ARTIFICIAL INTELIGENCE

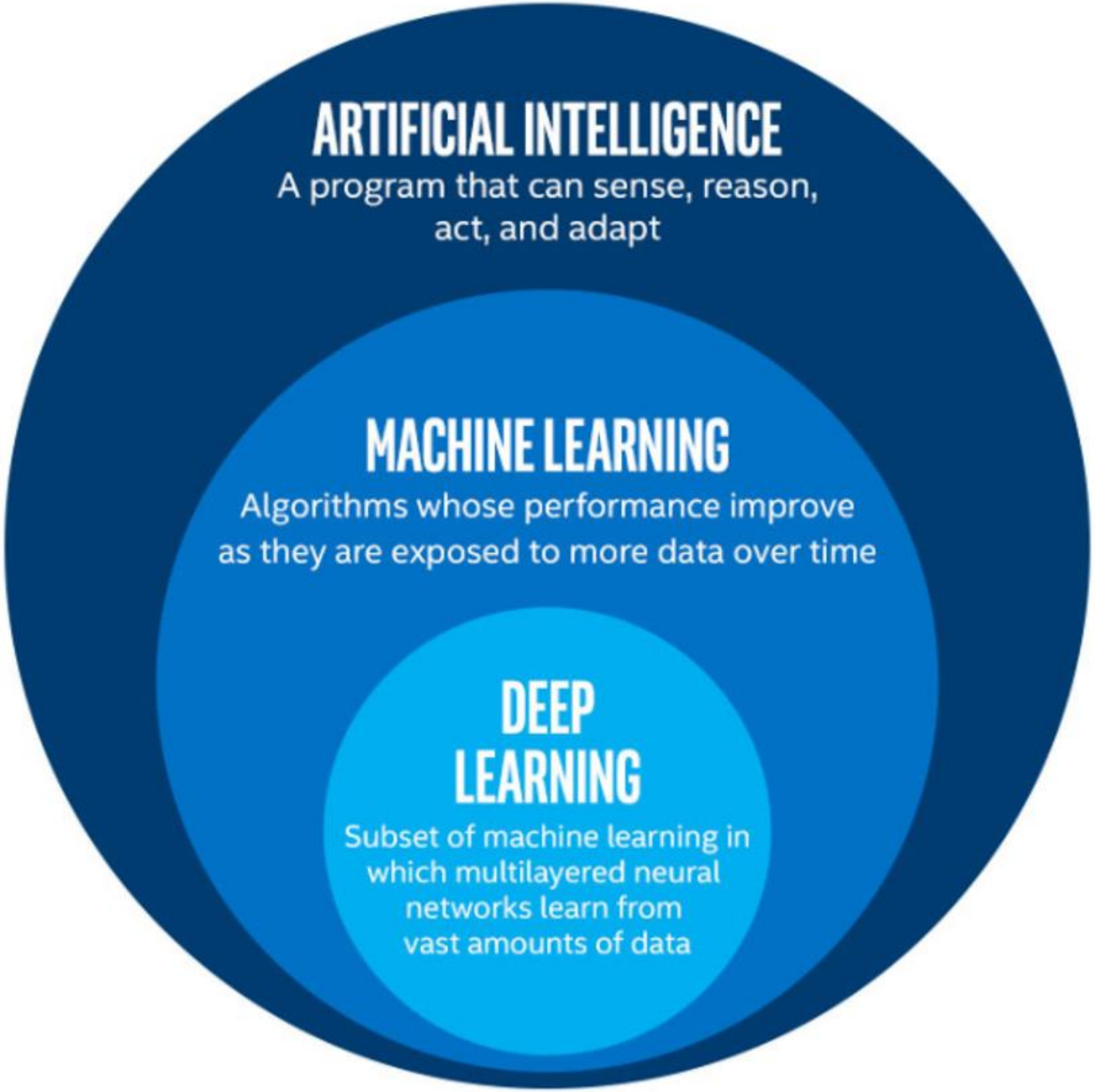
DEEP LEARNING & NEURAL NETWORKS

FEED FORWARD NEURAL NETWORKS 1

SPEAKER

Thakshila Dasun

BSc. Engineering Hons, MPhil (Reading)



AI

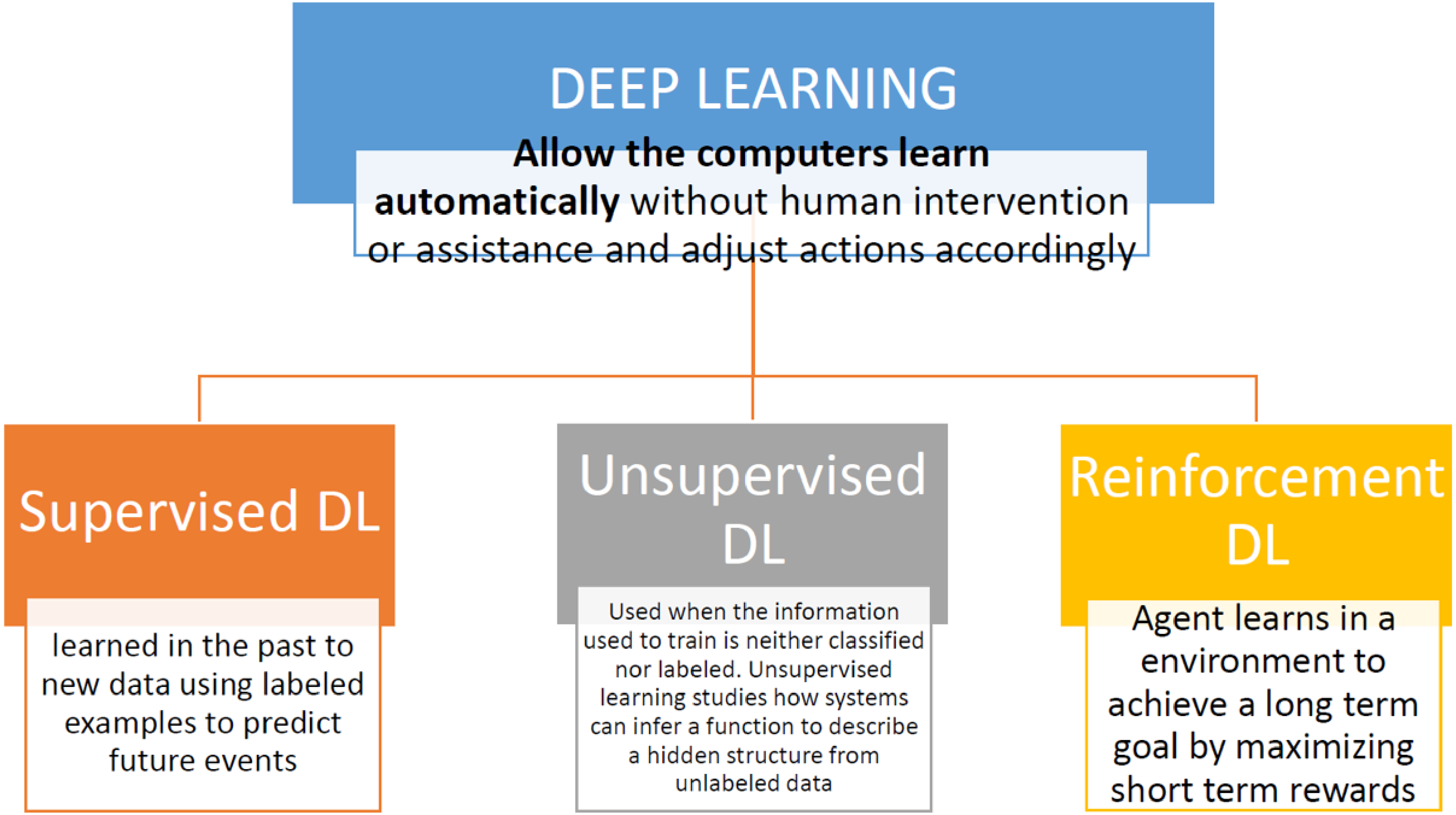
Simulation of human intelligence in machines that are programmed to think like humans and mimic their actions

ML

Computer Algorithms which can be trained for specific applications and used for future predictions

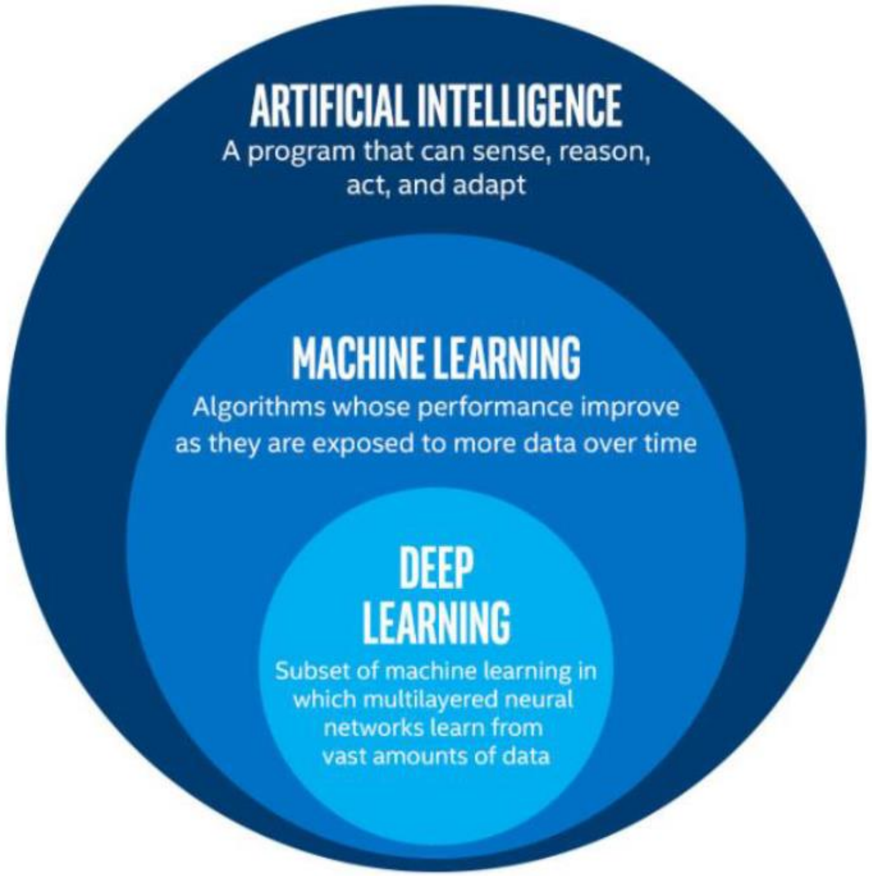
DL

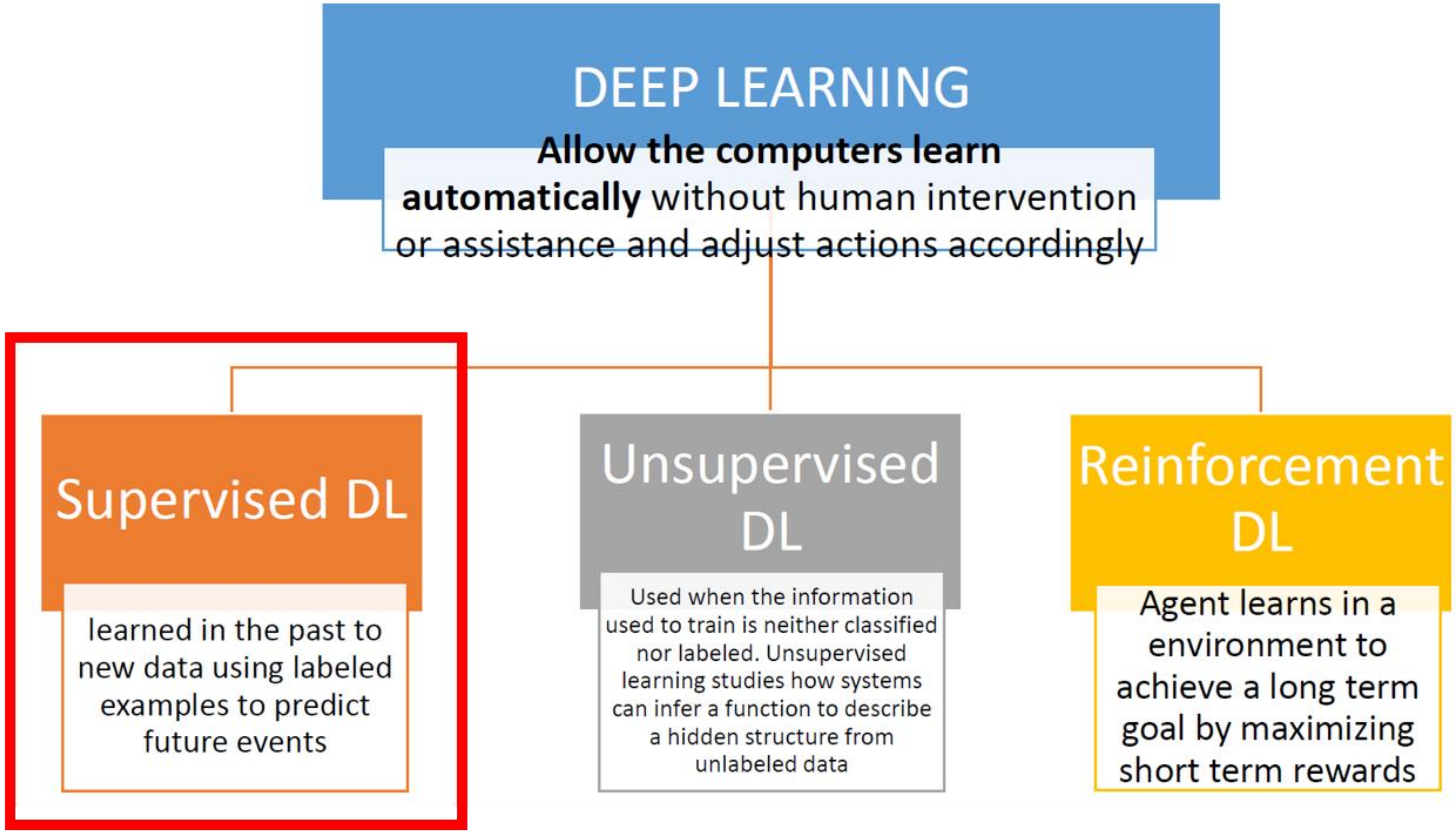
High level version of Machine Learning which uses Artificial Neural Networks as trainable Algorithms

