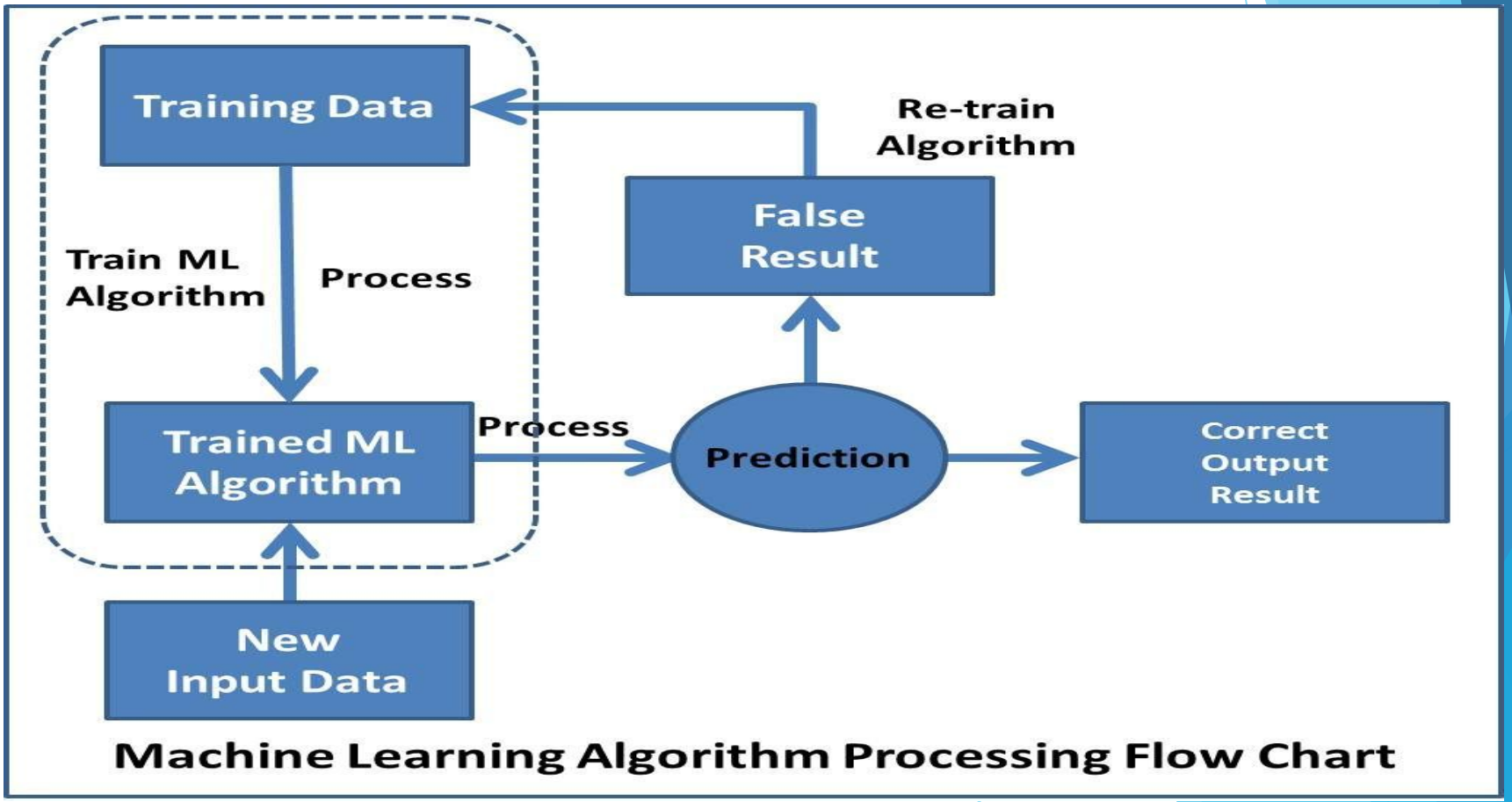
Contd..

- A Process that says how a machine learns.
- A branch of artificial intelligence (AI) and computer science
  - Focuses on the use of data and algorithms
  - Imitate the way that humans learn, gradually improving its accuracy.
- Machines learns automatically without being explicitly programmed or learning
  - Without any direct human intervention
- Algorithms are trained to find patterns and correlations in data sets
  - To make the best decisions and predictions based on that analysis.
- Designed to classify things, find patterns, predict outcomes, and make informed decisions

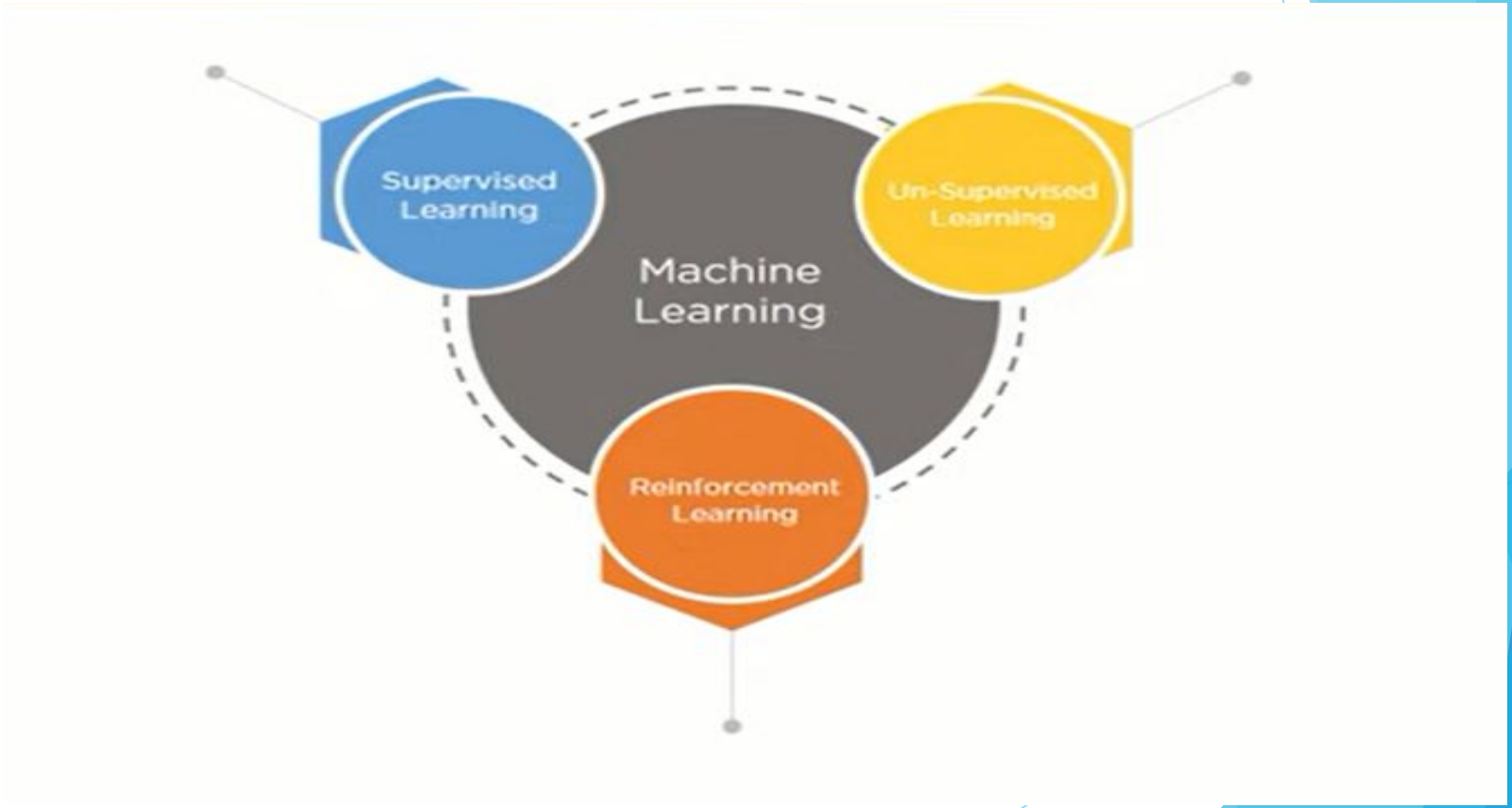
Process

- ✔ Feed good quality data
- ✔ Train the machines by building various machine learning models using the data and different algorithms.
- ✔ The choice of algorithms depends on what type of data we have
- ✔ what kind of task we are trying to automate.

Contd..


