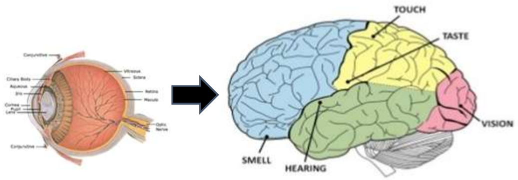
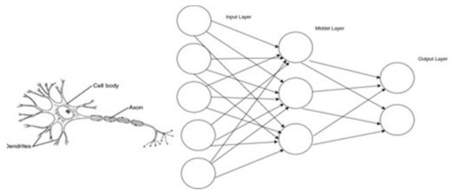
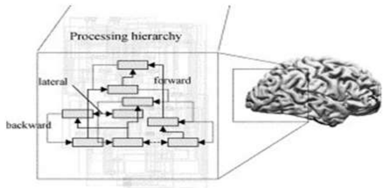
Artificial Intelligence

Dr. Eng. Wajdi SAADAOUI
Assistant Professor, ENIT, ENSI, ISSAT, Higher Education Ministry, Tunisia

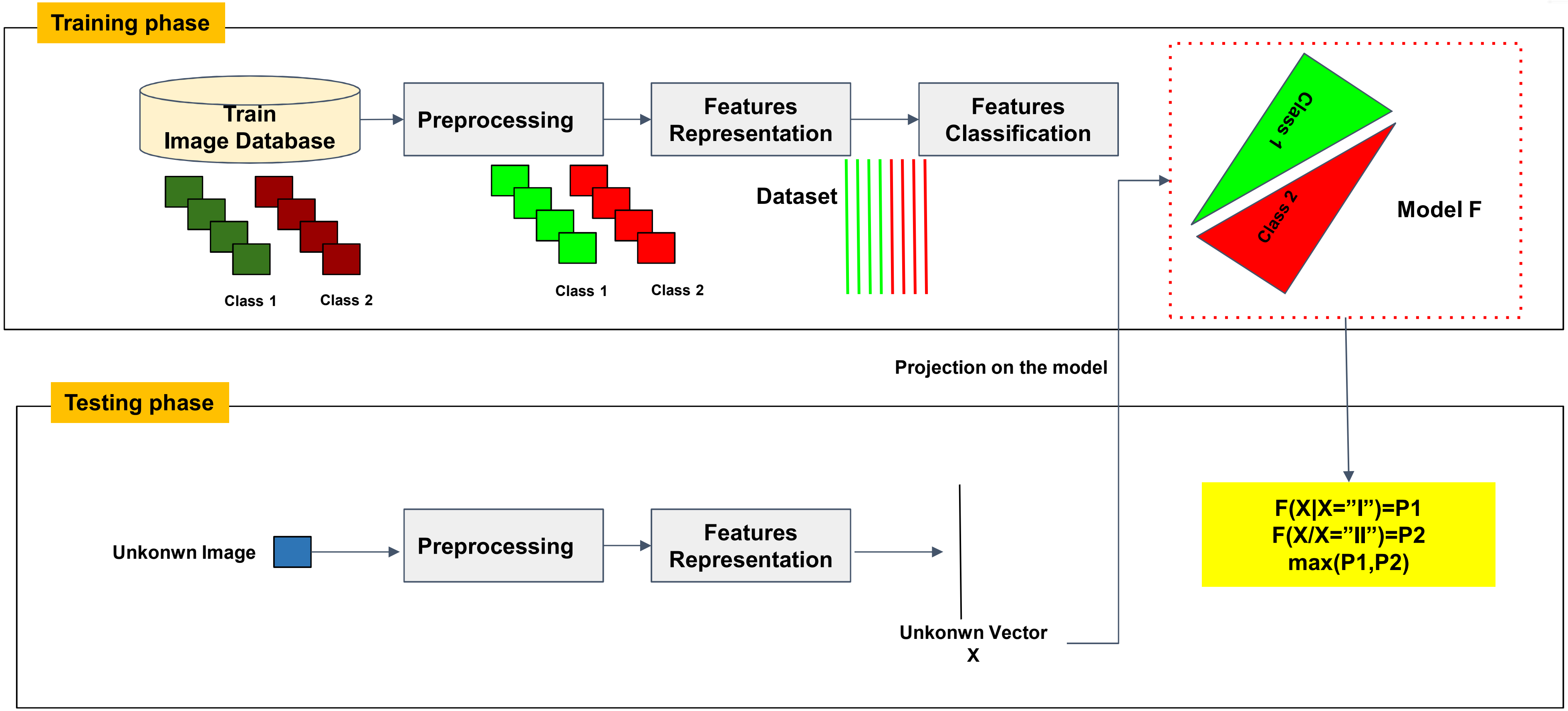
Outline

1. Artificial Intelligence (AI): from perception to reasoning
2. How to design and use a Machine Learning (ML)?
3. Machine Learning Techniques: A brief Review & Comparison
4. Neural Network: Theory and Application
5. Naïve Bayes: Theory and Application
6. Support Vector Machines (SVM): Theory and Application
7. How to select the appropriate Machine Learning
8. How to evaluate a Machine Learning Performance?

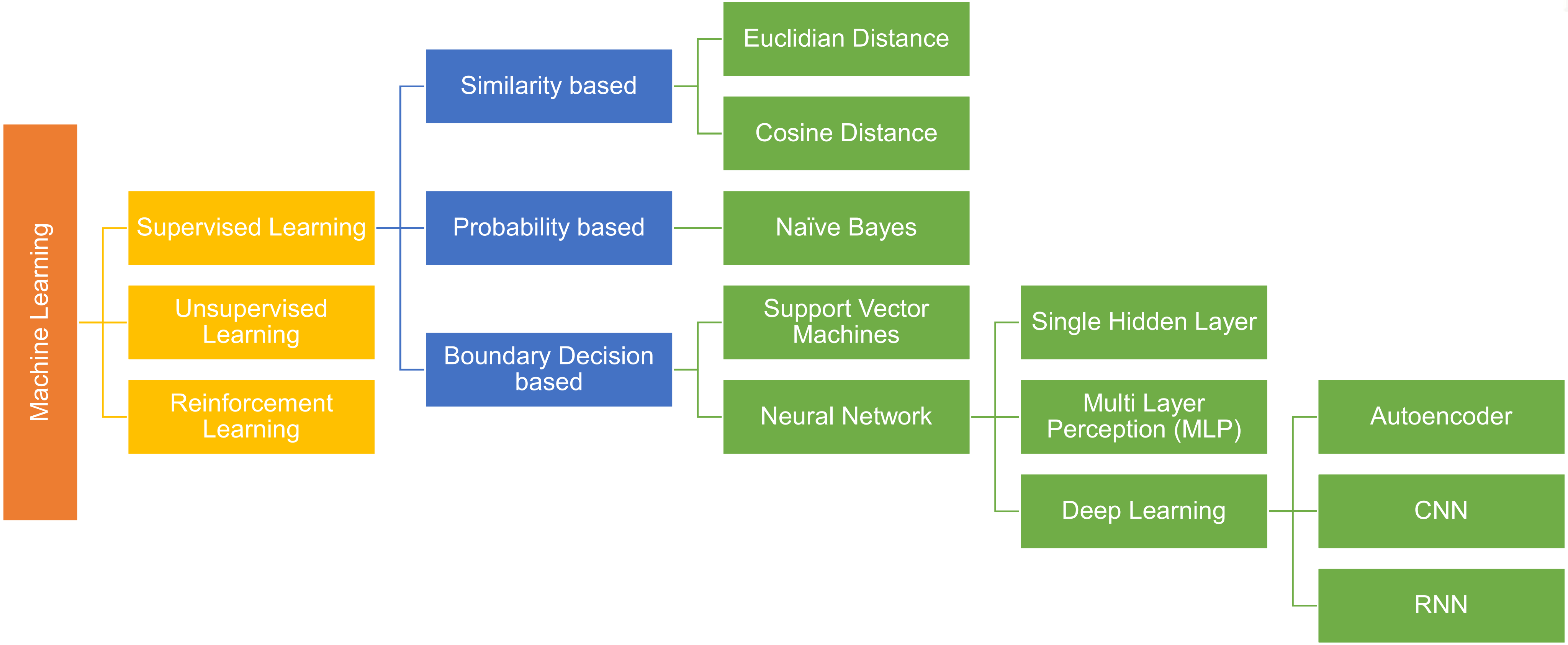
1. Artificial Intelligence (AI): from perception to reasoning