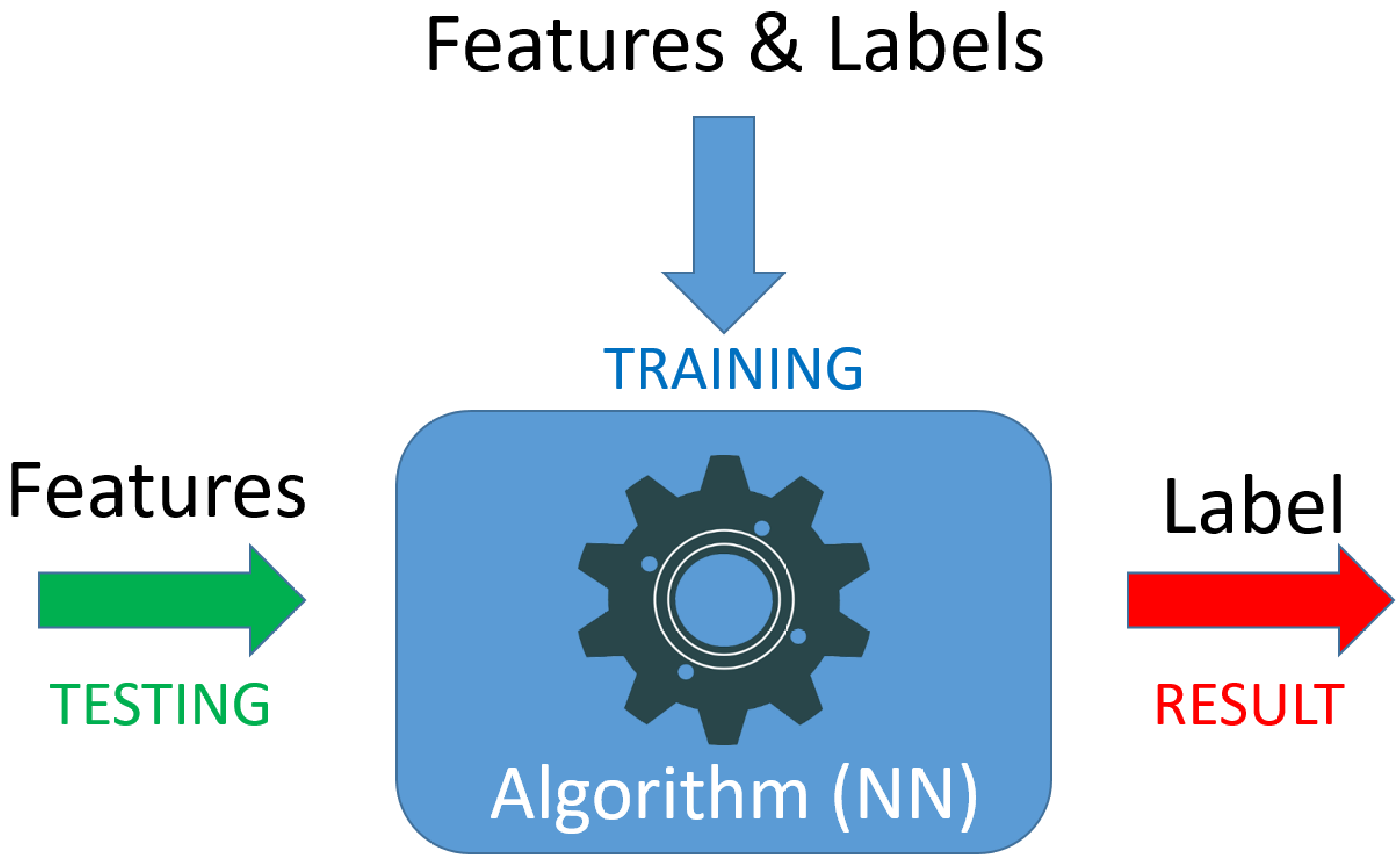


examples:

1. Supervised: Learn to swim with the guidance of an instructor
2. Unsupervised: Learn to swim by self studying
3. Reinforcement: Dropping off in the middle of the sea to learn swimming


ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

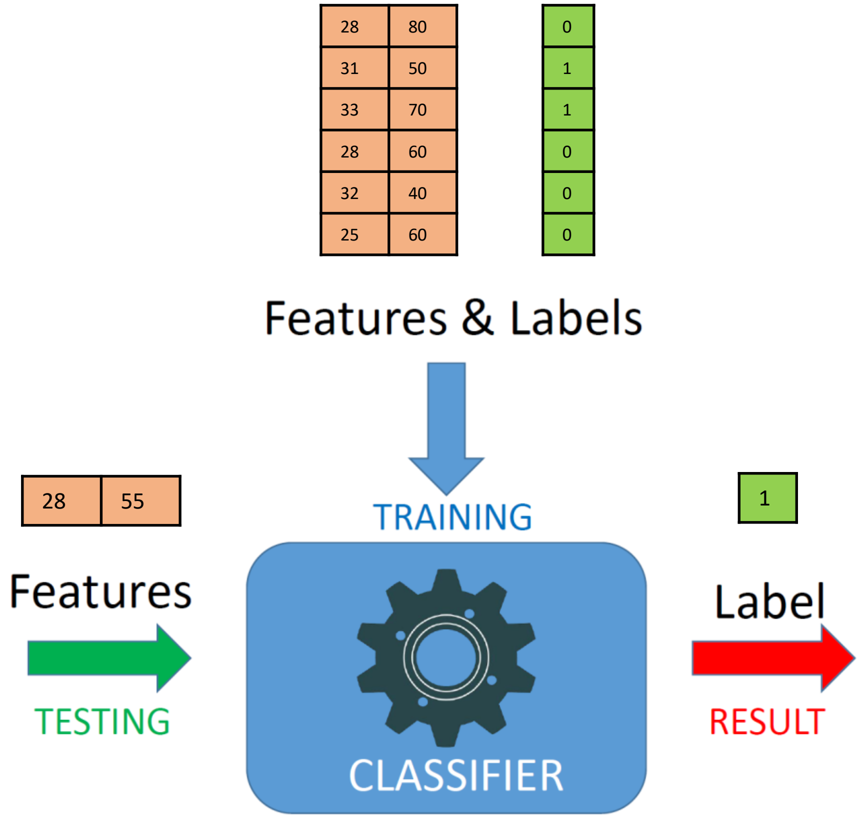

DEEP LEARNING
Allow the computers learn automatically without human intervention or assistance and adjust actions accordingly
Supervised DL
learned in the past to new data using labeled examples to predict future events
Unsupervised DL
Used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data
Reinforcement DL
Agent learns in a environment to achieve a long term goal by maximizing short term rewards


Features & Labels
TRAINING
Features
TESTING
Algorithm (NN)
Label
RESULT

Predicting whether it is going to RAIN today or NOT

1. Identify Feature and Labels
   - Labels: Possible solution of the problem
   - Feature: Critical Attribute that the labels are depended on
2. Find a dataset

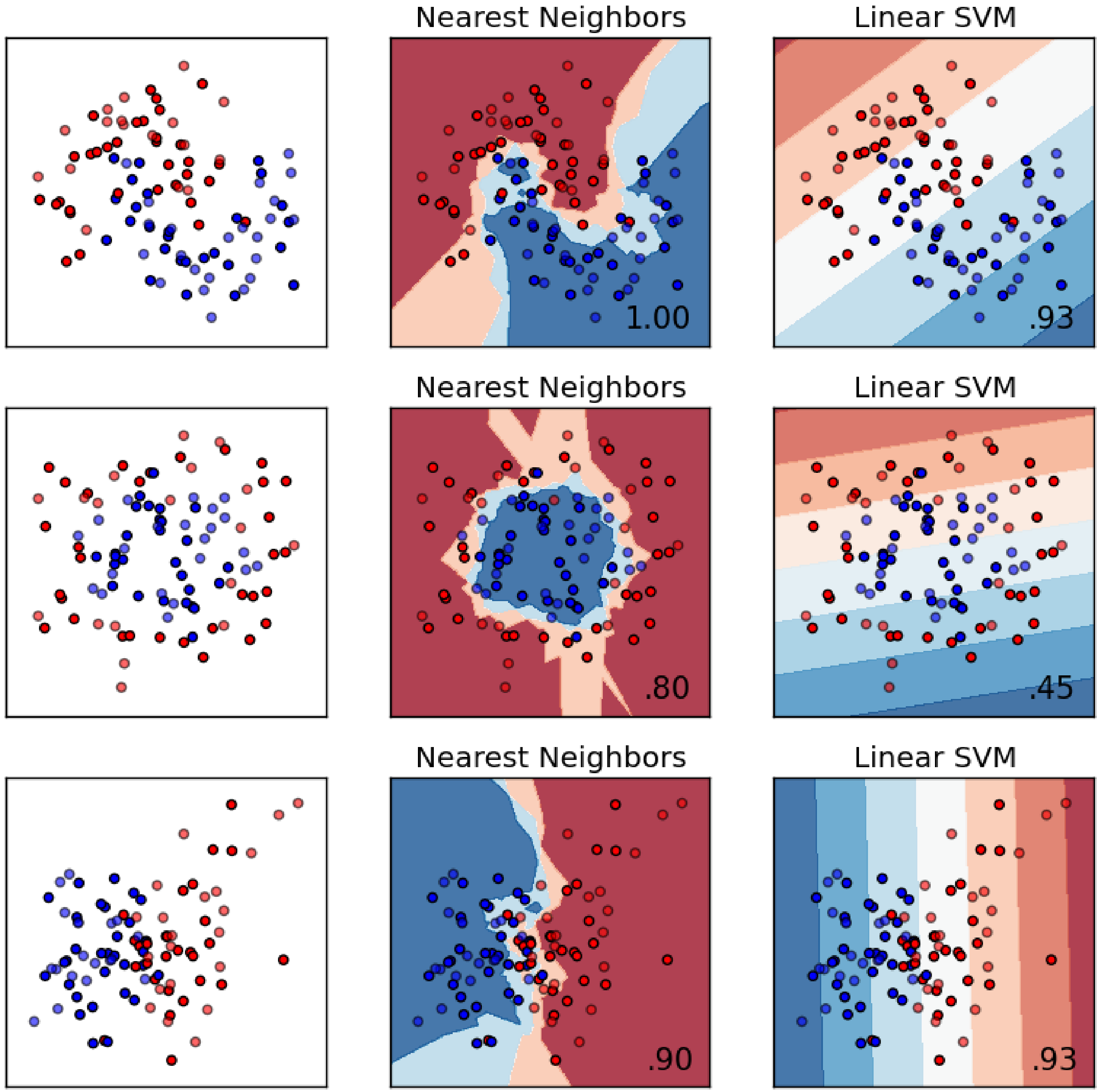
Features		Labels
Temperature (C)	Humidity %	
28	80	0
31	50	1
33	70	1
28	60	0
32	40	0
25	60	0