Training Data
Re-train Algorithm
False Result
Train ML Algorithm
Process
Trained ML Algorithm
Process
Prediction
Correct Output Result
New Input Data
Machine Learning Algorithm Processing Flow Chart

Types of Learning



Types of Learning

Supervised Learning

Unsupervised Learning

Reinforcement Learning

Definition of the Types

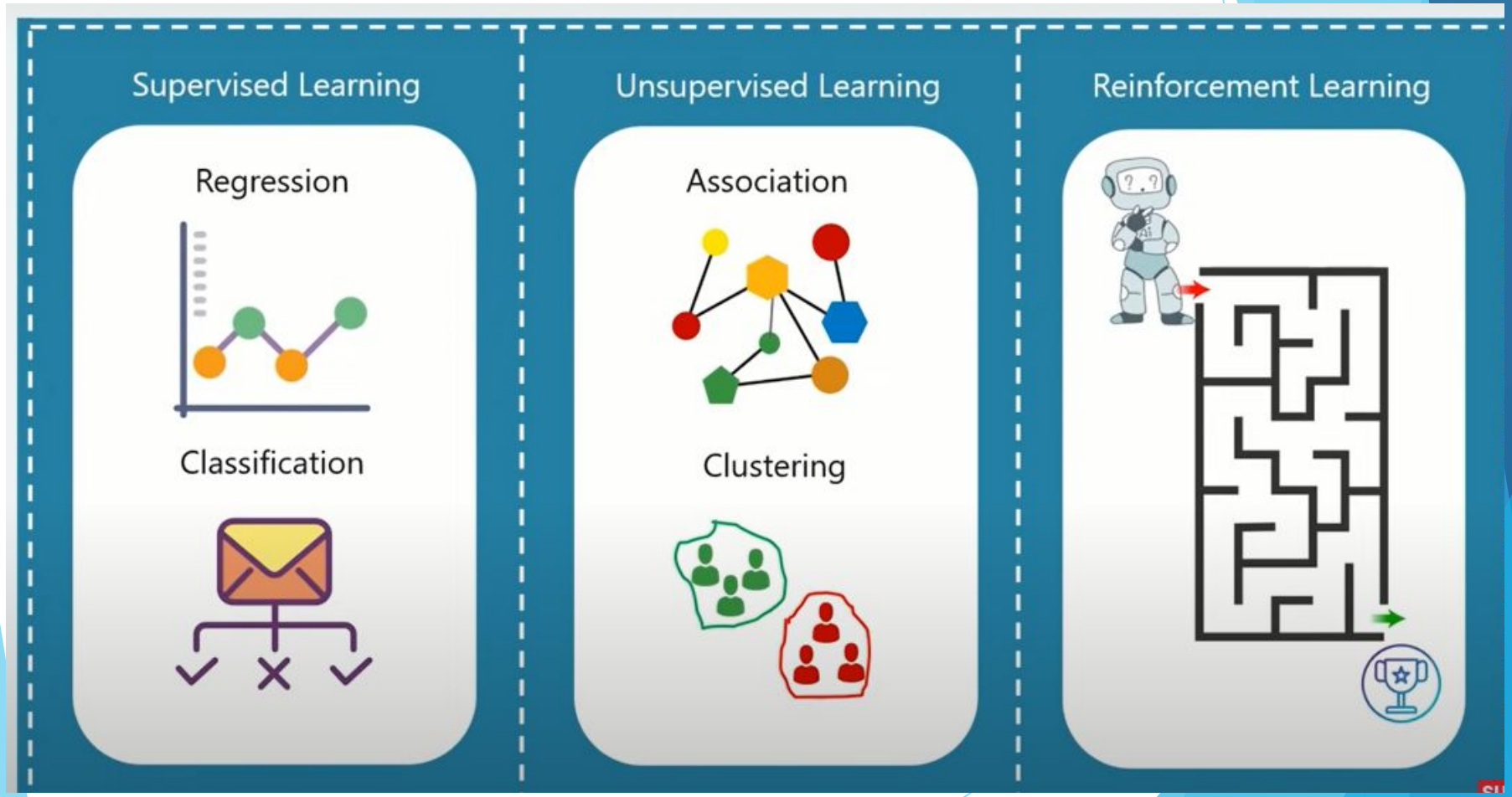
- Supervised : Train the Model with Labelled Data
  - To work accurately, labels are needed with the data
- Unsupervised : Train the model with unlabelled data
  - To make reliable, no human interaction is required
- Reinforcement : Learn by trial and error method
  - Take the cue from how human learns
  - Algorithm improves by its own

Supervised learning is a method in which we teach the machine using labelled data

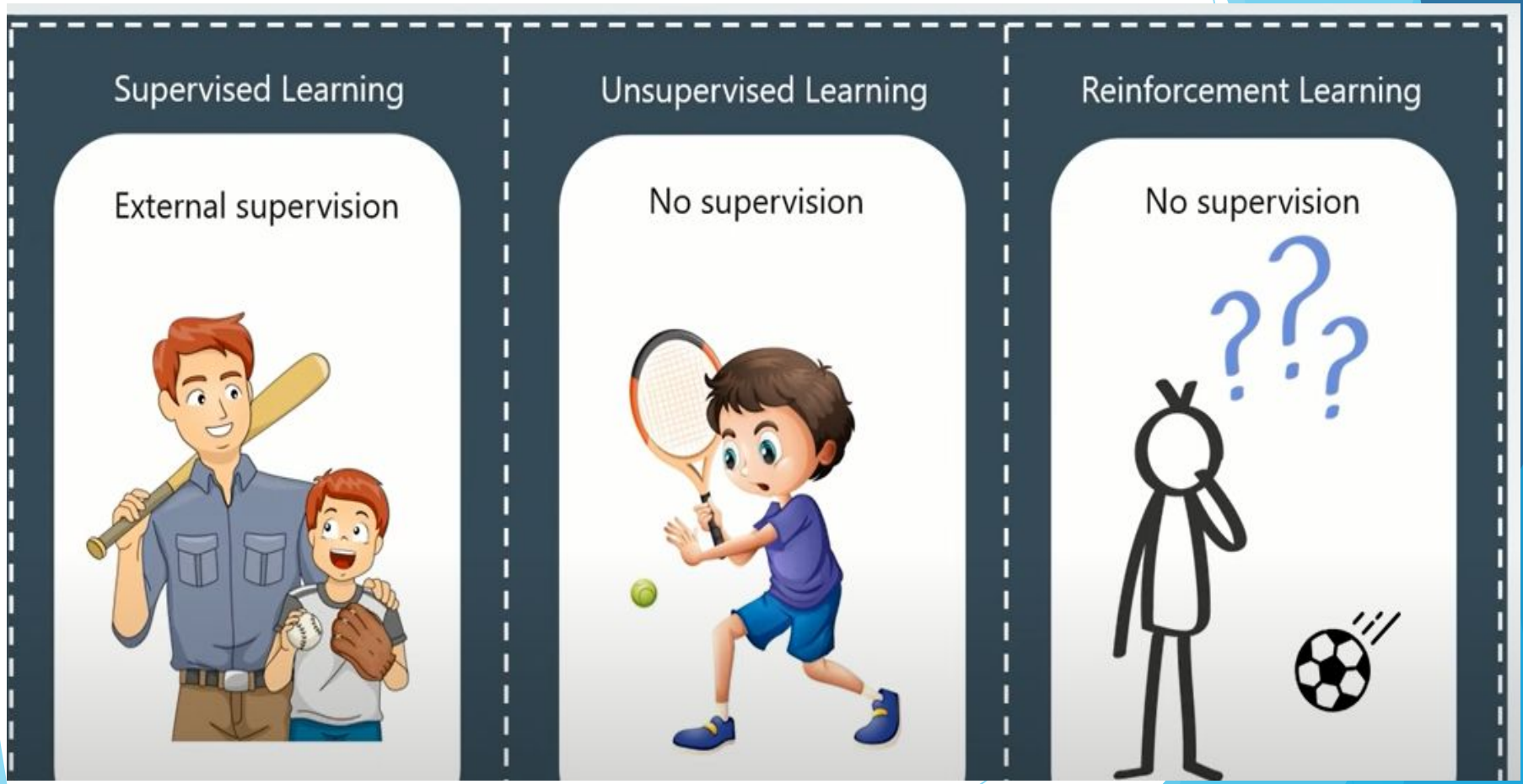
In unsupervised learning the machine is trained on unlabelled data without any guidance

In Reinforcement learning an agent interacts with its environment by producing actions & discovers errors or rewards