Intelligence		Artificial Intelligence
Perception *Living beings*		Image Processing
Optimization *Living beings*		Bio-Inspired Optimization
Learning *Baby, Animal, etc.*		Machine Learning
Reasoning *Human*		Fuzzy Logic

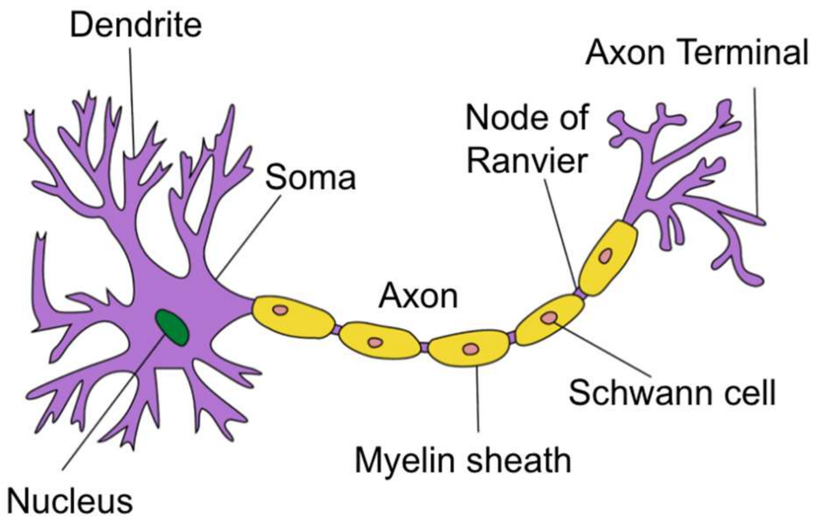
2. How to design and use a Machine Learning (ML)?



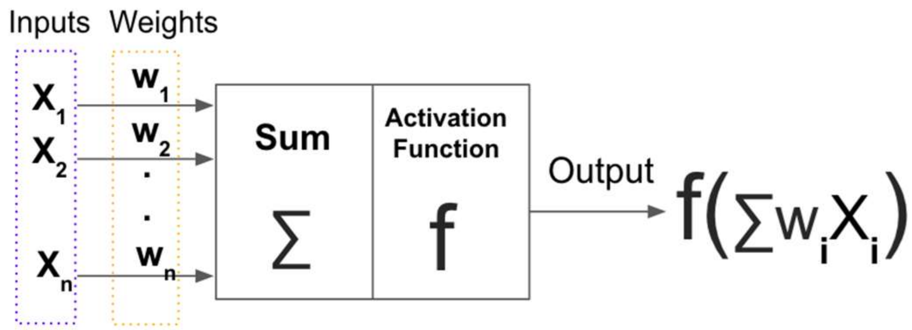
3. Machine Learning Techniques: A brief Review & Comparison