3. Train the Algorithm
4. Test & Results

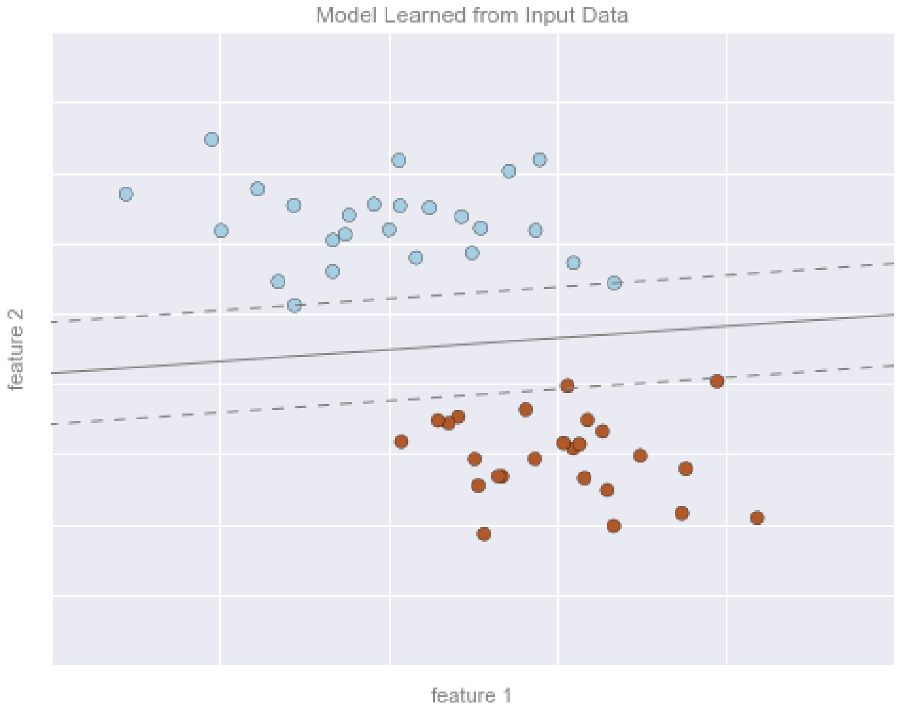
Classification & Regression

2 Types of Predictions in Machine Learning, Qualitative and Quantitative



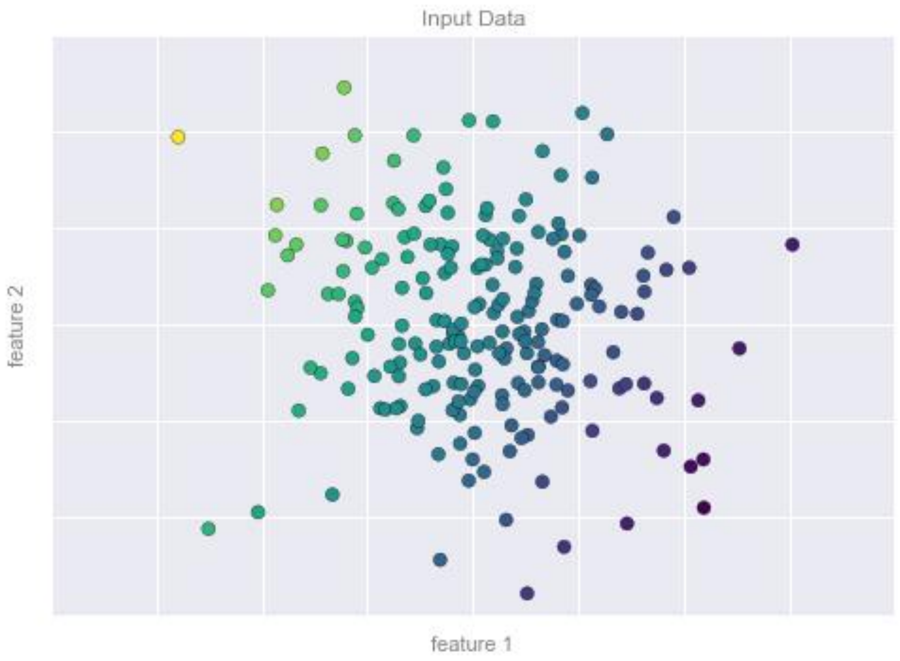
1. Classification: Predicting discrete labels

- “Classification" indicates that the data has discrete class label.
- Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y) or classes.
- The output variables are often called labels or categories. The mapping function predicts the class or category for a given observation

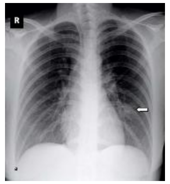


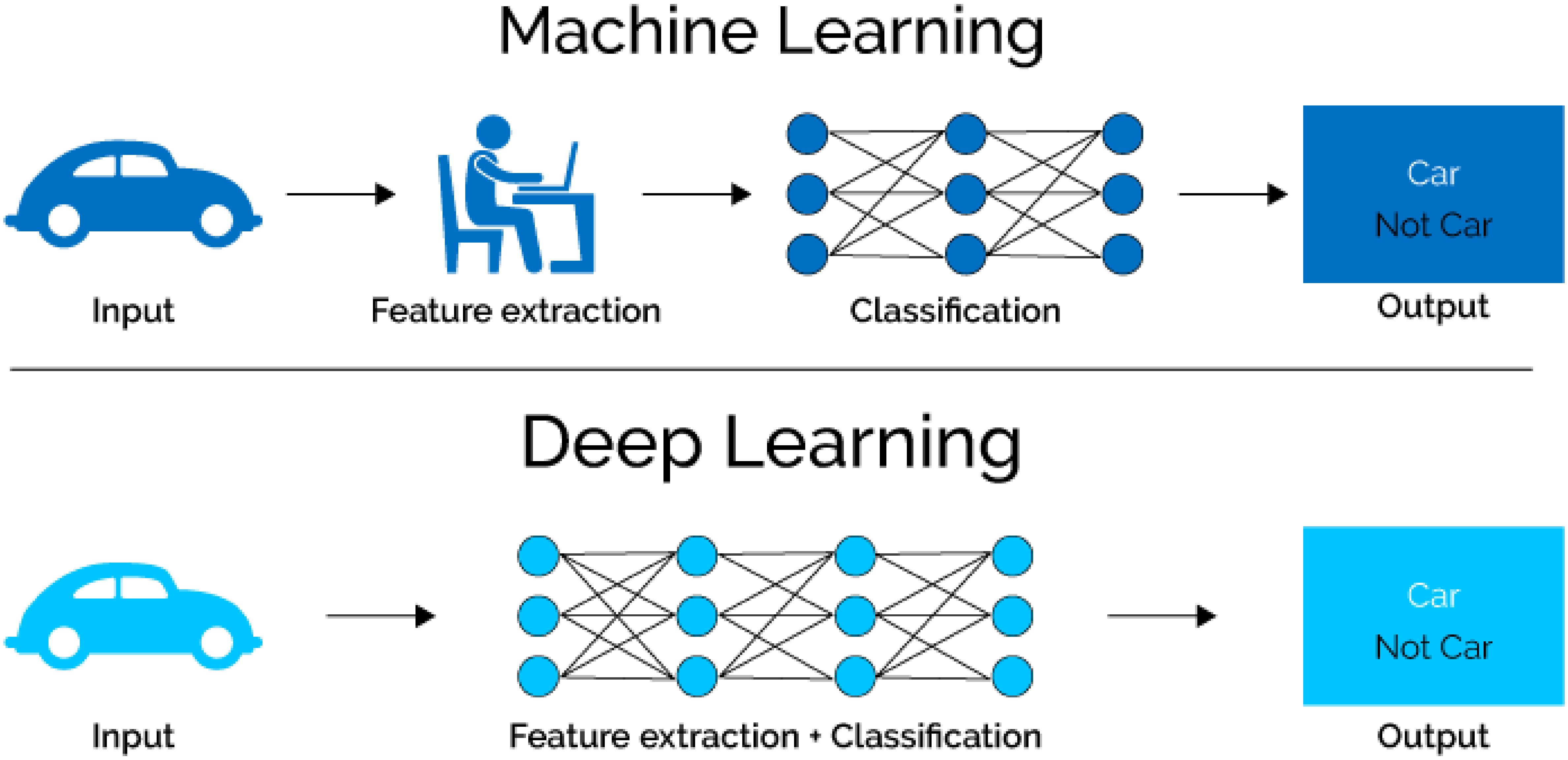
2. Regression: Predicting continuous labels

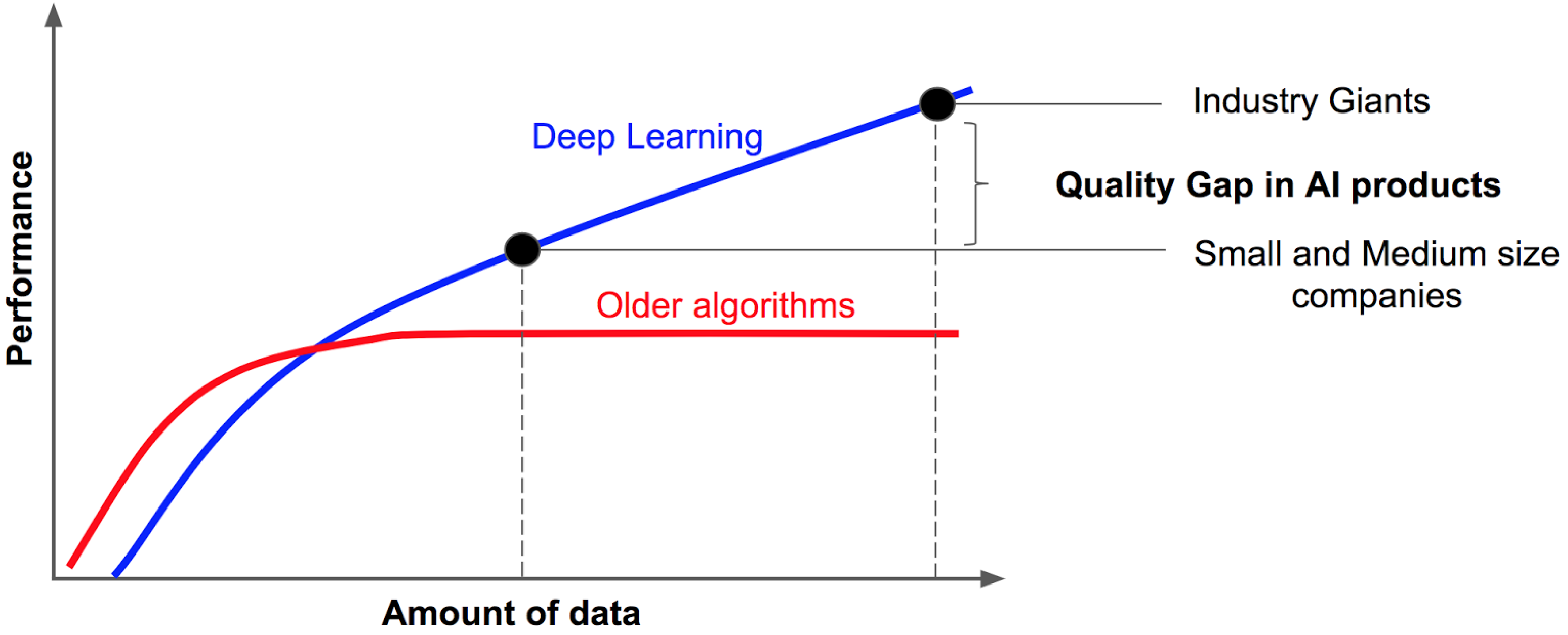
- Regression predictive modeling is the task of approximating a mapping function (f) from input variables (X) to a continuous output variable (y).
- A continuous output variable is a real-value, such as an integer or floating point value. These are often quantities, such as amounts and sizes.
- For example, a house may be predicted to sell for a specific dollar value, perhaps in the range of $100,000 to $200,000.



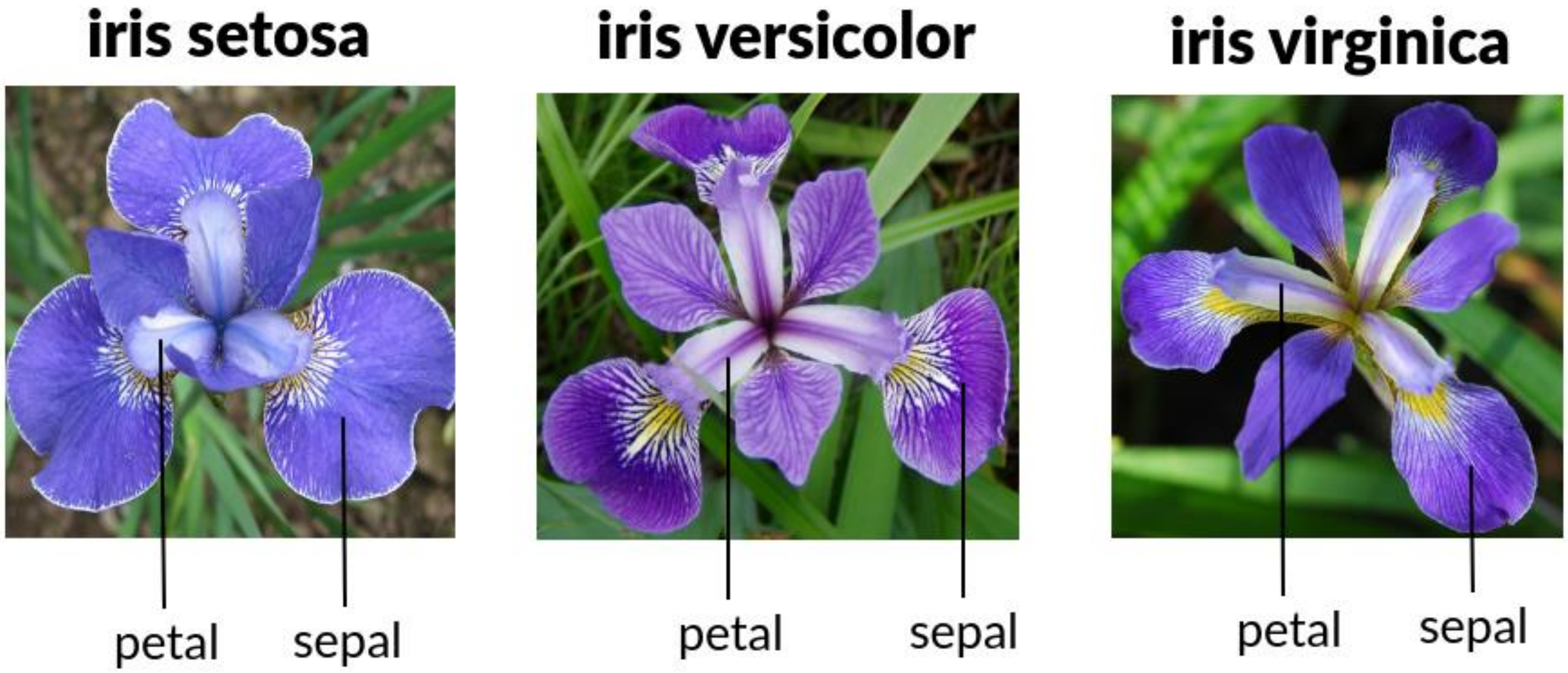
Some Examples

Application	Features	Labels
1. Self Driving Car	Image  	Steering Angle
2. Stock Market Share Value Prediction	Last 50 days stock Price 	Tomorrows Stock Price
3. Covid 19 Prediction using X-Ray Images	X-Ray Image  	Covid-19 Postive Covid-19 Negative
4. Chat Bots	Dialogue (Question)	Dialogue (Answer)
5. Object Detection APIs	Image  	Class (Cat, Dog,Apple etc)
6. Risk of having a heart disease	Age, Gender, TCL, HDL, Smoking	Risk (0-100%)