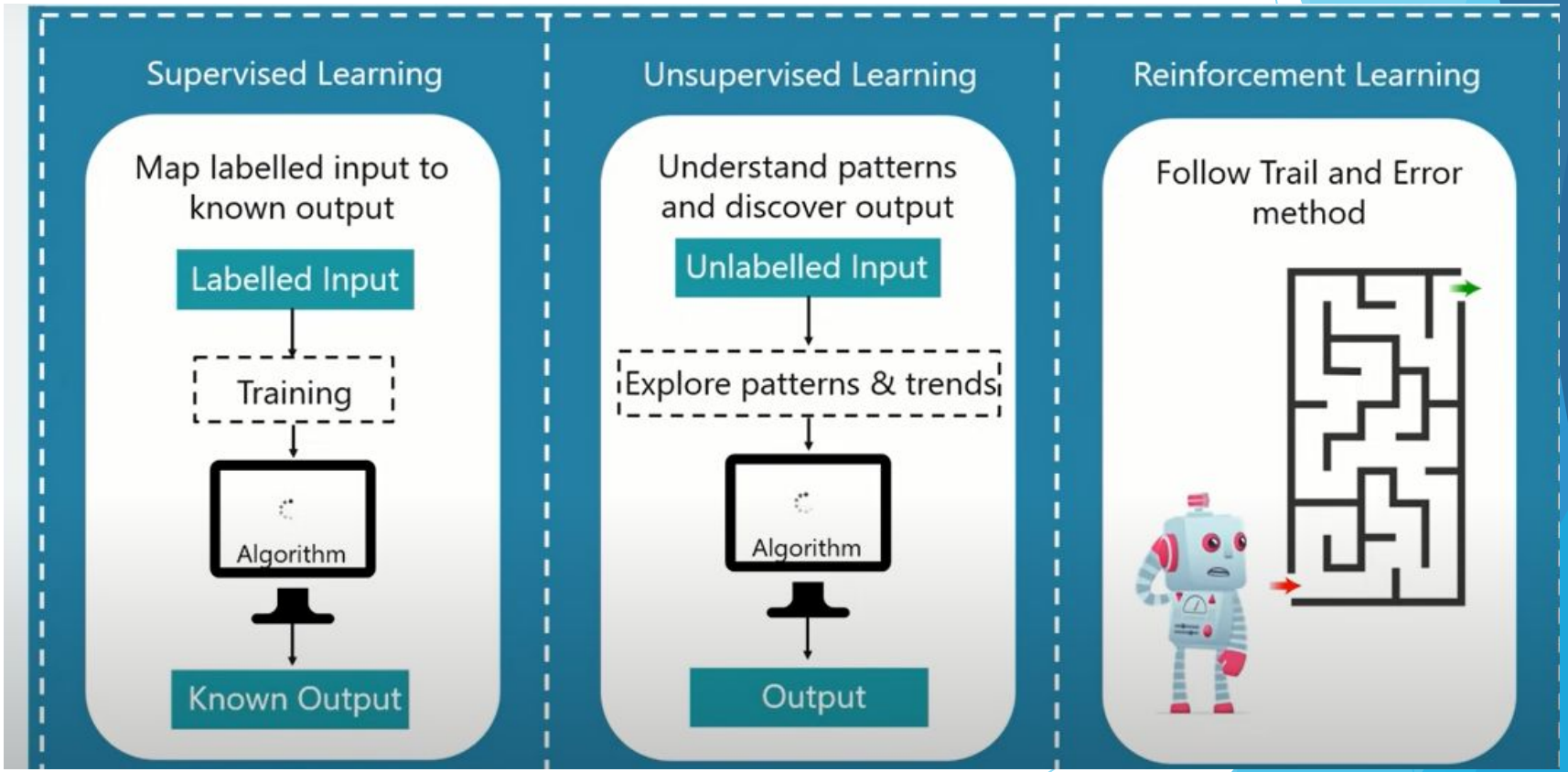
Examples


Supervised Learning
Regression
Classification
Unsupervised Learning
Association
Clustering
Reinforcement Learning

Training


Supervised Learning
External supervision
Unsupervised Learning
No supervision
Reinforcement Learning
No supervision

Approach


Supervised Learning
Map labelled input to known output
Labelled Input
Training
Algorithm
Known Output
Unsupervised Learning
Understand patterns and discover output
Unlabelled Input
Explore patterns & trends
Algorithm
Output
Reinforcement Learning
Follow Trail and Error method

Supervised Learning

Supervise the model how to work...

- Machines are trained using well "labelled" training data,
  - The labelled data means some input data is already tagged with the correct output.
    - Based data, machines predict the output.

- Feed the model with inputs along with its corresponding labels for training

- Then tests the model using unknown data to get the corresponding labels

- Aim : Find **a mapping function to map the input variable(x) with the output variable(y)**.

- Applications:
  - **Risk Assessment, Image classification, Fraud Detection, spam filtering**, etc.

Types:

Regression

- A method to determine the statistical relationship between a dependent variable and one or more independent variables.
- The change independent variable is directly associated with the change in the independent variables.
- Types::.
  1. Linear Regression
  2. Logistic Regression

- Applications:
  1. Stock prediction, House Price Prediction, Sales etc

How linear regression works: Sales Approach

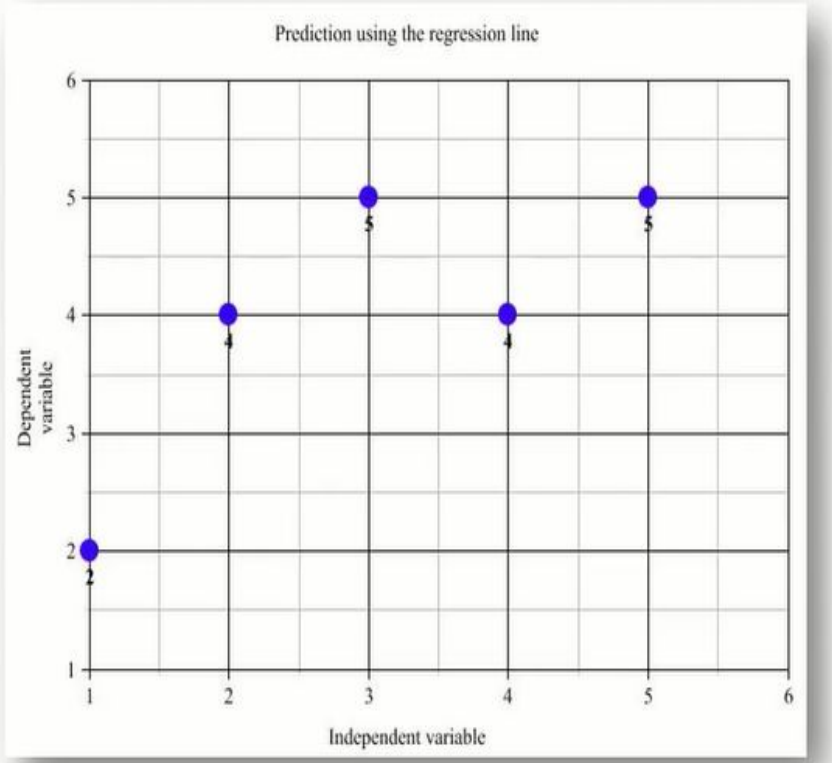
Develop a linear equation:

$$y = w1x + w0$$

- w1 and w0 are the weights
- W1 : Regression coefficient
- W0: Regression constant

Maths Behind Linear Regression

Independent variable	Dependent variable
X	Y
1	2
2	4
3	5
4	4
5	5



Plotting the data points

Contd..