4. Neural Network: Theory and Application



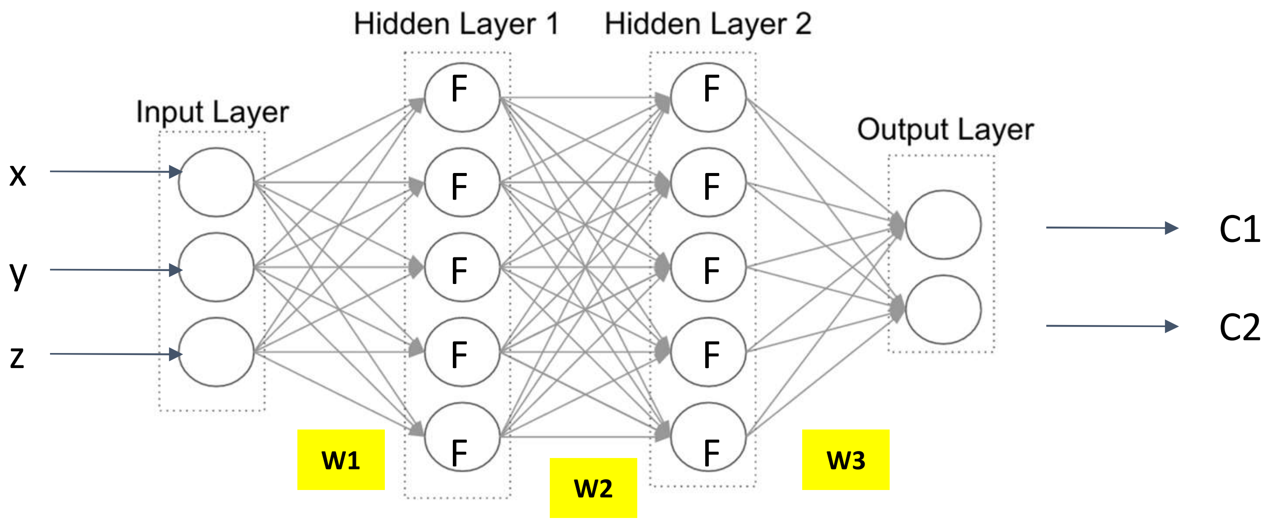
Structure of a typical neuron



Structure of artificial neuron

4. Neural Network: Theory and Application

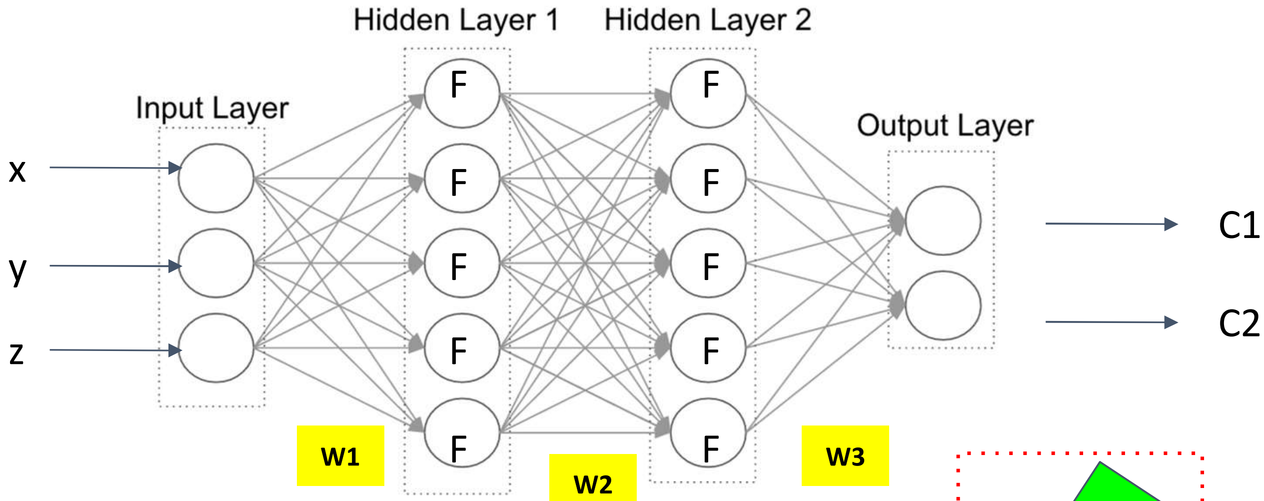
3x5x5x2



Feedforward neural network with 2 hidden layers

[x,y,z]: Input Vector
W1: Weight Matrix of Input Layer
W2: Weight Matrix of Hidden Layer
W3: Weight Matrix of Output Layer
F: Activation Function
C1: Class Output 1
C2: Class Output 2

4. Neural Network: Theory and Application

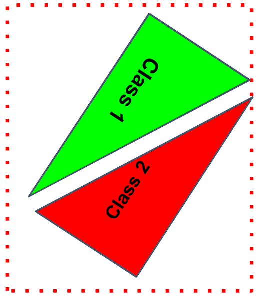


Feedforward neural network with 2 hidden layers

Input: 1x3 to classify

(N,M) x (MxP) = (N,P)

F(X|X="I")=P1
F(X/X="II")=P2
max(P1,P2)



Model F [P1, P2]

1x3 × 3x5 = 1x5 × 5x5 = 1x5 × 5x2 = 1x2

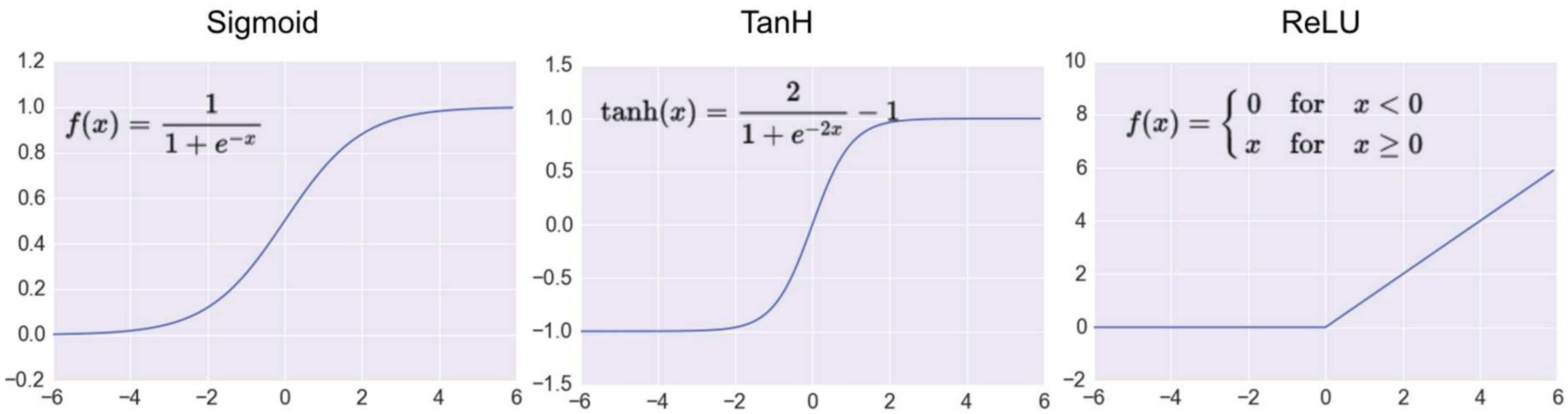
W1
$$\begin{pmatrix} w11 & w12 & w13 & w14 & w15 \\ w21 & w22 & w23 & w24 & w25 \\ w31 & w32 & w33 & w34 & w35 \end{pmatrix}$$

W2
$$\begin{pmatrix} w'11 & w'12 & w'13 & w'14 & w'15 \\ w'21 & w'22 & w'23 & w'24 & w'25 \\ w'31 & w'32 & w'33 & w'34 & w'35 \\ w'41 & w'42 & w'43 & w'44 & w'45 \\ w'51 & w'52 & w'53 & w'54 & w'55 \end{pmatrix}$$

W3
$$\begin{pmatrix} w''11 & w''12 \\ w''21 & w''22 \\ w''31 & w''32 \\ w''41 & w''42 \\ w''51 & w''52 \end{pmatrix}$$

4. Neural Network: Theory and Application

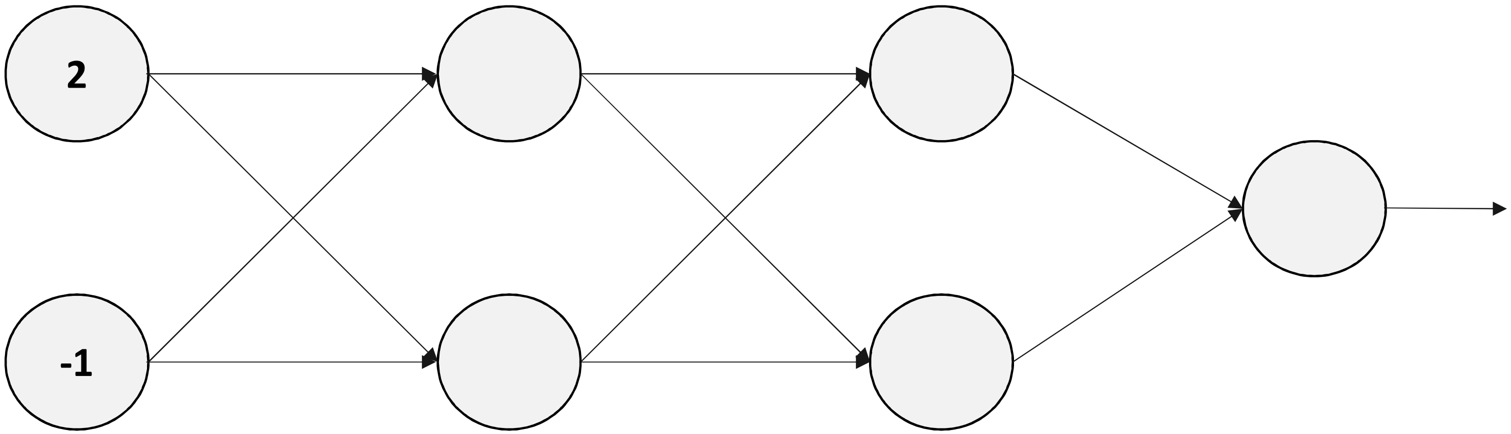
F= Nonlinear Activation Function to insert a Non Linear Representation into Neural Network