Machine Learning
Input
Feature extraction
Classification
Car
Not Car
Output
Deep Learning
Input
Feature extraction + Classification
Car
Not Car
Output


Industry Giants
Deep Learning
Quality Gap in AI products
Performance
Small and Medium size companies
Older algorithms
Amount of data

Iris Flower Example



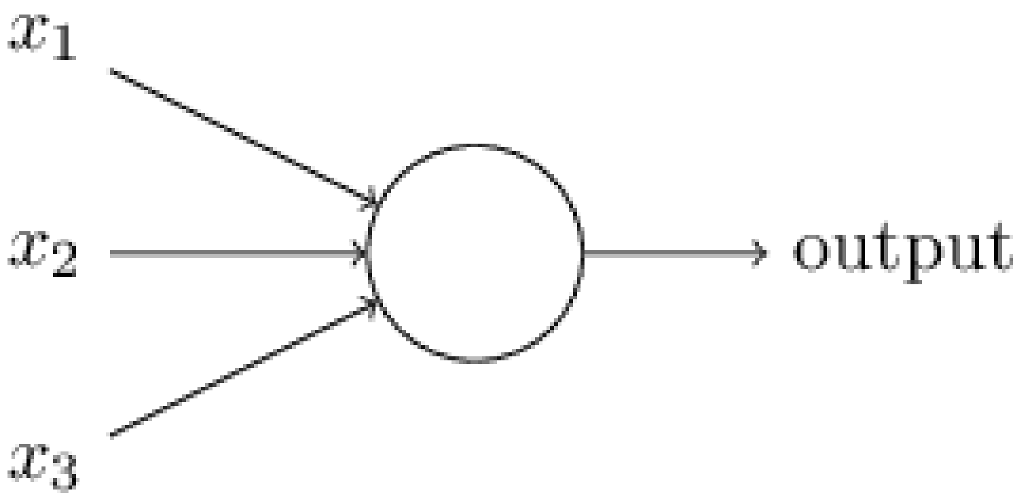
DEEP LEARNING

"In a neural network we don't tell the computer how to solve our problem. Instead, it learns from observational data, figuring out its own solution to the problem at hand."

-In 2006 was the discovery of techniques for learning in so-called deep neural networks. These techniques are now known as deep learning. They've been developed further, and today deep neural networks and deep learning achieve outstanding performance on many important problems in computer vision, speech recognition, and natural language processing-

History In Brief (1)

- The idea of neural networks began unsurprisingly as a model of how neurons in the brain function.
  - 1943: Portrayed with a simple electrical circuit by neurophysiologist Warren McCulloch and mathematician Walter Pitt
  - 1950-1960: Perceptrons were developed by the scientist Frank Rosenblatt,

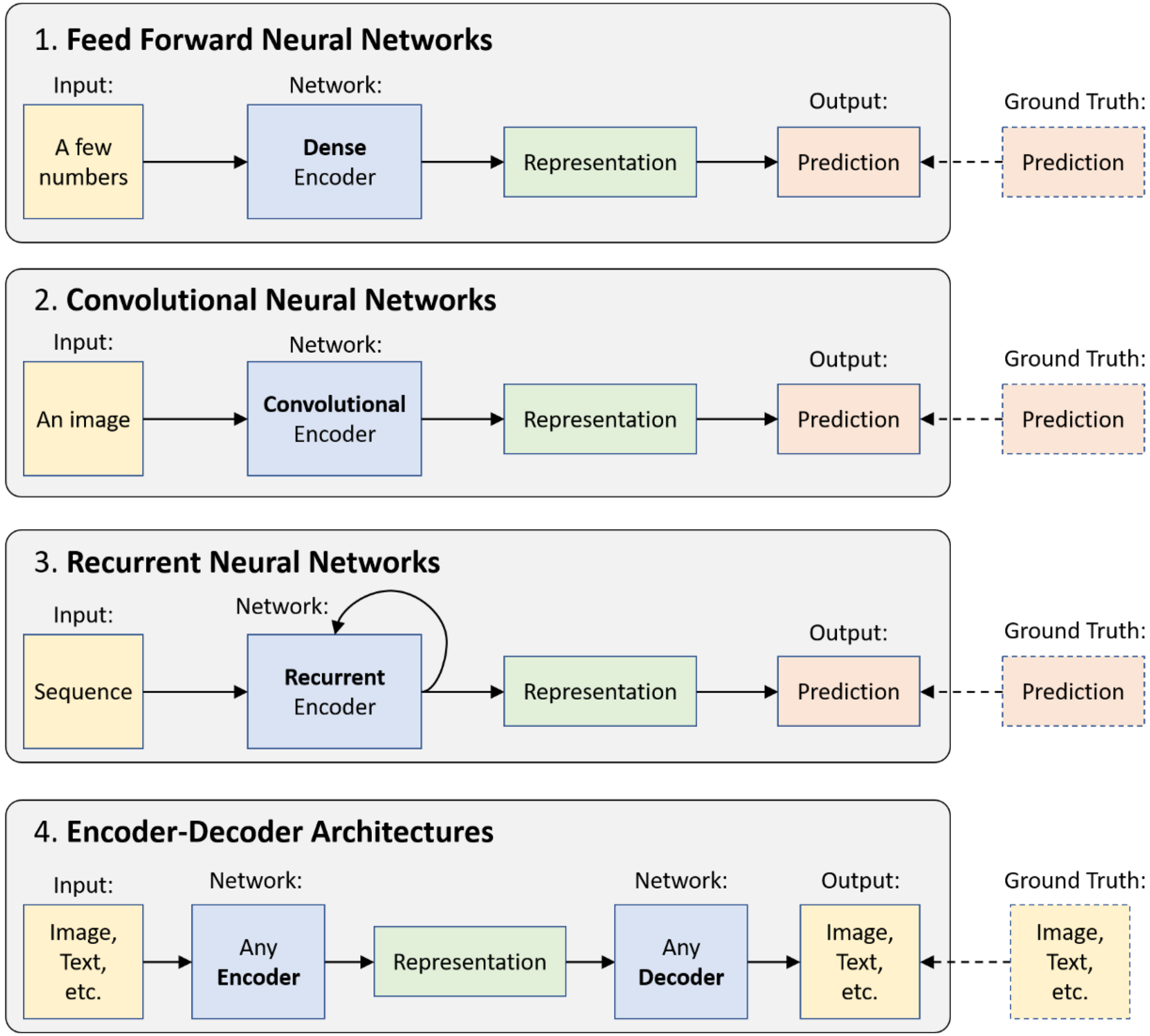


History In Brief (2)

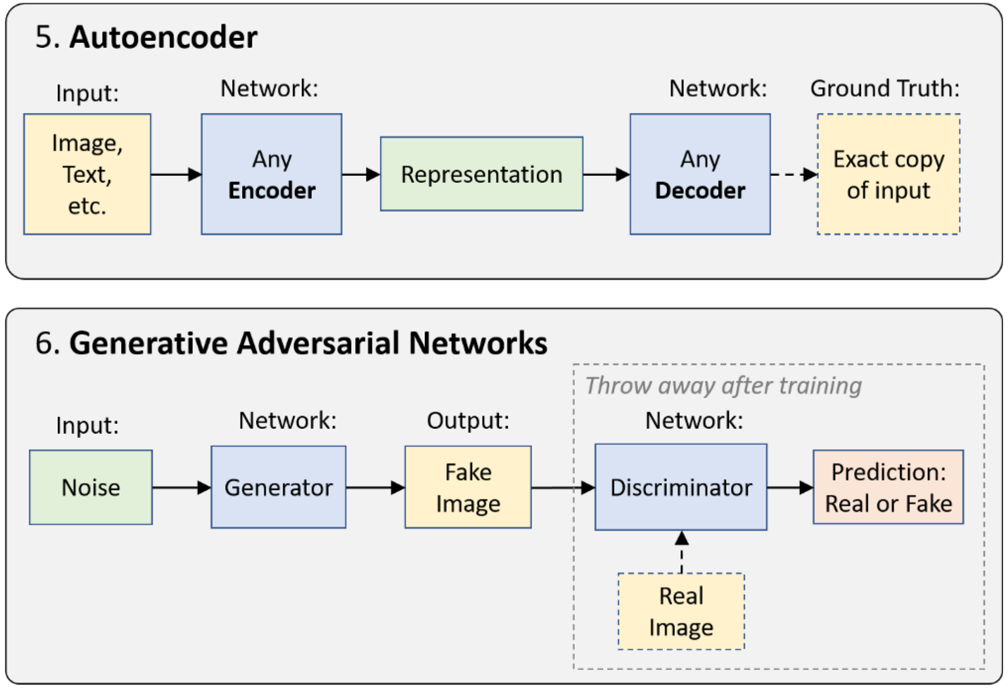
- 1974-86: Backpropagation Algorithm, Recurrent NL
- 1989-98: Convolutional Neural Networks, Bi Directional RNN, Long Short Term Memory (LSTM), MNIST Data Set
- 2006: “Deep Learning” Concept
- 2009: ImageNet
- 2012: AlexNet, Dropout
- 2014: DeepFace
- 2016: AlphaGo
- 2017: AlphaGo Zero
- 2018: BERT