X	Y
1	2
2	4
3	5
4	4
5	5

Consider point (3,5)

Y=w1x+w0

Assume,

w1=0.6

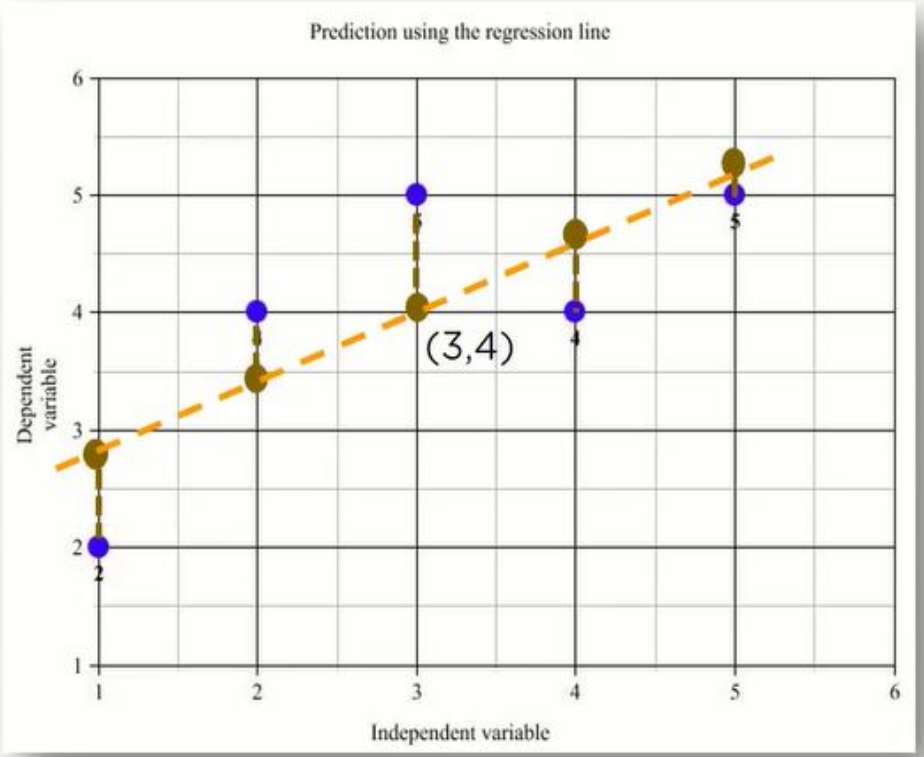
w0=2.2

x=3

Calculate Y = 0.6*3+2.2

= 4

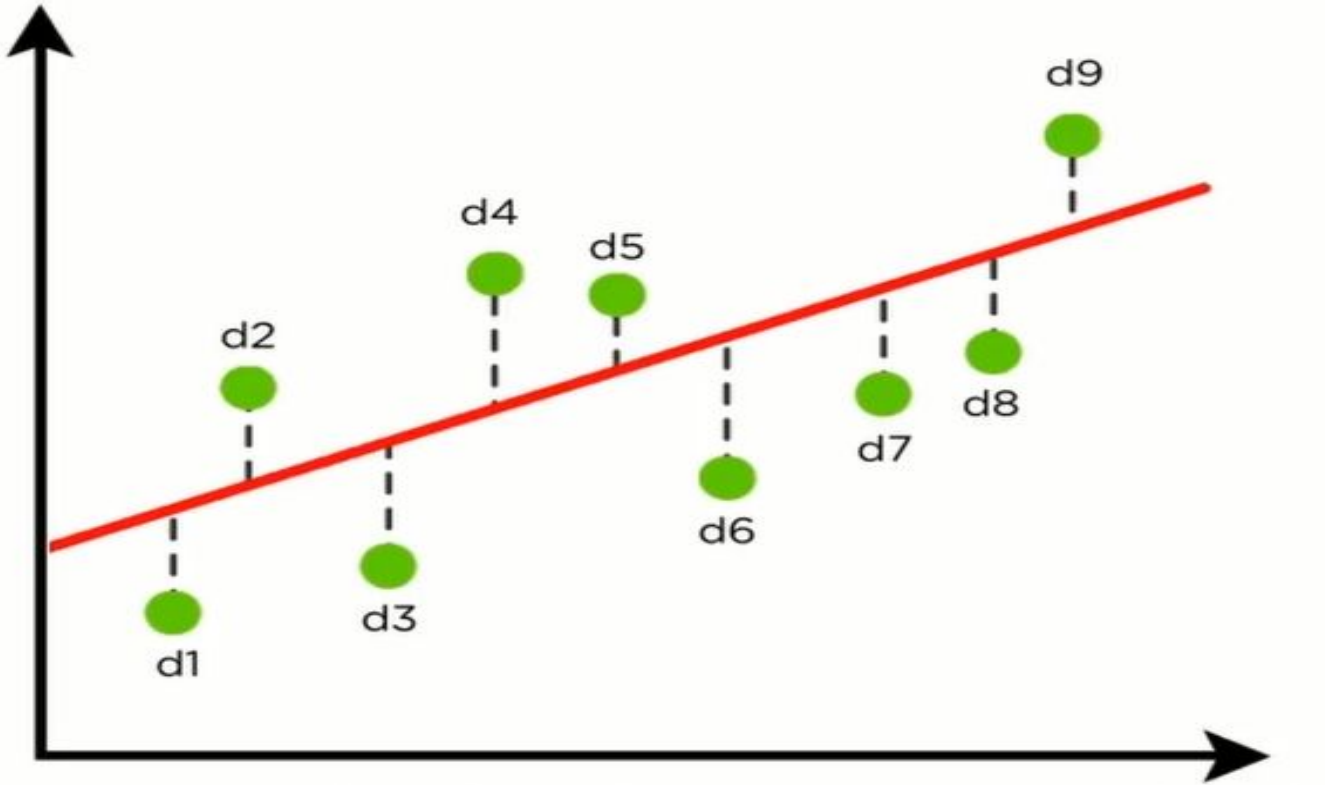
New point, (3,4)



X	Y	Y_{pred}
1	2	2.8
2	4	3.4
3	5	4
4	4	4.6
5	5	5.2

Plotting the new predicted points

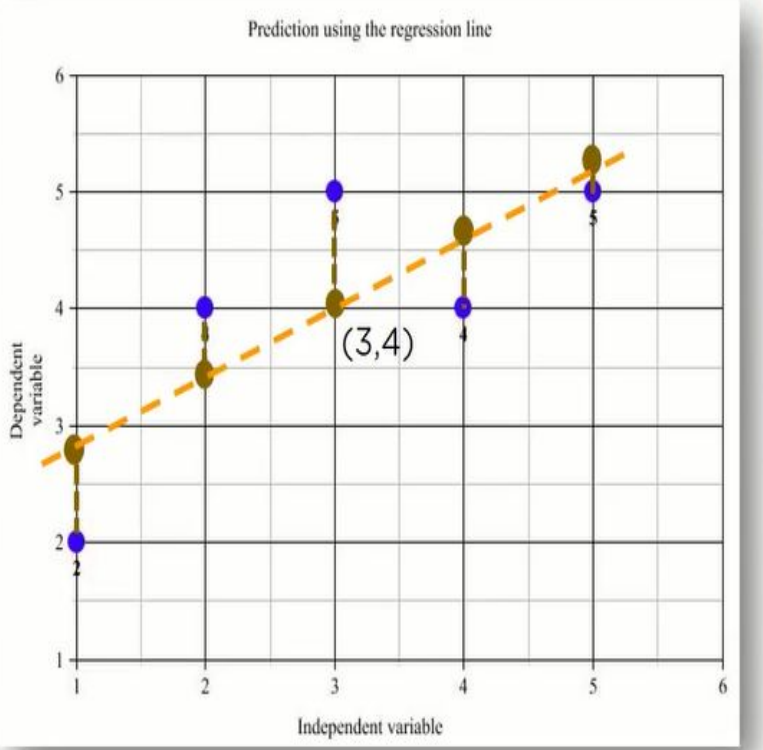
Calculate the Loss or Error



$$D = d1^2 + d2^2 + d3^2 + d4^2 + d5^2 + d6^2 + d7^2 + d8^2 + d9^2$$

Loss value (Error) = D

Target: **Reduce the loss**
Find correct pattern



X	Y	Y_{pred}	$(Y-Y_{pred})$	$(Y-Y_{pred})^2$
1	2	2.8	-0.8	0.64
2	4	3.4	0.6	0.36
3	5	4	1	1
4	4	4.6	-0.6	0.36
5	5	5.2	-0.2	0.04

$\sum = 2.4$

The sum of squared errors for this regression line is 2.4. We check this error for each line and conclude the best fit line having the least e square value.

D=2.4

Classification

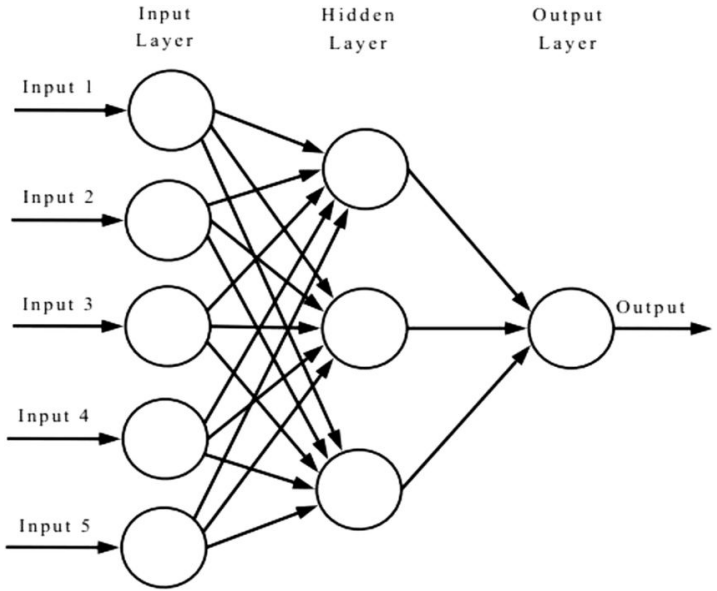
- Categorise the input data into different classes
- Learn the inputs and produce the corresponding classes as outputs.