4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

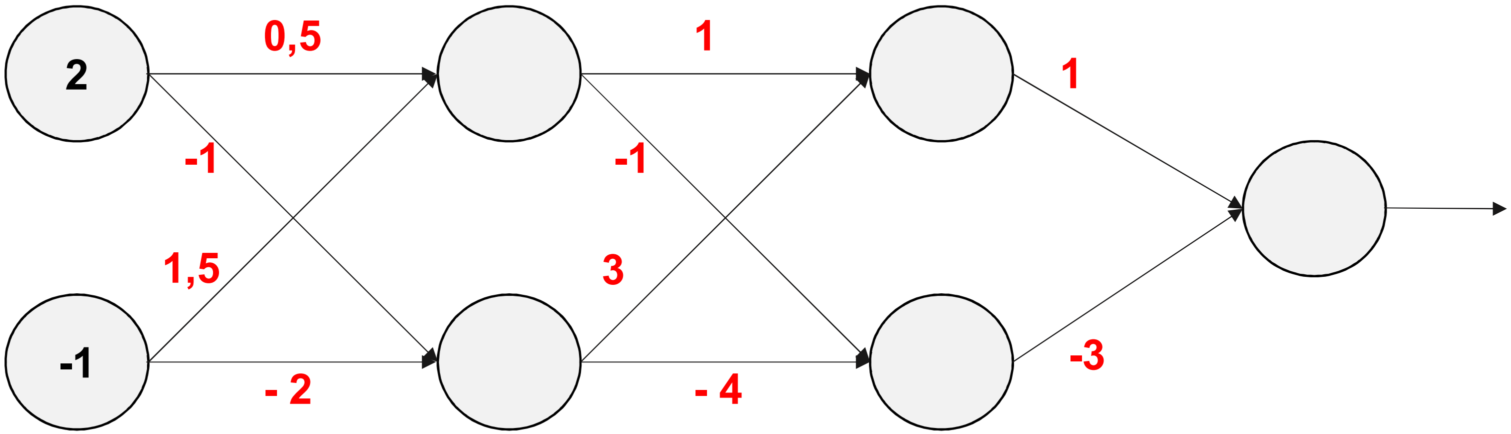


4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

Step 1: Weights' Initialization

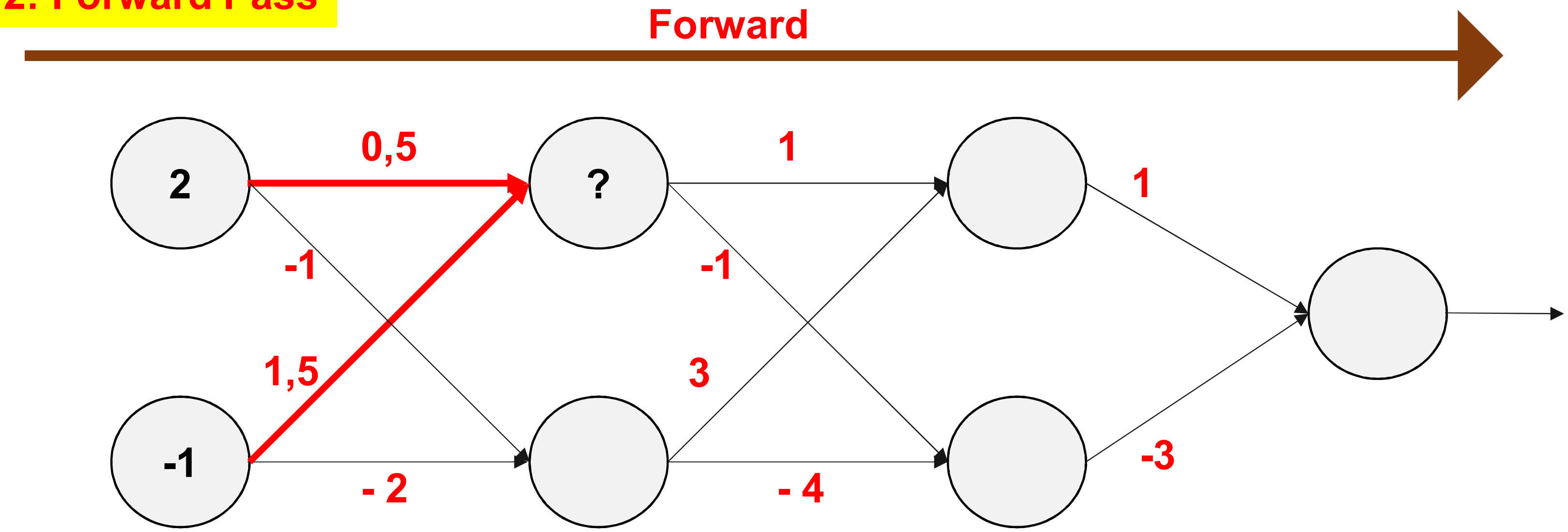


4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

Step 2: Forward Pass

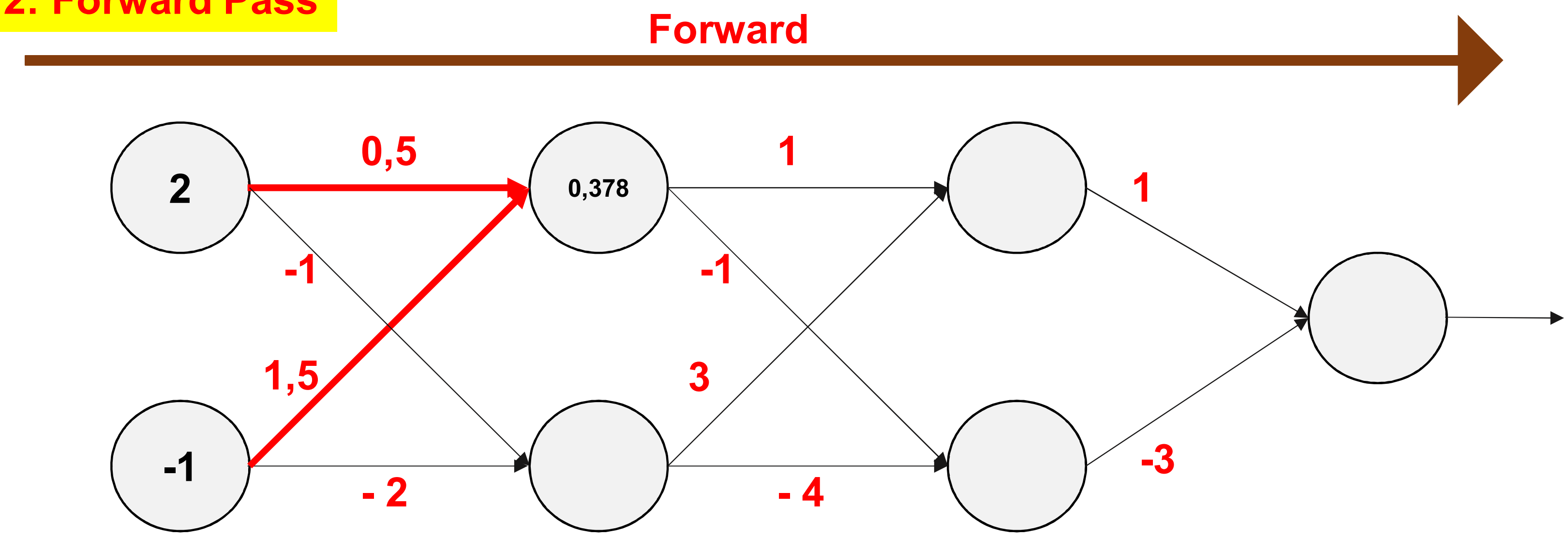


$$? = \boldsymbol{logistic}\left(\mathbf{0,5} * \mathbf{2} + \mathbf{1,5} * (-\mathbf{1})\right) = \boldsymbol{logistic}\,(-\mathbf{0,5}) = \mathbf{0,378}$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1 **Logistic function =** $\frac{1}{1+e^{-x}}$