Neural Network Architectures

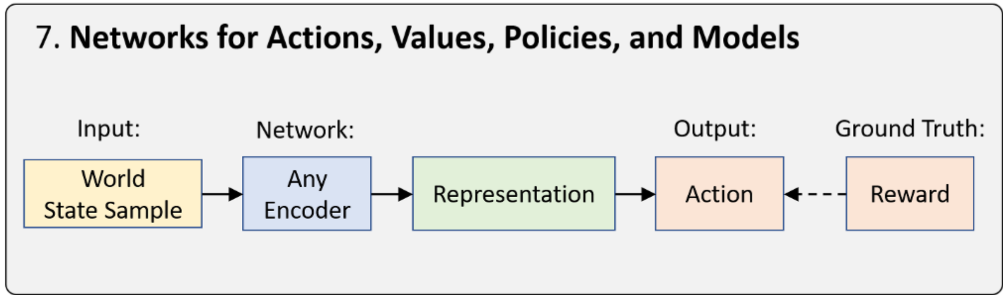
Supervised Learning



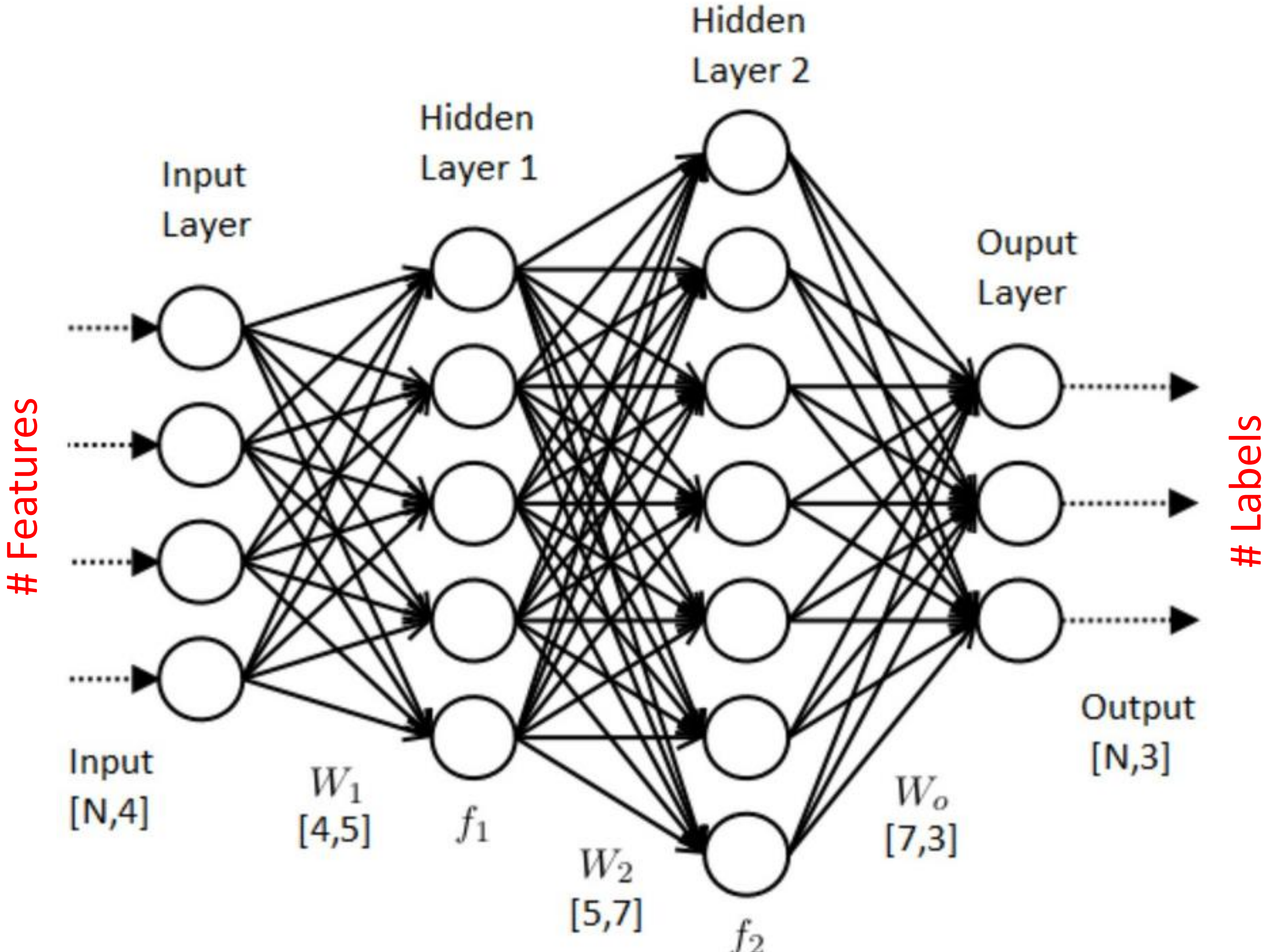
Unsupervised Learning



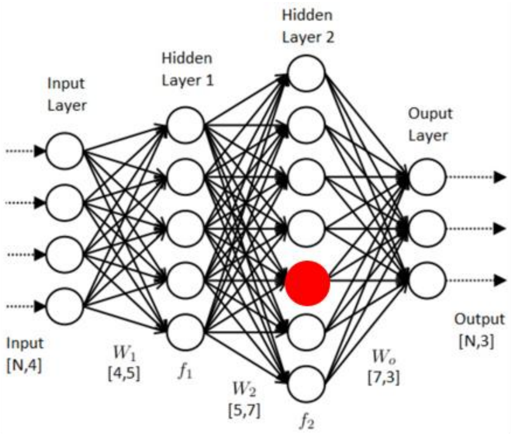
Reinforcement Learning



Feed Forward Neural Network Architecture



Feed Forward Neural Network Architecture (2)

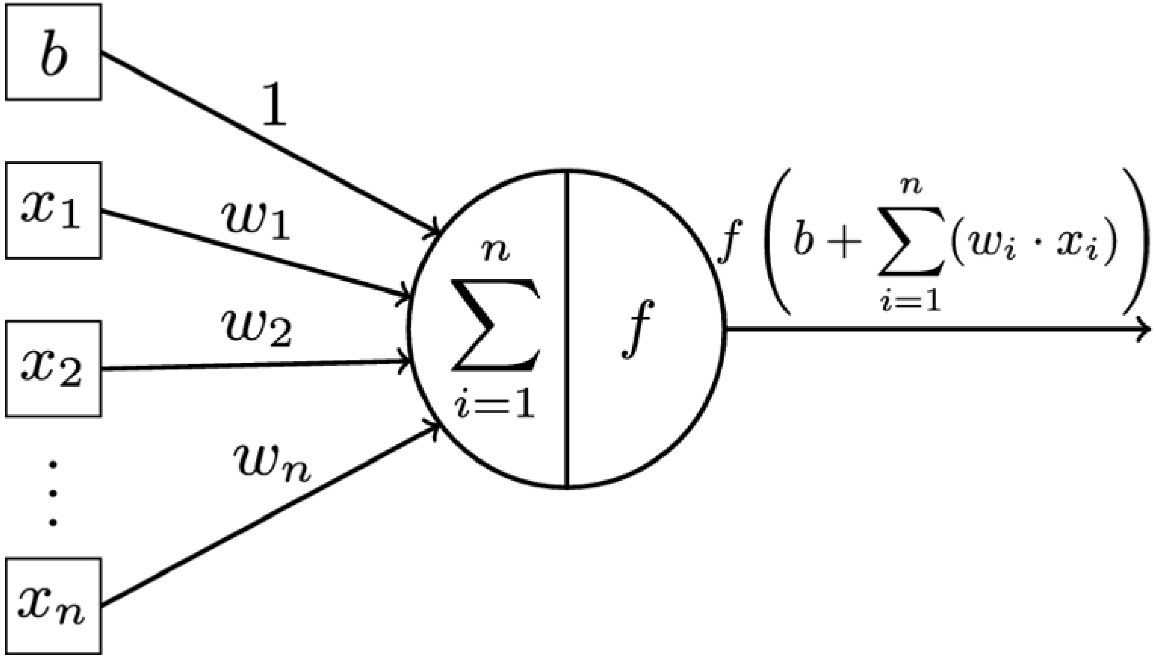


Weights (W) : All the Nets have

Biases (b) : All the Neuron other than neurons in the input layer has

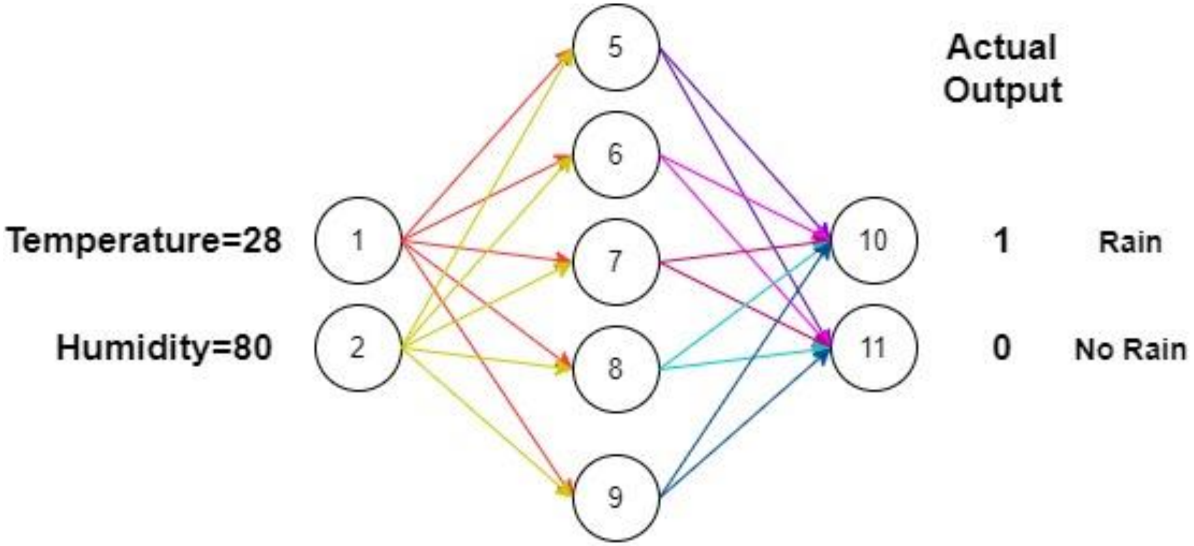
$$Z = \left(\sum_{i=1}^{n} wi \cdot xi\right) + \mathrm{b}$$

$$Y = F(Z)$$



Feed Forward Neural Network Architecture (3)

Features		Labels
Temperature (C)	Humidity %	
28	80	0
31	50	1
33	70	1
28	60	0
32	40	0
25	60	0