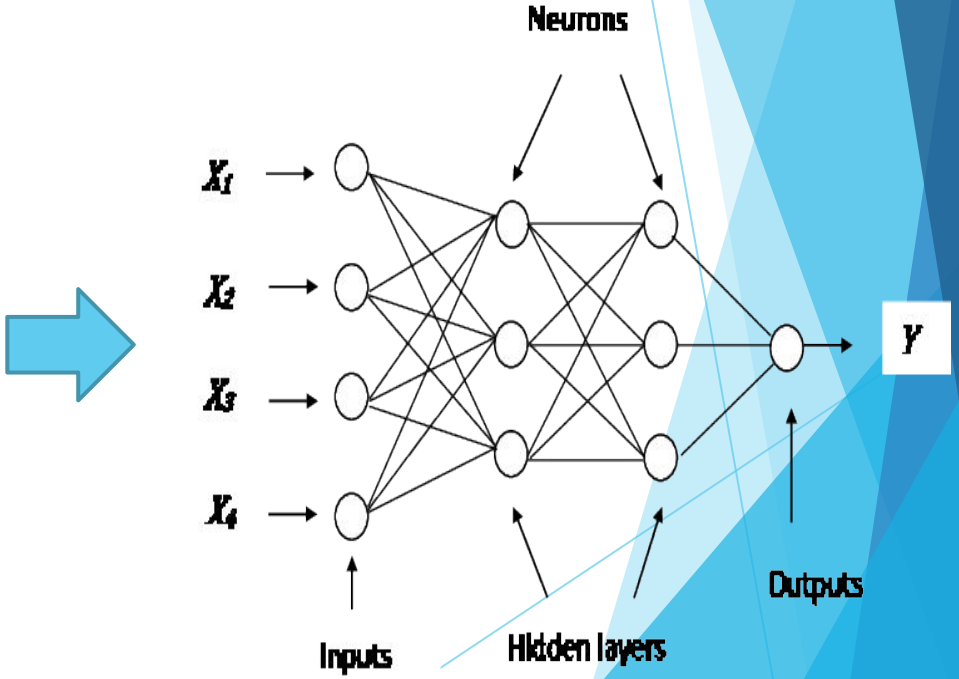
- Can use Logistic Regression, SVM, Decision Tree, Neural Network, etc
- Types of Neural network used:
  - Convolutonal neural network
  - Recurrent neural network
  - Deep neural network

Deep neural network

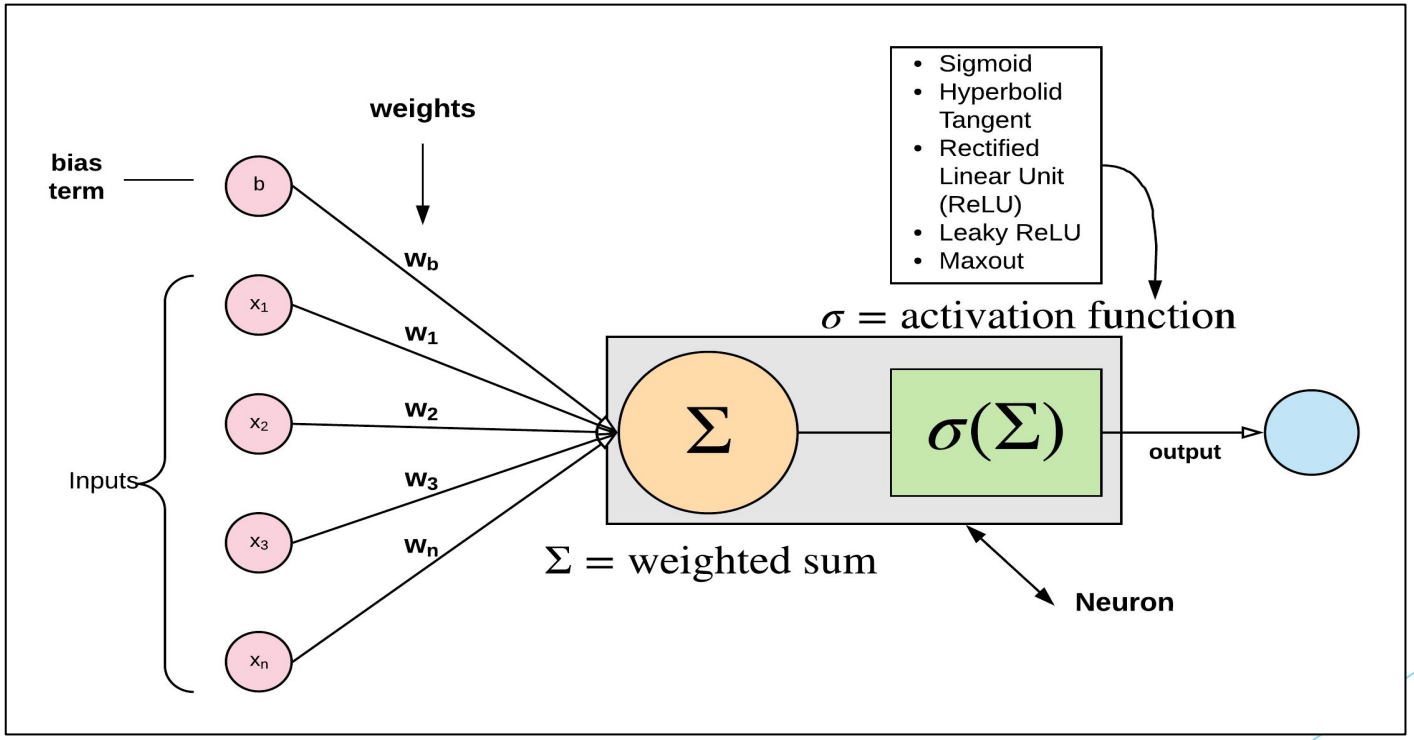
Multiple layers in neural network

More than one hidden layers





Feedforward Neural Network



Equation: y= f(x1w1+x2w2+x3w3+x4w4+…+xnwn+b)

Maths behind Feedforward Neural Network

- *Consider 2 inputs, x1 and x2::*
- *y= f(w1x1+w2x2+b)*

- W1=0.2
- W2=0.5
- X1=2
- X2=4
- b=0

- Weighted sum=0.4+2.0=2.4
- Applying sigmoid activation function (for single class classification):
  - Predicted output (y)=0.7 (assume)
  - Check,
    - If y<0.5 then Class 0
    - Else Class 1
    - So here,
    - Output= class 1

Binary Classification

- **Inputs : Features**

- **Output (Classes) : Class variable (0 or 1)**
  - **Single Neuron**
  - **Use Sigmoid activation normally**

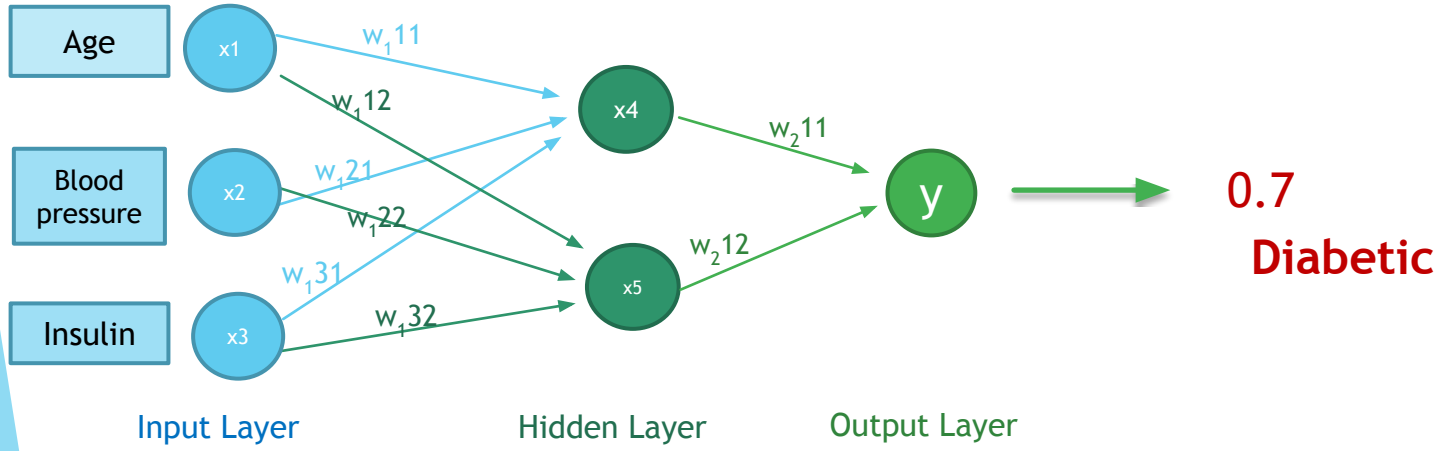
Binary Classification

- **Inputs (Features):**

- Diastolic blood pressure (mm Hg)
- 2-Hour serum insulin (mu U/ml)
- Age (years)