Step 2: Forward Pass



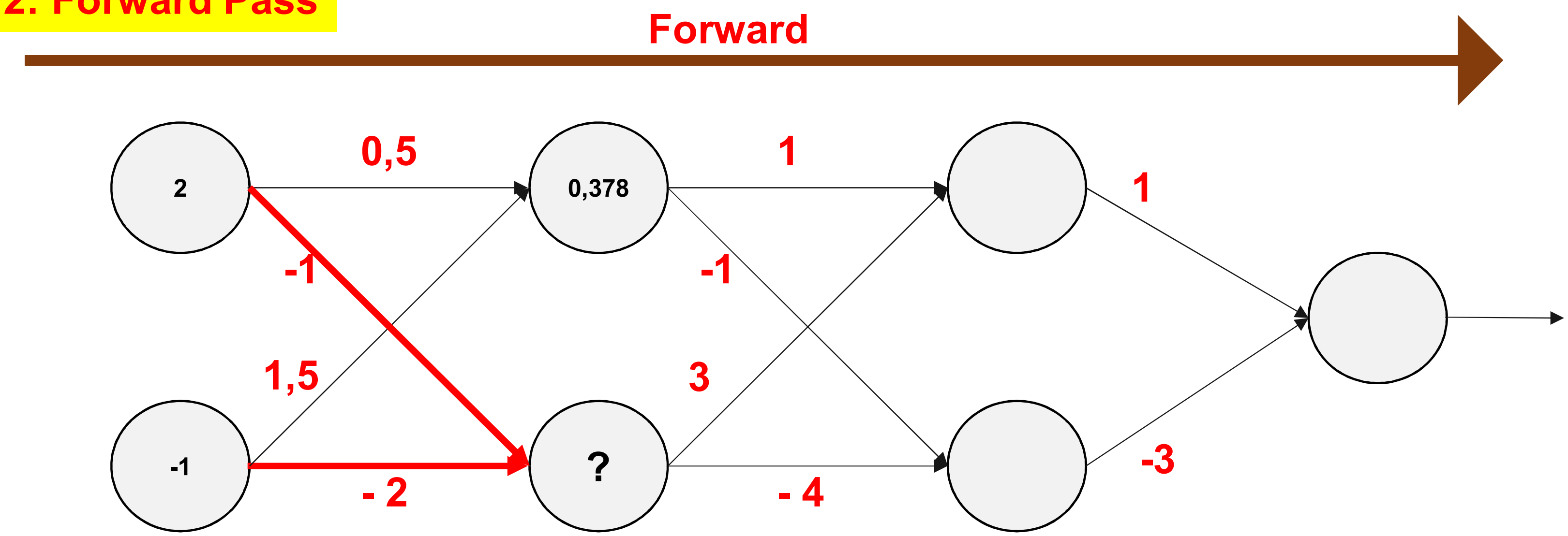
$$? = logistic\,(0{,}5 * 2 + 1{,}5 * (-1)) = logistic\,(-0{,}5) = 0{,}378$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

Step 2: Forward Pass

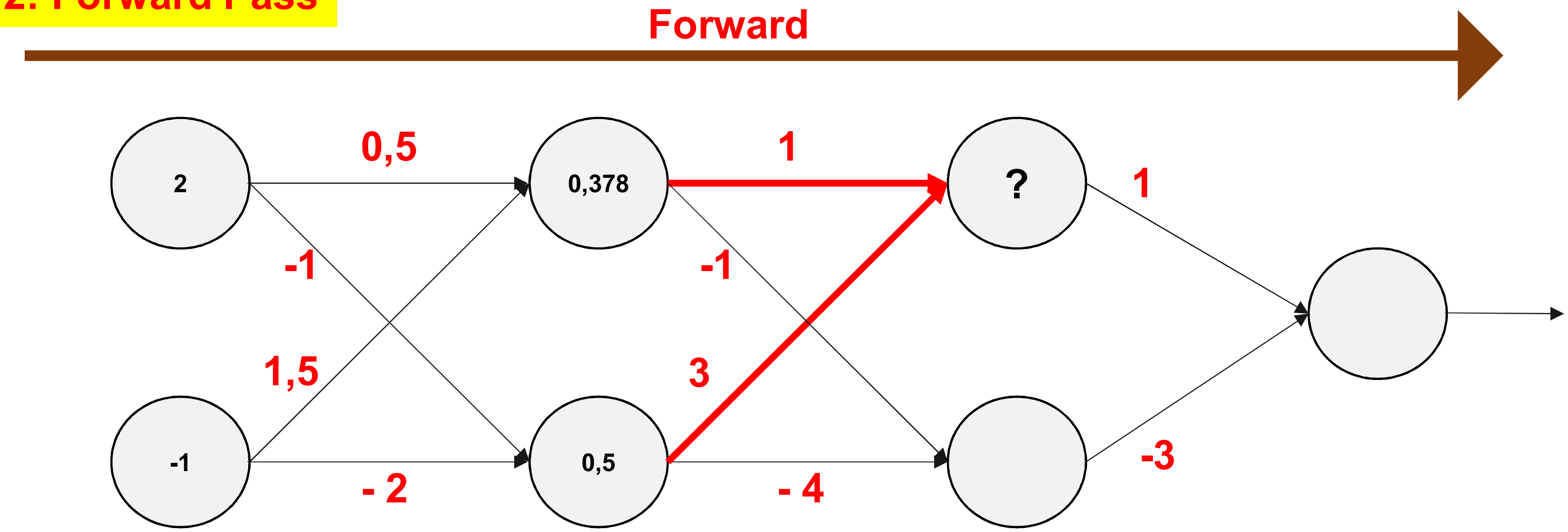


$$? = logistic\left((-1) * 2 + (-2) * (-1)\right) = logistic\ (0) = 0{,}5$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1 **Logistic function =** $\frac{1}{1+e^{-x}}$