W15,W16,W17,W18,W19

W25,W26,W27,W28,W29

b5,b6,b7,b8,b9

W510,W511

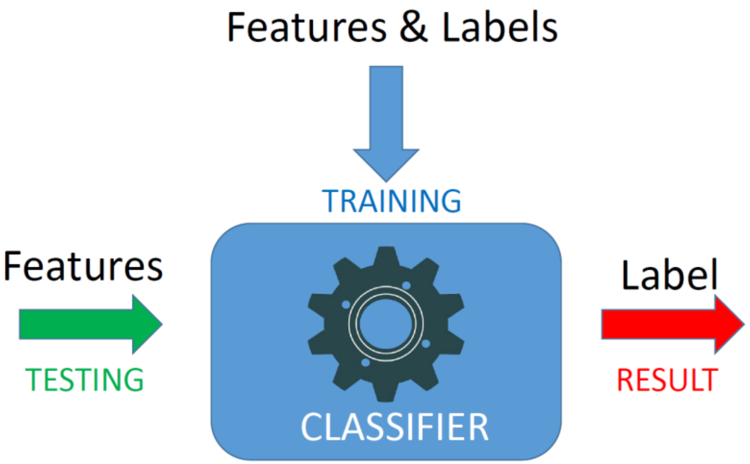
W610,W611

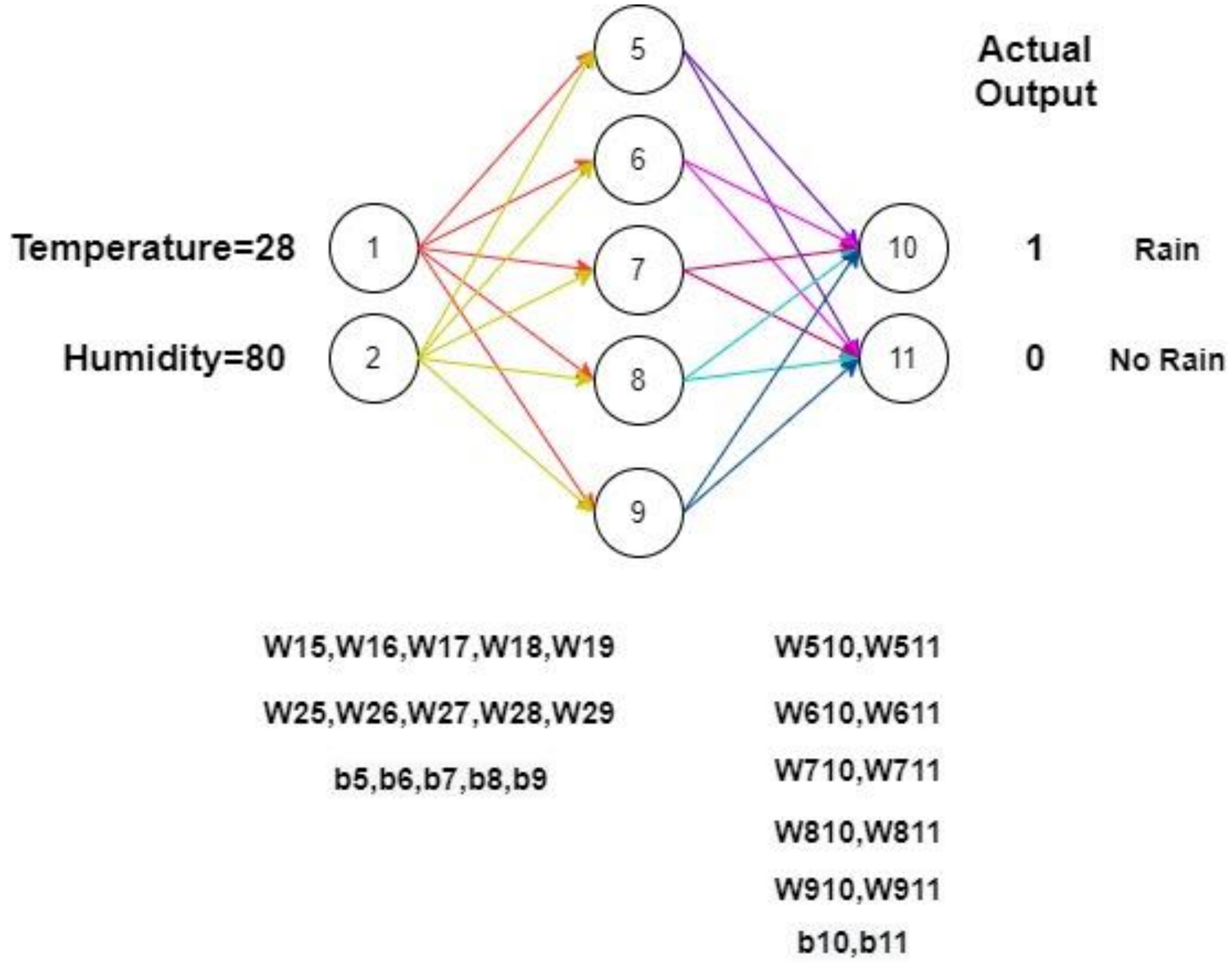
W710,W711

W810,W811

W910,W911

b10,b11



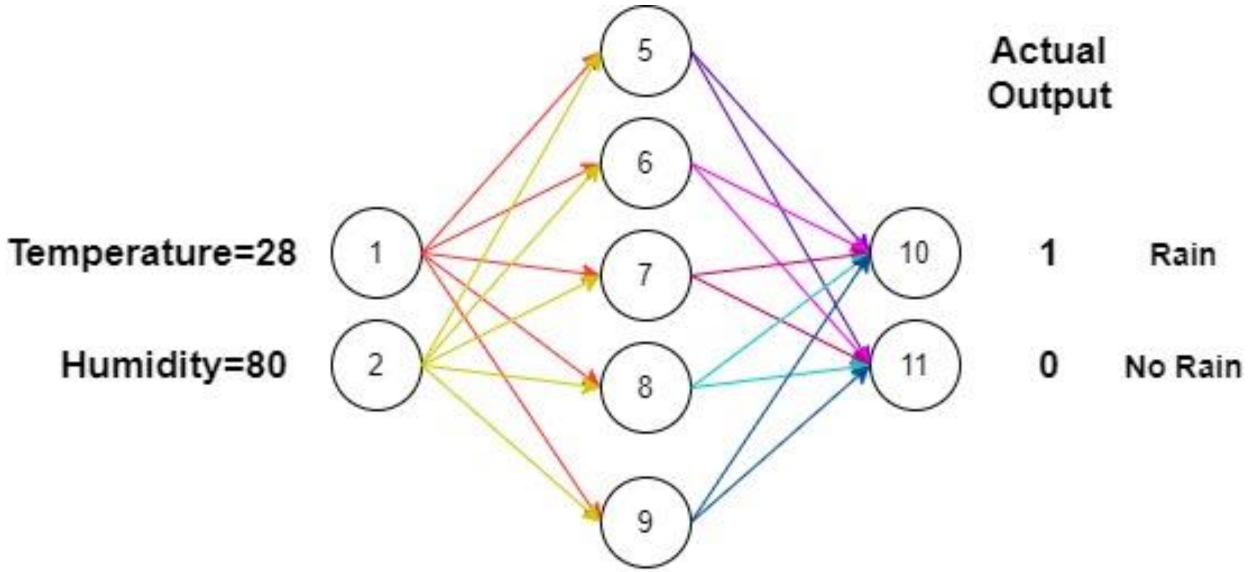


Layer 1 – 1st Hidden Layer

Y5=F(W15*X1 + W25*X1)
Y6=F(W16*X1 + W26*X1)
Y7=F(W17*X1 + W27*X1)
Y8=F(W18*X1 + W28*X1)
Y9=F(W19*X1 + W29*X1)

Layer 2 – Output Layer

Y10=F(W510*Y5 + W610*Y6 + W710*Y7 + W810*Y8 + W910*Y9)
Y11=F(W510*Y5 + W610*Y6 + W710*Y7 + W810*Y8 + W910*Y9)



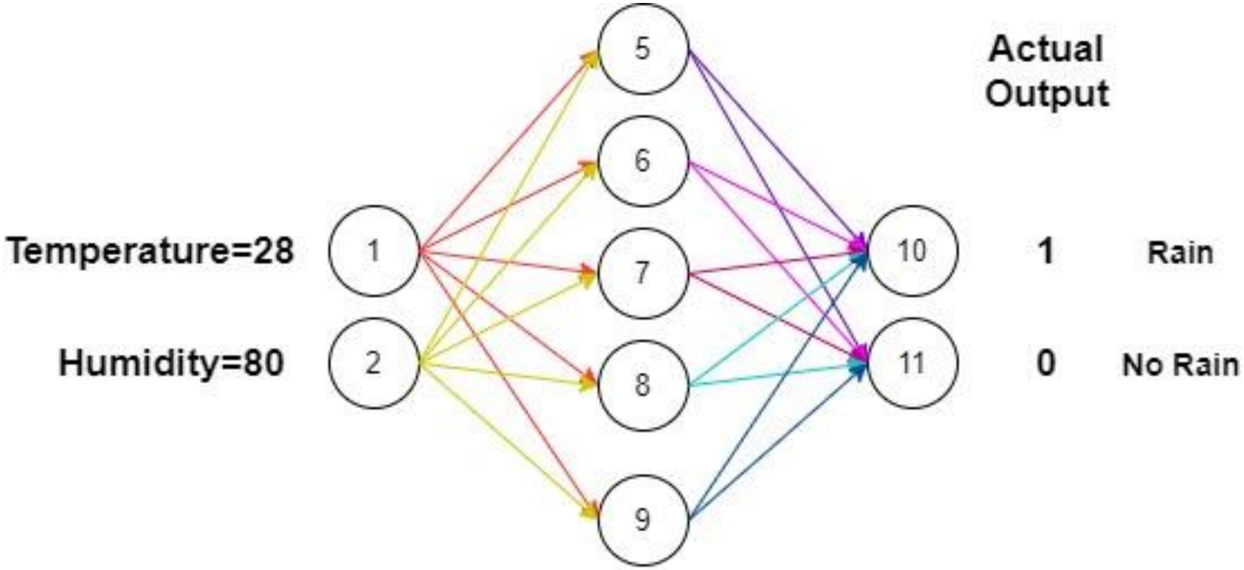
Layer 1 – 1st Hidden Layer

Y5=F(W15*X1 + W25*X1)
Y6=F(W16*X1 + W26*X1)
Y7=F(W17*X1 + W27*X1)
Y8=F(W18*X1 + W28*X1)
Y9=F(W19*X1 + W29*X1)

Layer 2 – Output Layer

Y10=F(W510*Y5 + W610*Y6 + W710*Y7 + W810*Y8 + W910*Y9)
Y11=F(W510*Y5 + W610*Y6 + W710*Y7 + W810*Y8 + W910*Y9)

Y predicted (Y10 and Y11) = G (Ws,Bs,Xs)



Layer 1 – 1st Hidden Layer

Y5=F(W15*X1 + W25*X1)
Y6=F(W16*X1 + W26*X1)
Y7=F(W17*X1 + W27*X1)
Y8=F(W18*X1 + W28*X1)
Y9=F(W19*X1 + W29*X1)

Layer 2 – Output Layer

Y10=F(W510*Y5 + W610*Y6 + W710*Y7 + W810*Y8 + W910*Y9)
Y11=F(W510*Y5 + W610*Y6 + W710*Y7 + W810*Y8 + W910*Y9)

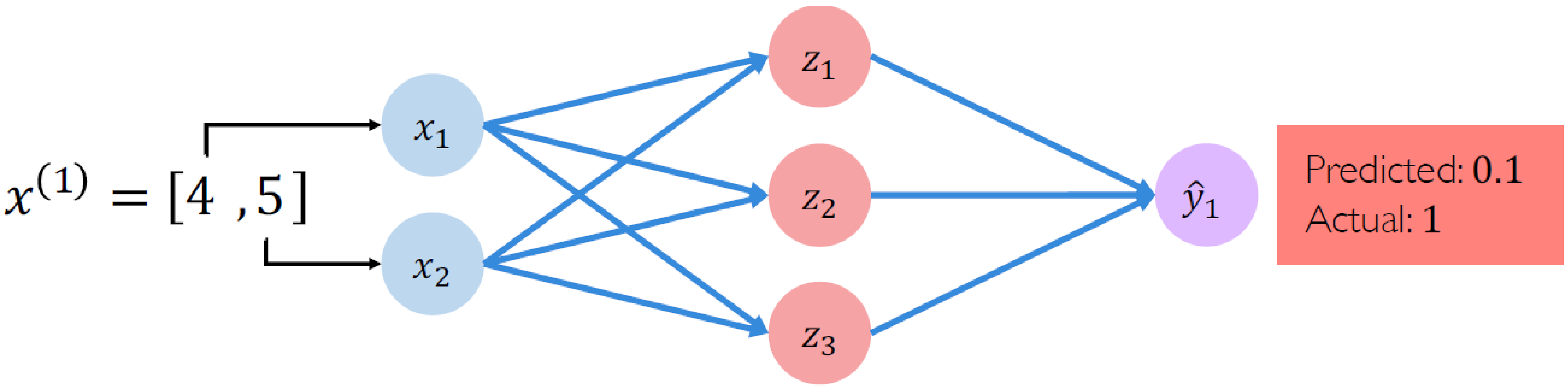
Y predicted (Y10 and Y11) = G (Ws,Bs,Xs)

Y Predicted (Output of the NN)	Y actual
Y10 = 0.2	1
Y11 = 0.8	0

How to Correct this?

Loss Functions

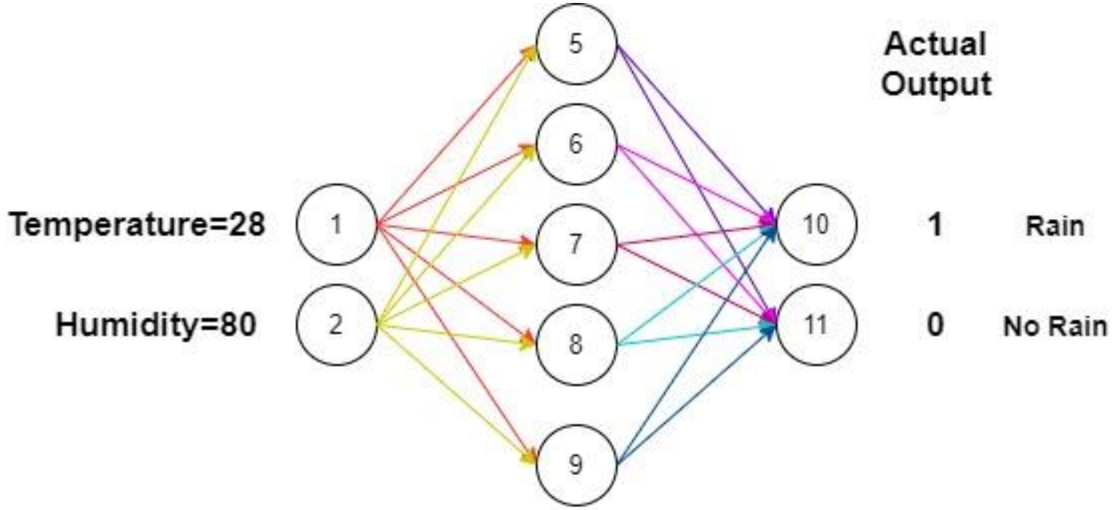
The loss of our network measures the cost incurred from incorrect predictions



$$\mathcal{L}\Big(\underbrace{f\big(x^{(i)}; \boldsymbol{W}\big)}_{\text{Predicted}}, \underbrace{y^{(i)}}_{\text{Actual}}\Big)$$

Error/Loss

Y Predicted (Output of the NN)	Y actual
Y10 = 0.2	1
Y11 = 0.8	0



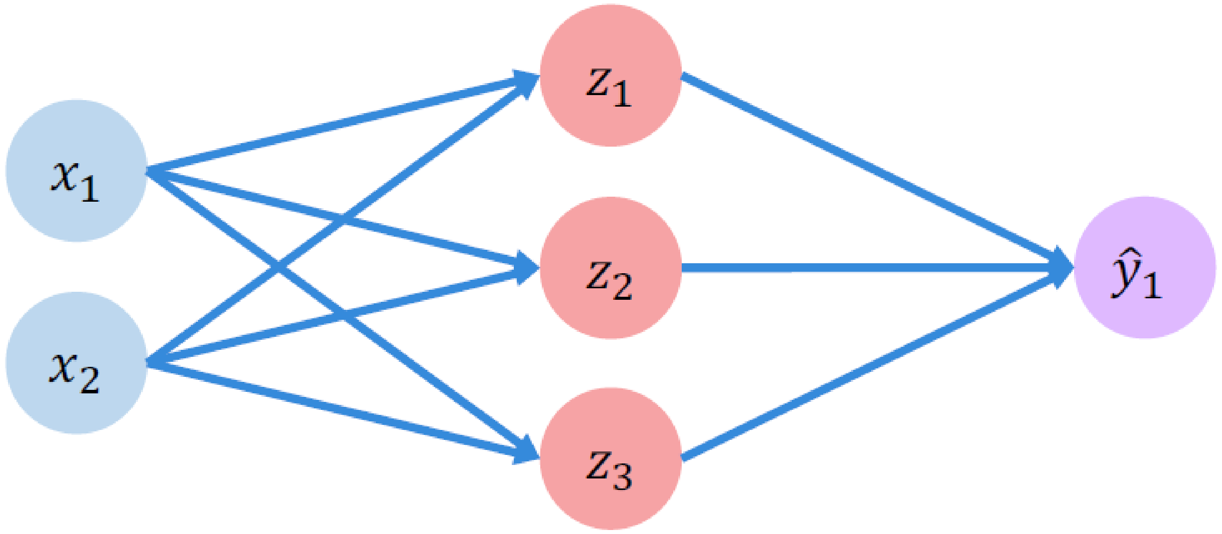
$$E = G\,(Y\ actual - Y\ predicted\ )$$

$$E = G\,(Ws, Bs) - \text{27 Parameters}$$

Mean Squared Error

Mean squared error loss can be used with regression models that output continuous real numbers

$$\boldsymbol{X} = \begin{bmatrix} 4, & 5 \\ 2, & 1 \\ 5, & 8 \\ \vdots & \vdots \end{bmatrix}$$