- **Output (Classes)**
  - Patient is having Diabetes or not
  - Output : 0 or 1

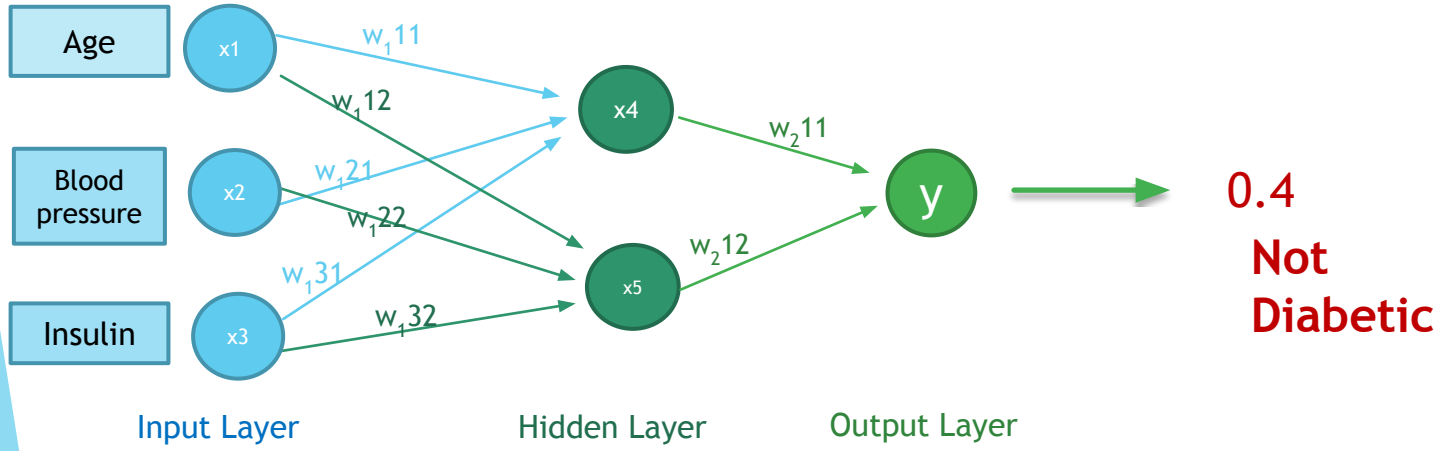
Diabetic Detection



0.7

Diabetic

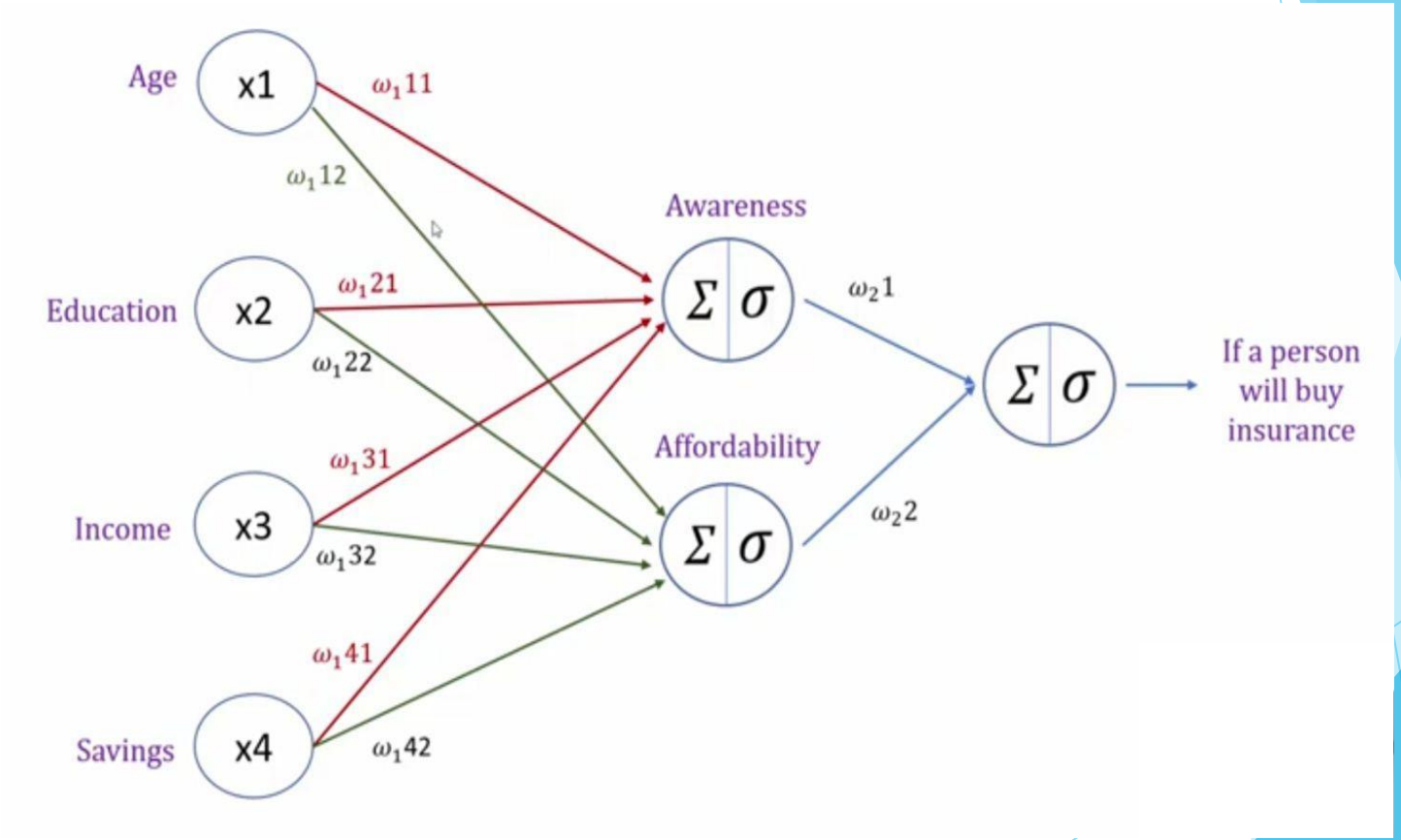
Diabetic Detection



0.4

Not Diabetic

Insurance will be granted or Not

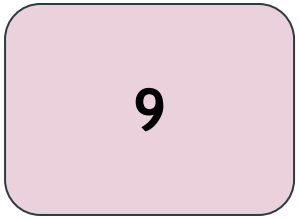


Multiclass Classification (Digit recognition)