Step 2: Forward Pass

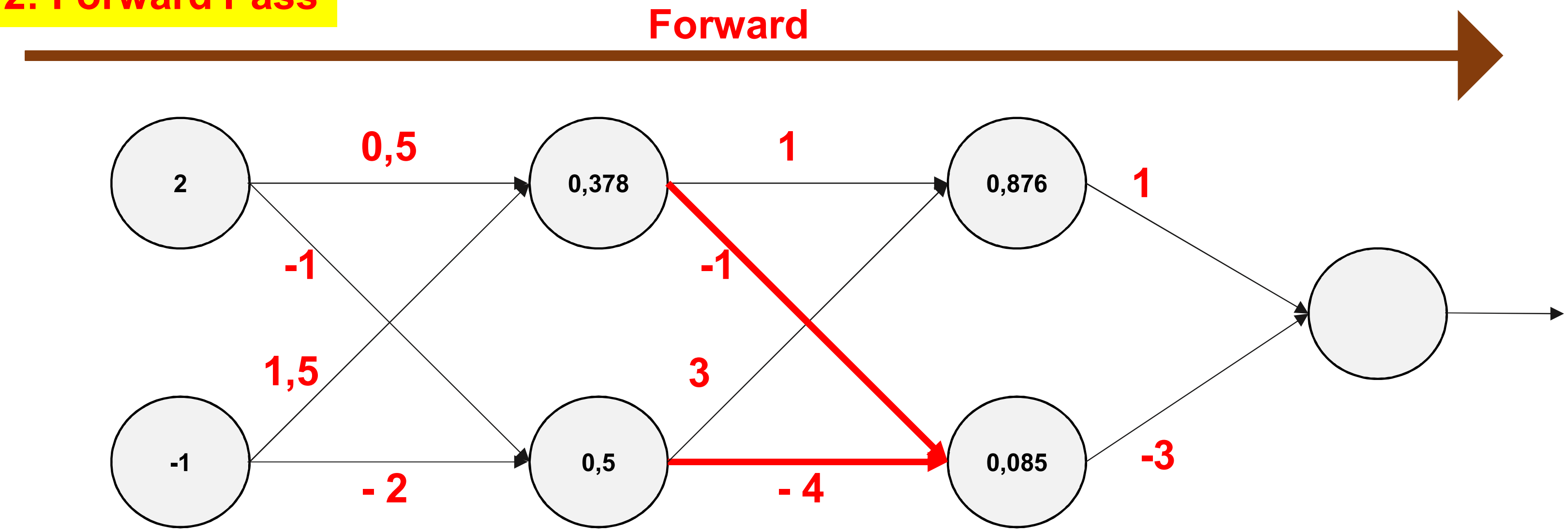


$$? = logistic\ (1 * 0{,}378 + 3\ * 0{,}5) = logistic\ (1{,}878) = 0{,}876$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1 **Logistic function =** $\frac{1}{1+e^{-x}}$

Step 2: Forward Pass

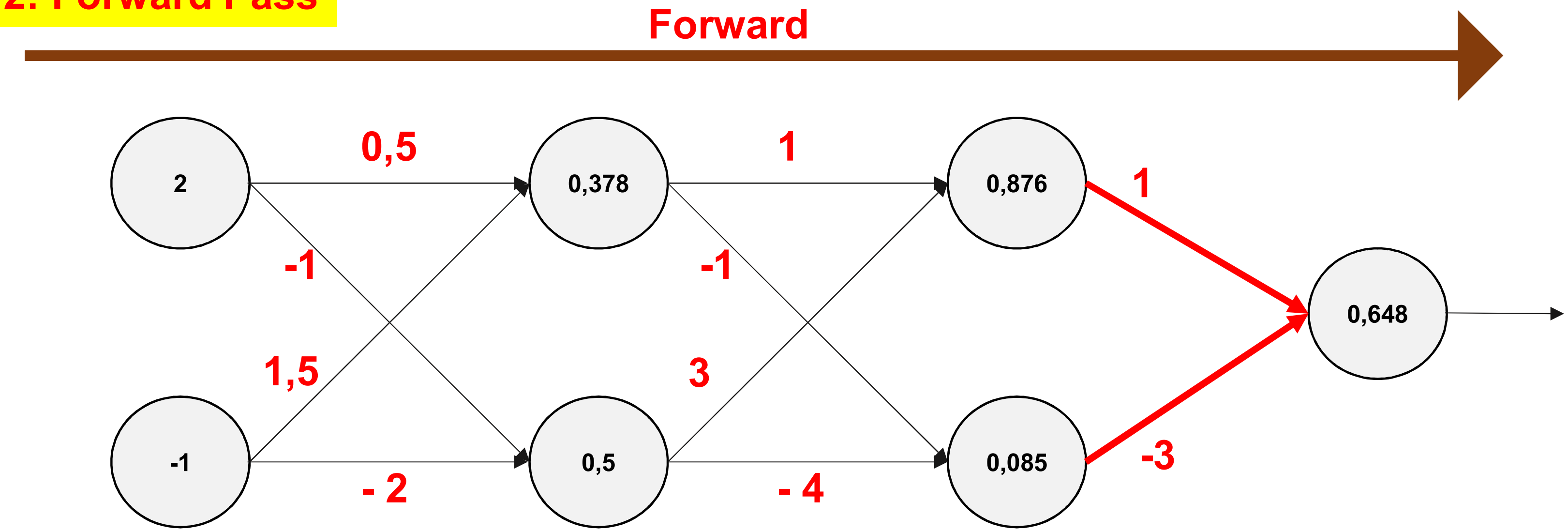


$$? = logistic\ ((-1) * 0{,}378 + (-4)\ * 0{,}5) = 0{,}085$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1 **Logistic function =** $\frac{1}{1+e^{-x}}$

Step 2: Forward Pass



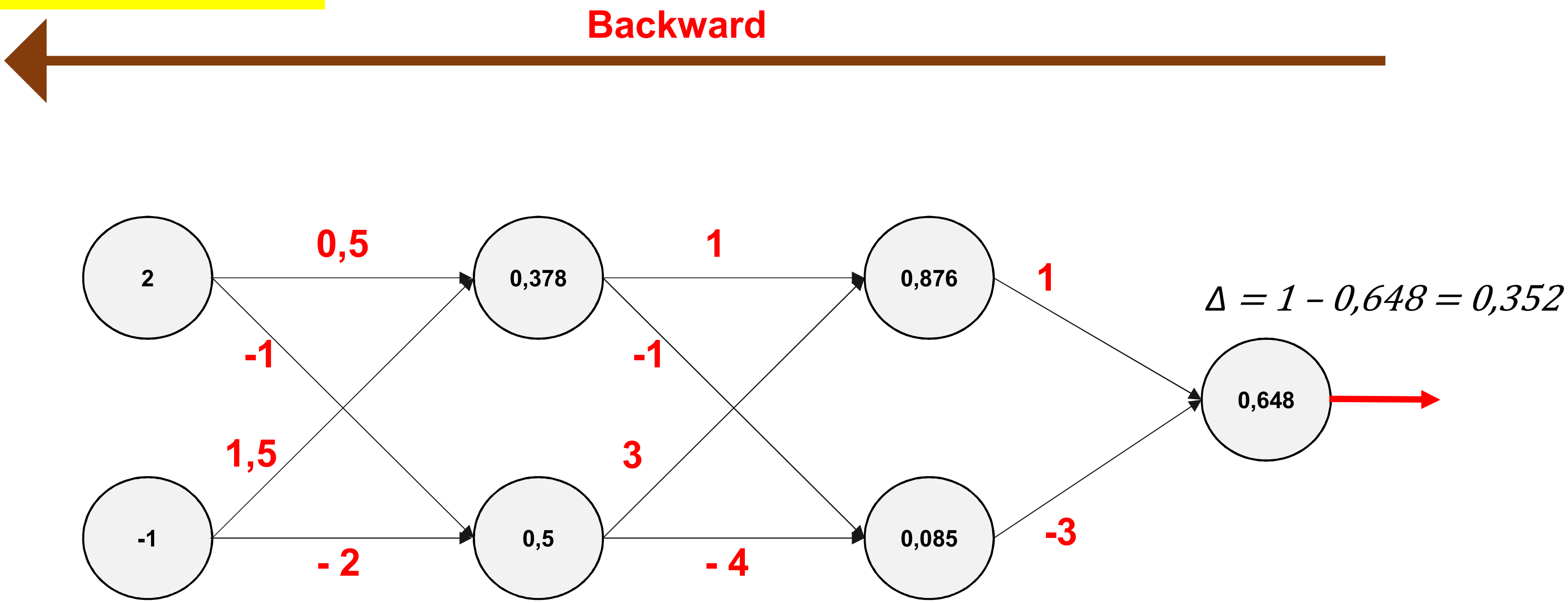
$$? = \ logistic\ (1 * 0{,}876 + (-3)\ * 0{,}085) = 0{,}648$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