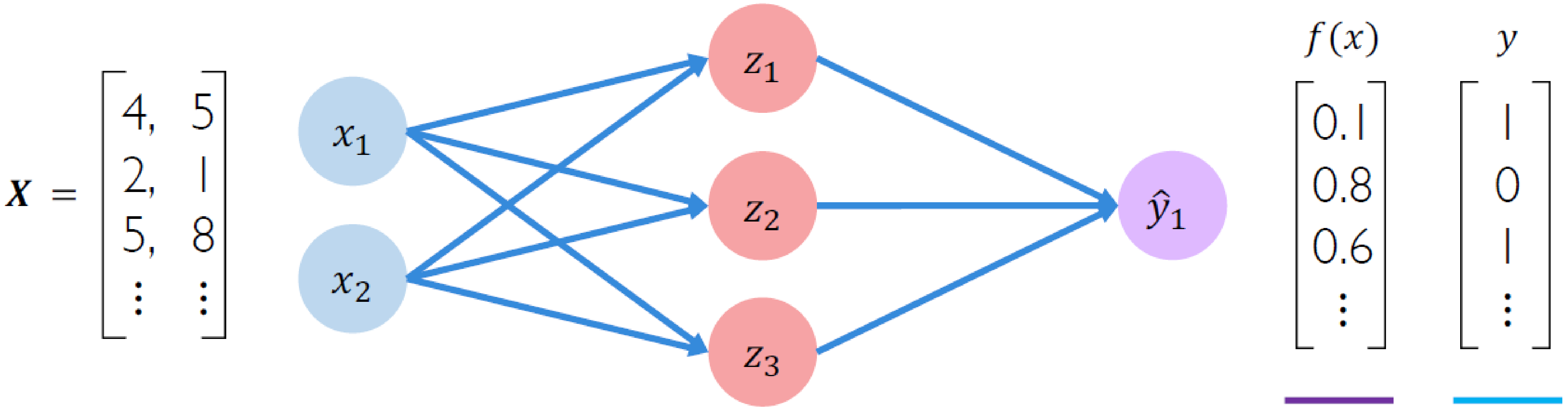


$$\underbrace{\begin{array}{cc} f(x) & y \\ \begin{bmatrix} 30 \\ 80 \\ 85 \\ \vdots \end{bmatrix} & \begin{bmatrix} 90 \\ 20 \\ 95 \\ \vdots \end{bmatrix} \end{array}}_{\text{Final Grades (percentage)}}$$

$$\boldsymbol{J}(\boldsymbol{W}) = \frac{1}{n}\sum_{i=1}^{n}\Big(\underbrace{y^{(i)}}_{\text{Actual}} - \underbrace{f\big(x^{(i)};\boldsymbol{W}\big)}_{\text{Predicted}}\Big)^2$$

Cross Entropy

Cross entropy loss can be used with models that output a probability between 0 and 1



$$\boldsymbol{J}(\boldsymbol{W}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{y^{(i)}}_{\text{Actual}} \log\Big(\underbrace{f\big(x^{(i)};\boldsymbol{W}\big)}_{\text{Predicted}}\Big) + (1-\underbrace{y^{(i)}}_{\text{Actual}})\log\Big(1-\underbrace{f\big(x^{(i)};\boldsymbol{W}\big)}_{\text{Predicted}}\Big)$$

Loss Optimization

We want to find the network weights that achieve the lowest loss

$$\boldsymbol{W}^* = \underset{\boldsymbol{W}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(f\left(x^{(i)}; \boldsymbol{W}\right), y^{(i)}\right)$$

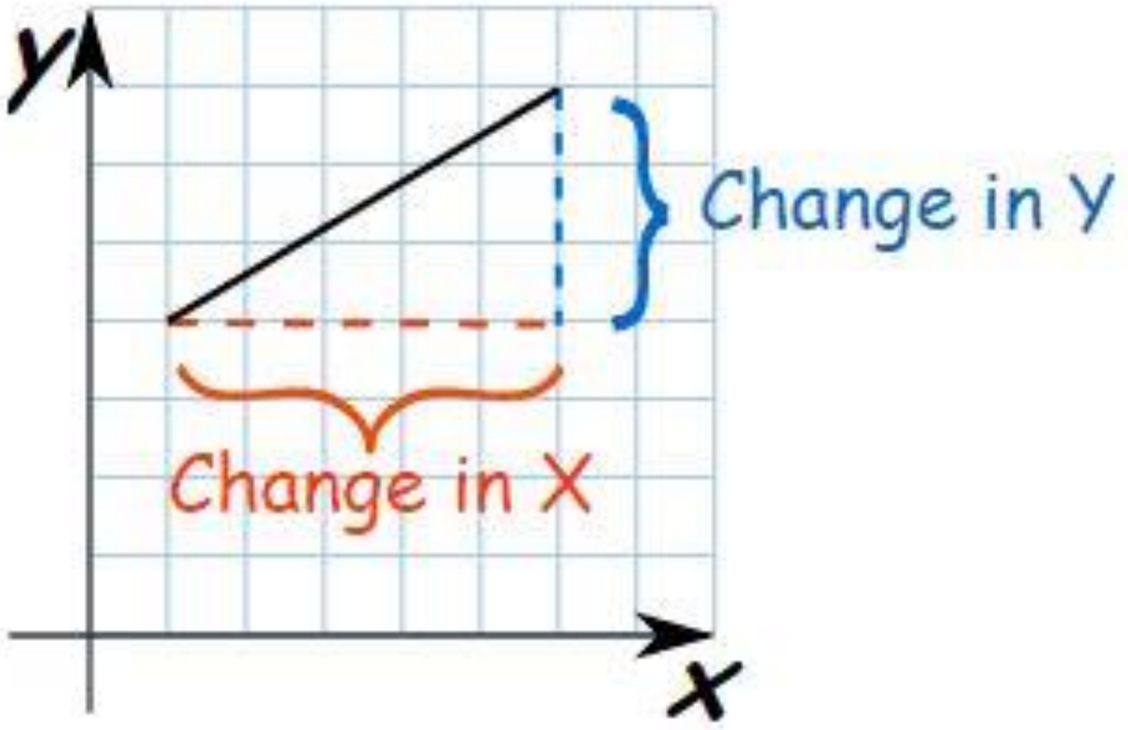
$$\boldsymbol{W}^* = \underset{\boldsymbol{W}}{\operatorname{argmin}}\, J(\boldsymbol{W})$$

Remember:

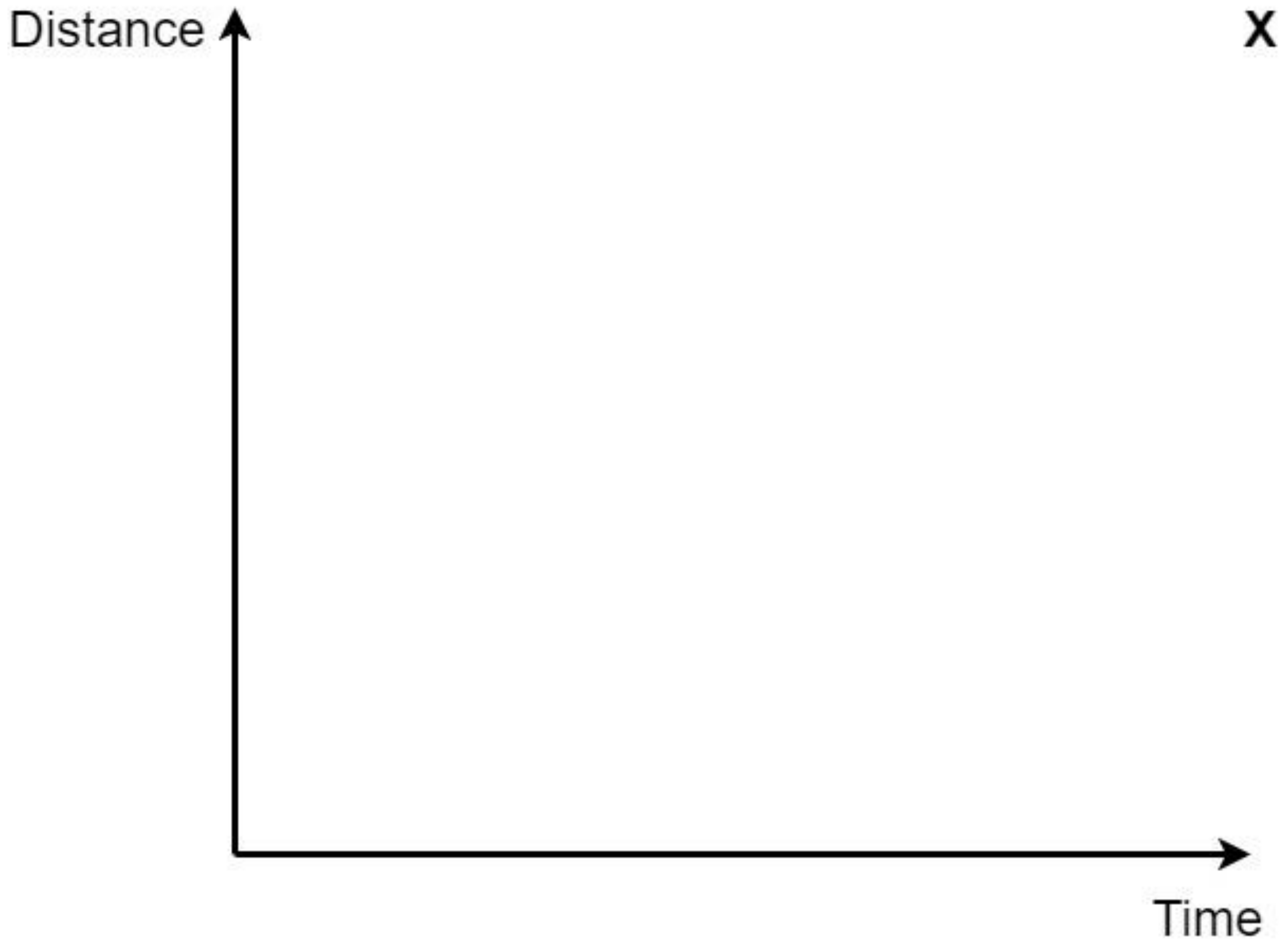
$$\boldsymbol{W} = \left\{\boldsymbol{W}^{(0)}, \boldsymbol{W}^{(1)}, \cdots\right\}$$