- Input: Features

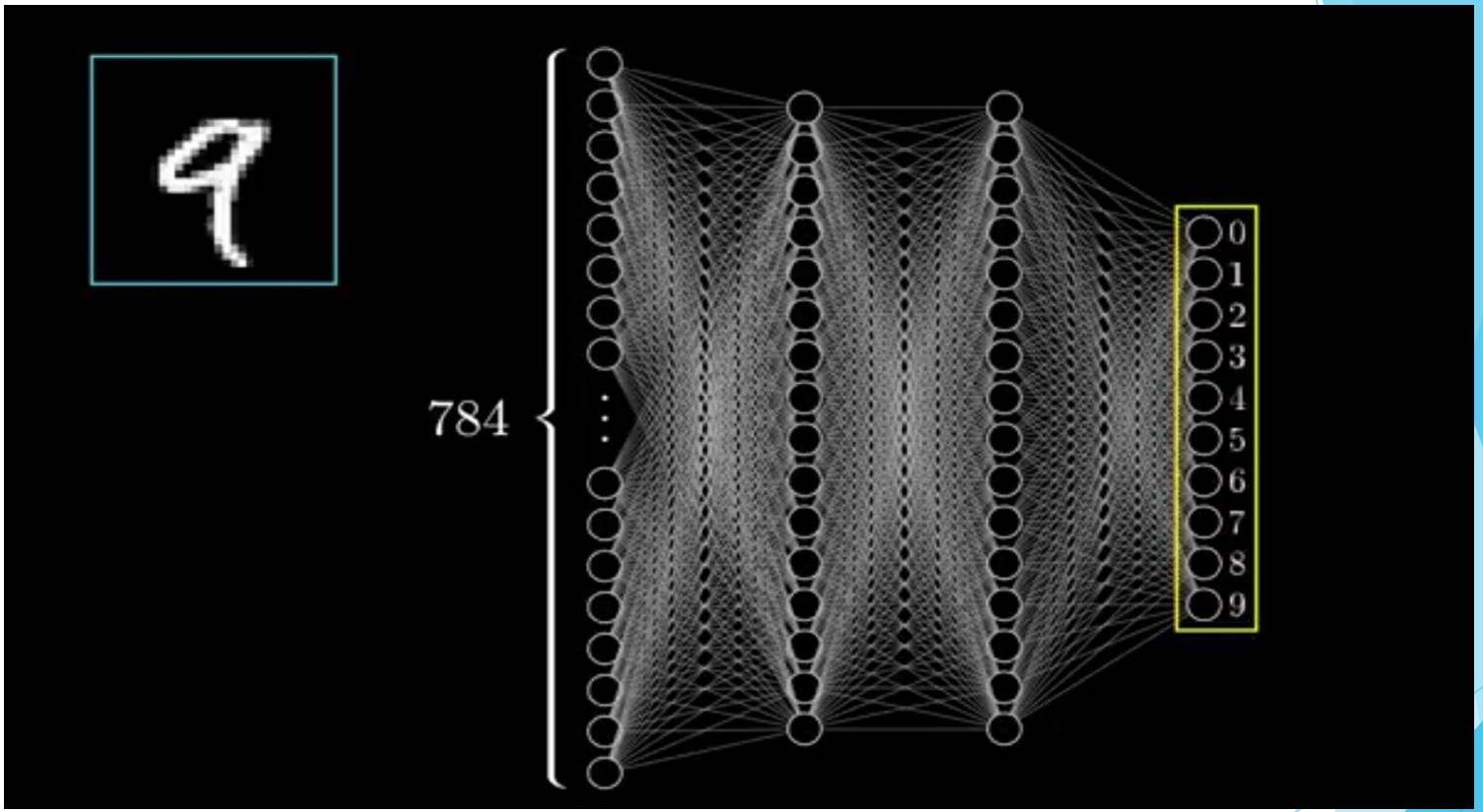
- Output: Corresponding classes
  - Output nodes : Number of classes
  - Use Softmax activation