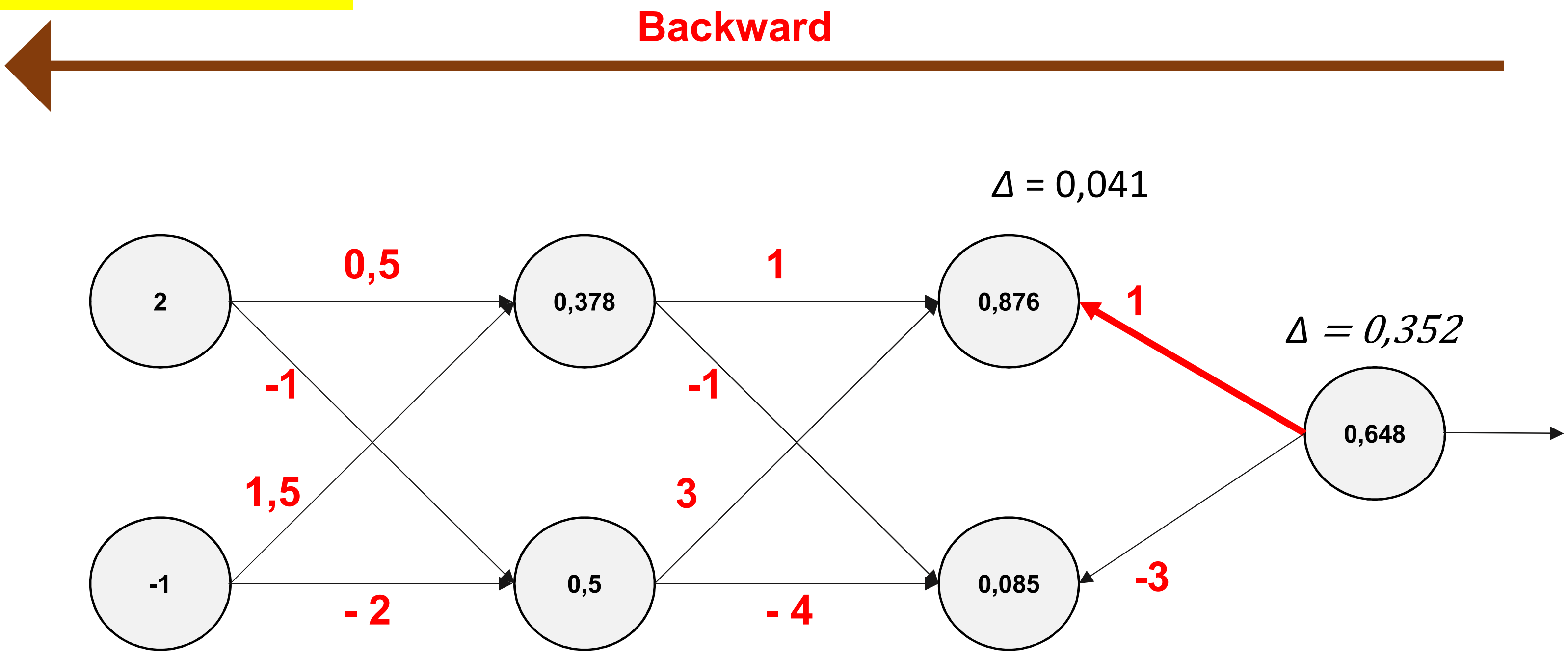
Step 3: Backward Pass



4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1 **Logistic function =** $\frac{1}{1+e^{-x}}$

Step 3: Backward Pass



$\Delta = 0{,}876 * (1 - 0{,}876) * (1 * 0{,}352) = 0{,}041$

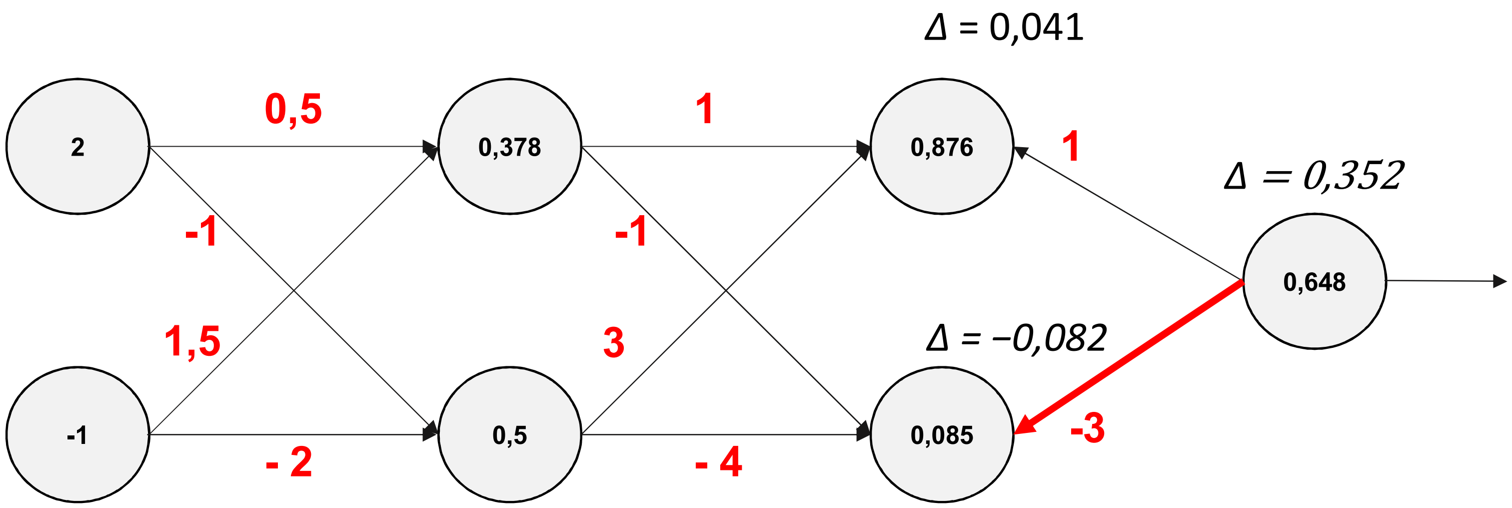
4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