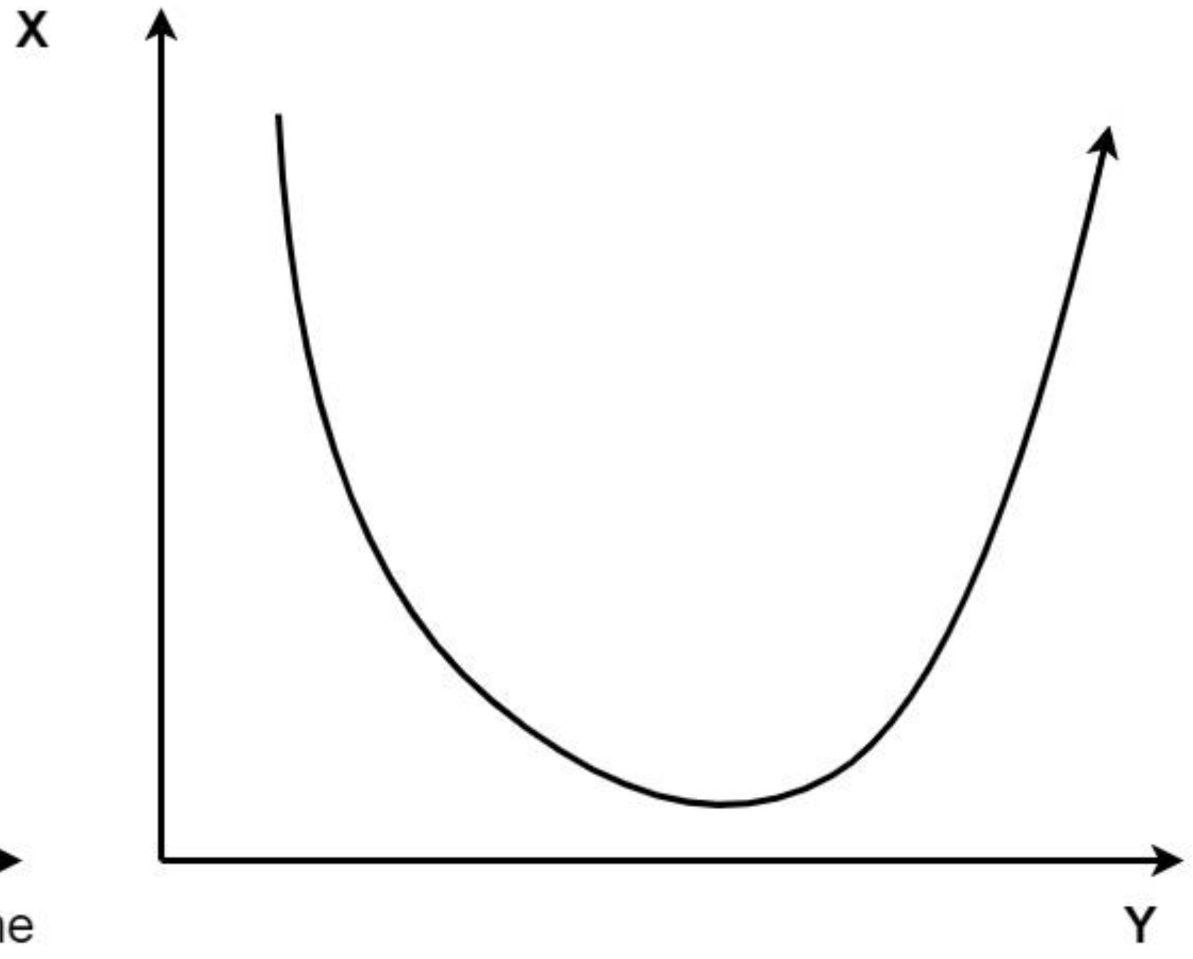
Gradient

$$\textbf{Gradient} = \frac{\text{Change in Y}}{\text{Change in X}}$$




E
W1

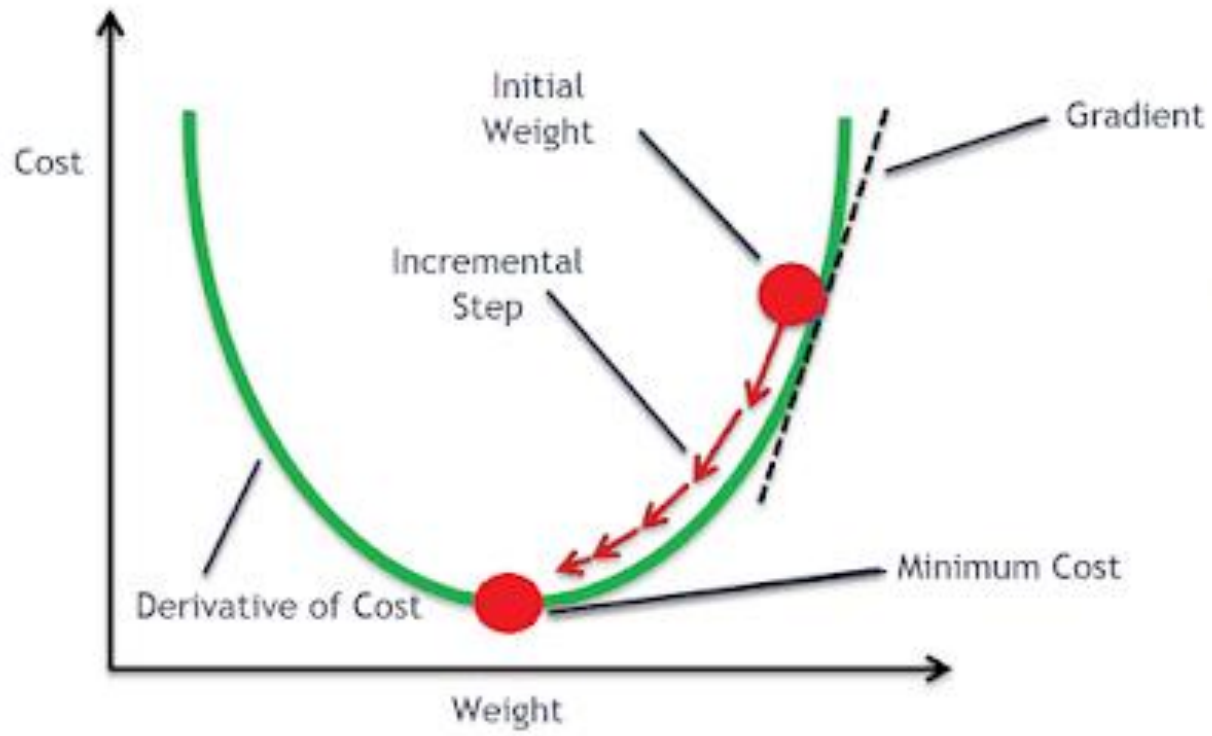

Distance
Time

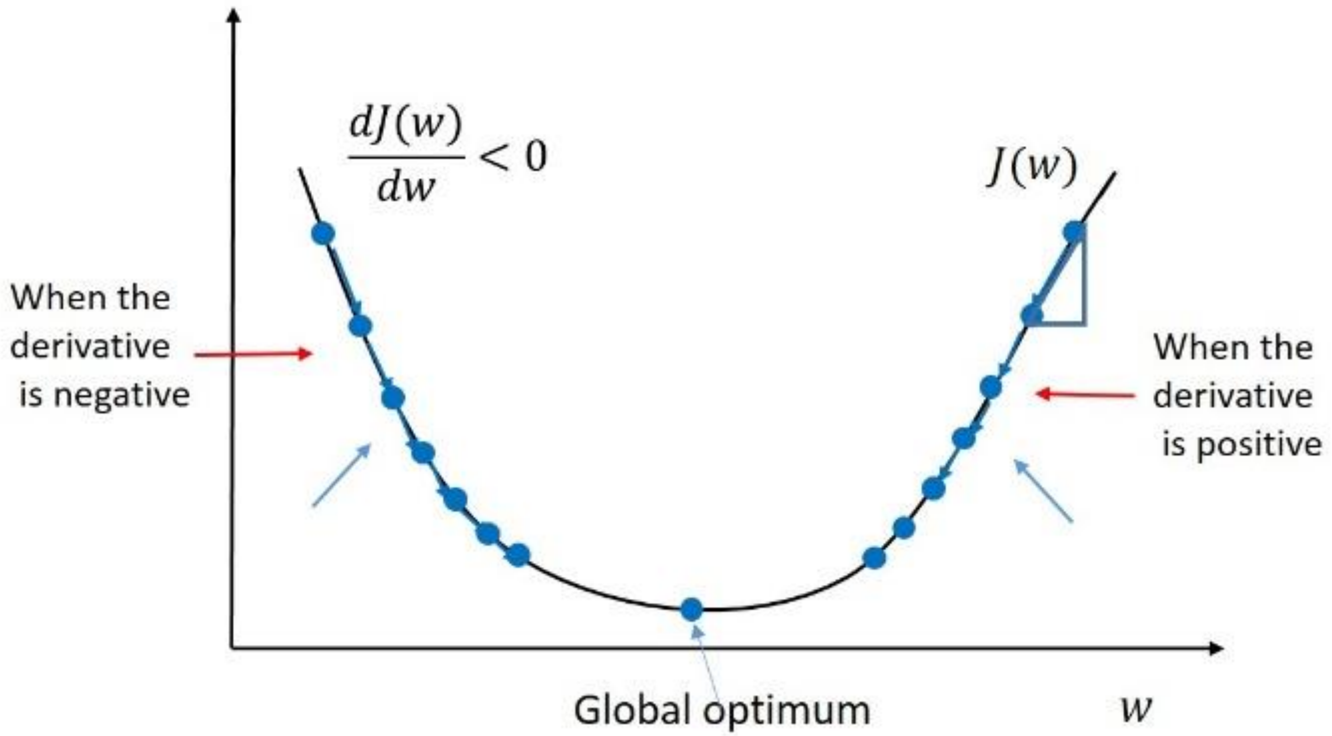

X
Y

Gradient Descent

Compute gradient, $\dfrac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$

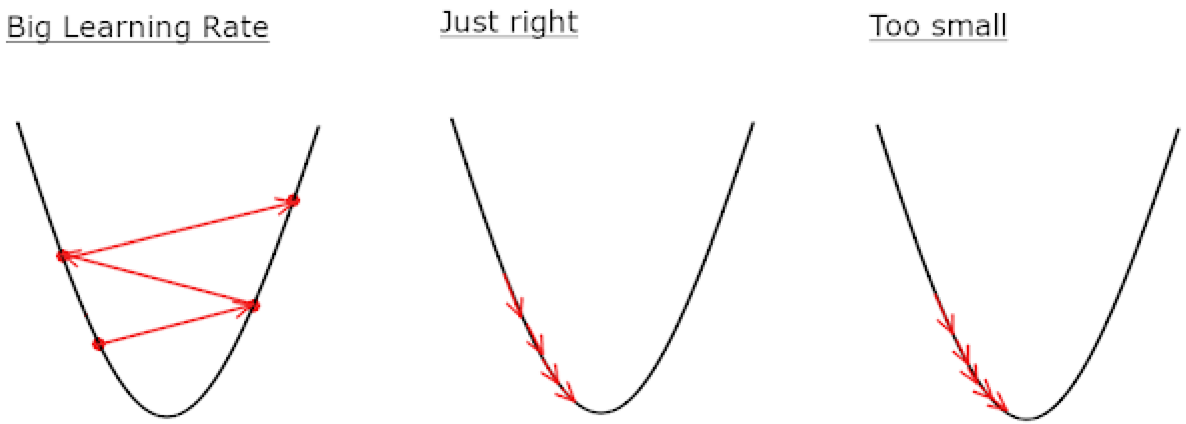
Update weights, $\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \dfrac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$

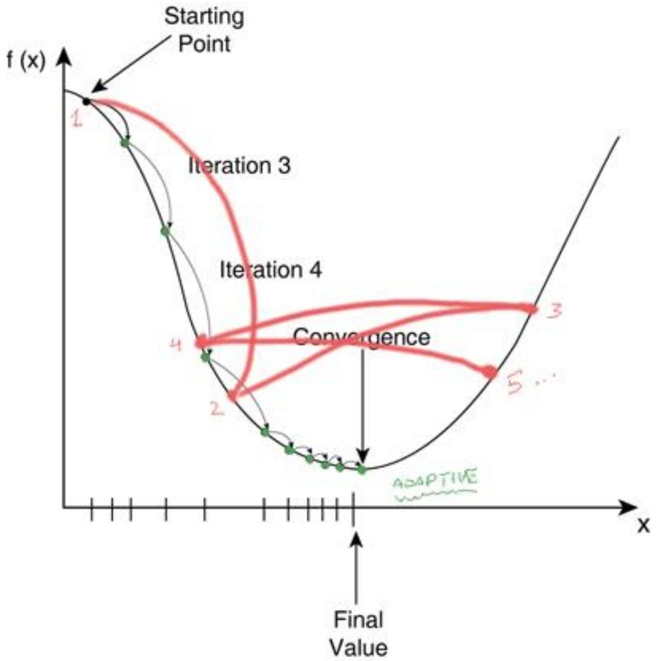


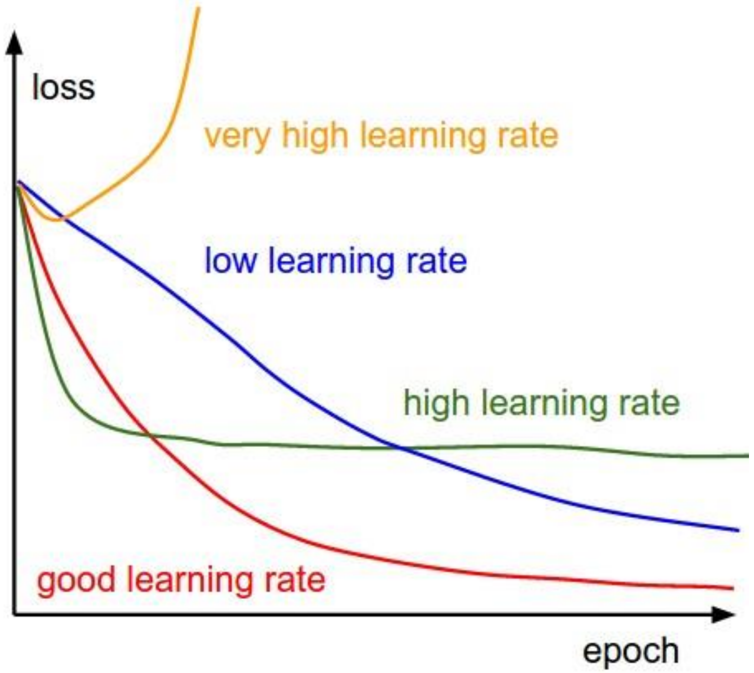



E
W1

The Learning Rate (η)







Adam

- Adam stands for **Adaptive Moment Estimation.** Adaptive Moment Estimation (Adam) is another method that computes adaptive learning rates for each parameter.

AdaDelta

- It is an extension of **AdaGrad** which tends to remove the *decaying learning Rate* problem of it. Instead of accumulating all previous squared gradients, ***Adadelta*** limits the window of accumulated past gradients to some fixed size **w**.

Adagrad

- It simply allows the learning Rate -**η** to **adapt** based on the parameters. So it makes big updates for infrequent parameters and small updates for frequent parameters. For this reason, it is well-suited for dealing with sparse data.

Gradient Vector and backpropagation

$$W = \begin{bmatrix} w1 \\ w2 \\ w3 \\ . \\ . \\ wn \end{bmatrix} \qquad -\eta \frac{\partial J(W)}{\partial W} = \begin{bmatrix} \Delta w1 \\ \Delta w2 \\ \Delta w3 \\ . \\ . \\ \Delta wn \end{bmatrix}$$