Digit recognition

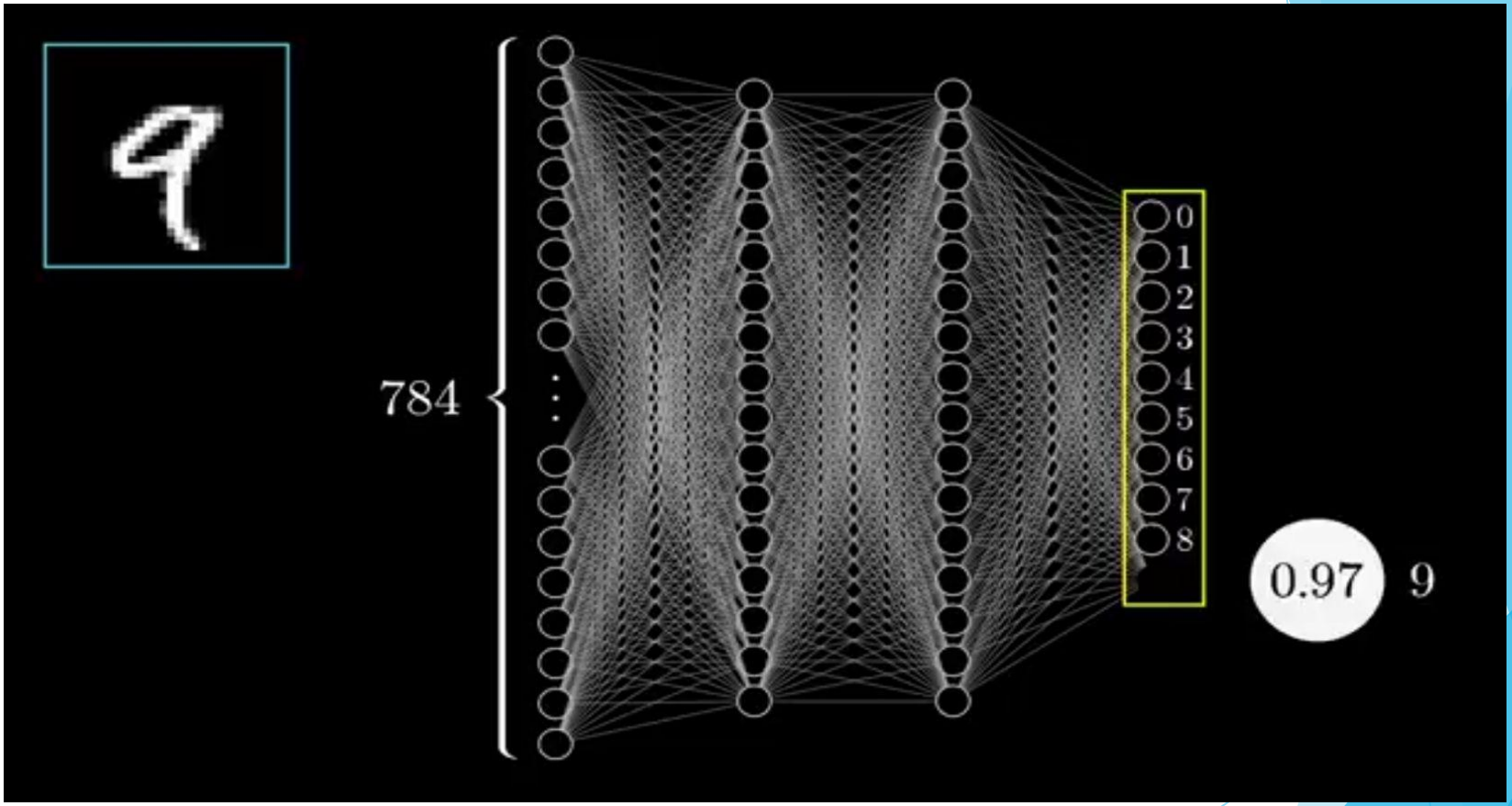


Inputs: 0 to 9 Handwritten Digits in image format

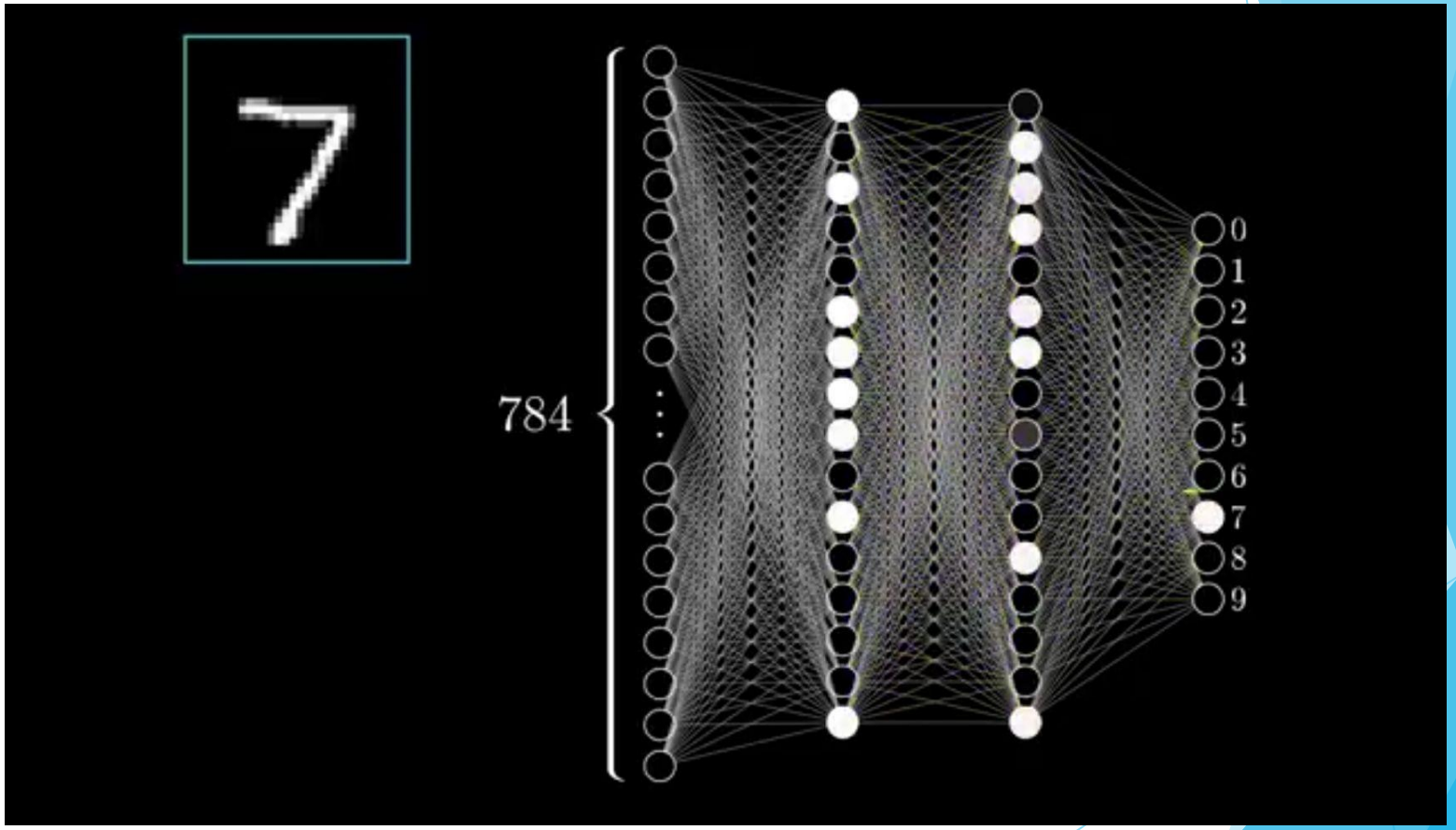
Outputs: Corresponding Digits



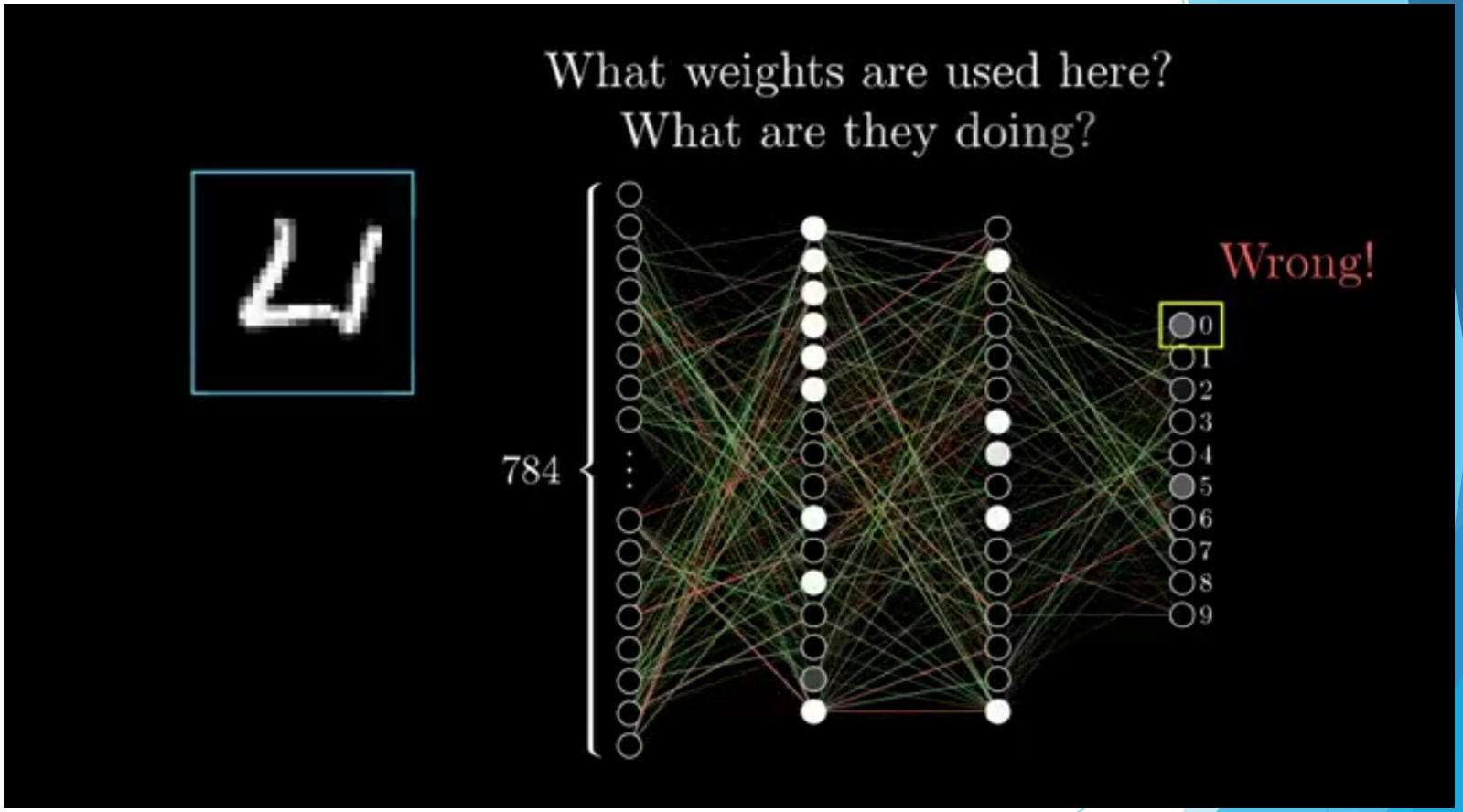
- Last Layer predicts Number 9 by predicting high score in 10th node
  - 10th Node represents 9 Digit



- Last Layer predicts Number 9 by predicting high score in 10th node
  - 10th Node represents 9 Digit
  - 10th node holds value 0.97, which is heights in thiscase



Predicts 7



- Trying to predict 4, But predicted 0, so wrong output
- Update weights to reduce loss

Solution?

- Calculate loss using formula:
  - Mean squared error (MSE) = $\sum_{k=0}^{n} (\text{y\_true}_k - \text{y\_pred}_k)^2$ / (**Number of outputs** n)
  - Reduce loss by updating weights
  - To update weights, model needs to backtrack
    - Called backpropogation
  - Continue this process until loss will be minimum
  - Iterate a number of times (epochs)

Applications..

- Recognition Tasks:
  - Face Recognition
  - Speech Recognition
  - Hand writing, text recognition etc
- Self Driving Cars
  - Automated cars
- Email Filtering, Malware detection
  - Spam detection
- Translation
  - Language Translation
- Medical Diagnosis
  - Diabetic/cancer detection

Few Sources of Practice Code

- https://www.geeksforgeeks.org/python-programming-language/
- https://towardsdatascience.com/solving-a-simple-classification-problem-with-python-fruits-lovers-edition-d20ab6b071d2
- https://training.atmosera.com/binary-classification-with-neural-networks/
- https://intellipaat.com/blog/confusion-matrix-python/#:~:text=Confusion%20matrix%20in%20python%20helps%20us%20describe%20the,the%20related%20terminologies%20can%20be%20a%20bit%20confusing.
- https://www.kaggle.com/code/sivakumarai/insurance-premium-prediction-linear-regression
- https://stackoverflow.com/questions/67650412/how-to-find-the-predicted-output-of-a-classification-neural-network-in-python
- https://towardsdatascience.com/the-complete-guide-to-neural-networks-multinomial-classification-4fe88bde7839
- https://medium.com/luca-chuangs-bapm-notes/build-a-neural-network-in-python-multi-class-classification-e940f74bd899
- https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/
- https://builtin.com/data-science/data-clustering-python
- https://towardsdatascience.com/spam-detection-with-logistic-regression-23e3709e522
- https://www.kaggle.com/code/balaka18/email-spam-classification/notebook
- https://www.atmosera.com/blog/multiclass-classification-with-neural-networks/
- https://kshitizregmi.github.io/posts/2021/08/computer-vision/
- https://www.kaggle.com/code/karthik7395/binary-classification-using-neural-networks/notebook

Thanking you...