Step 3: Backward Pass

Backward



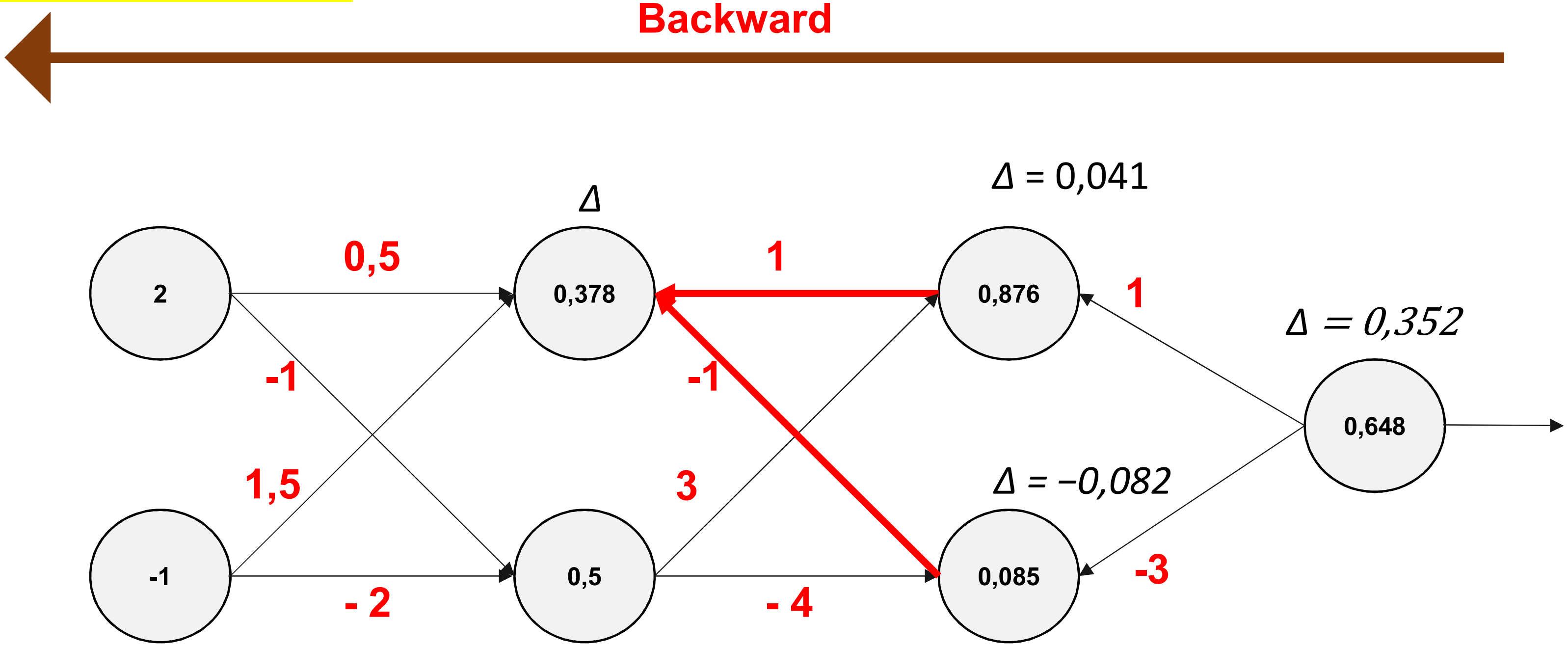
$$\Delta = 0{,}085 * (1 - 0{,}085) * ((-3) * 0{,}352) = -0{,}082$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

Step 3: Backward Pass

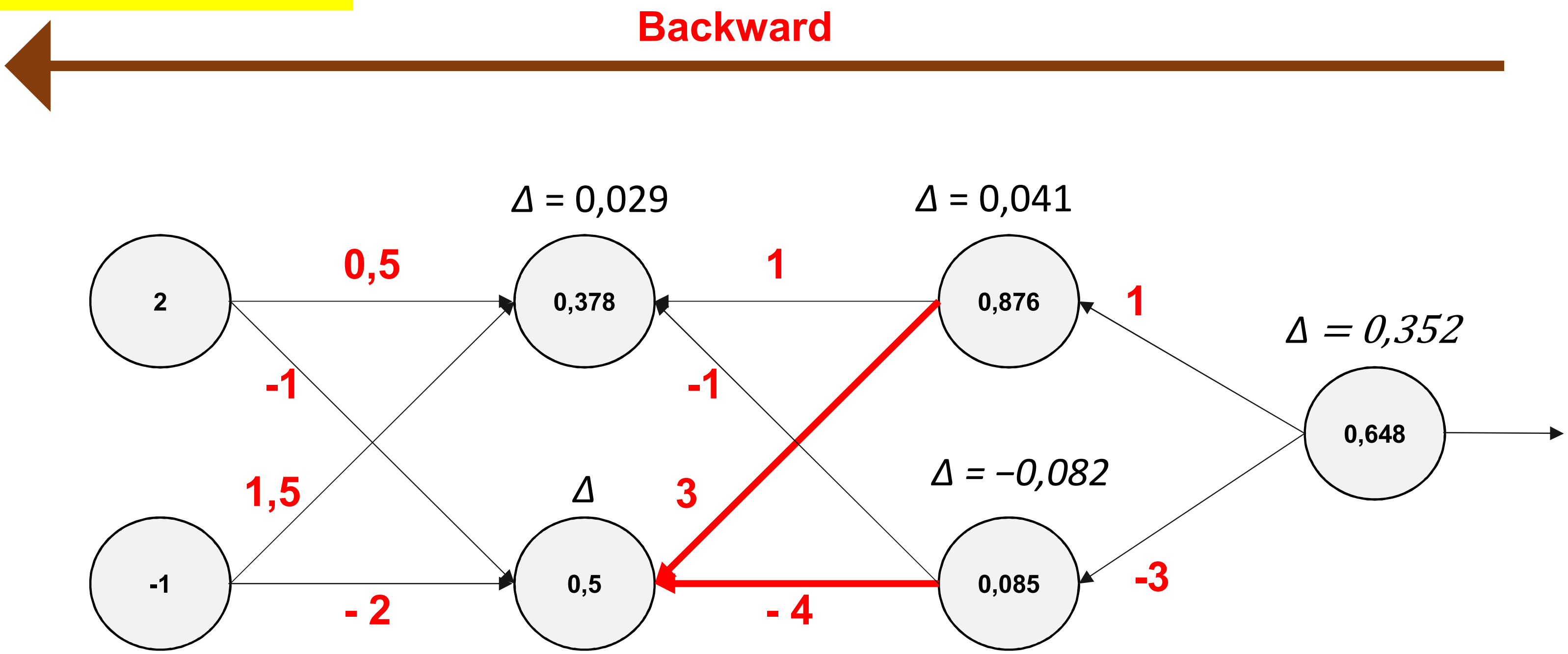


$$\Delta = 0{,}378 * (1 - 0{,}378) * [\, 1 * 0{,}041 + (-1) * (-0{,}082) \,] = 0{,}029$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1 **Logistic function =** $\frac{1}{1+e^{-x}}$

Step 3: Backward Pass

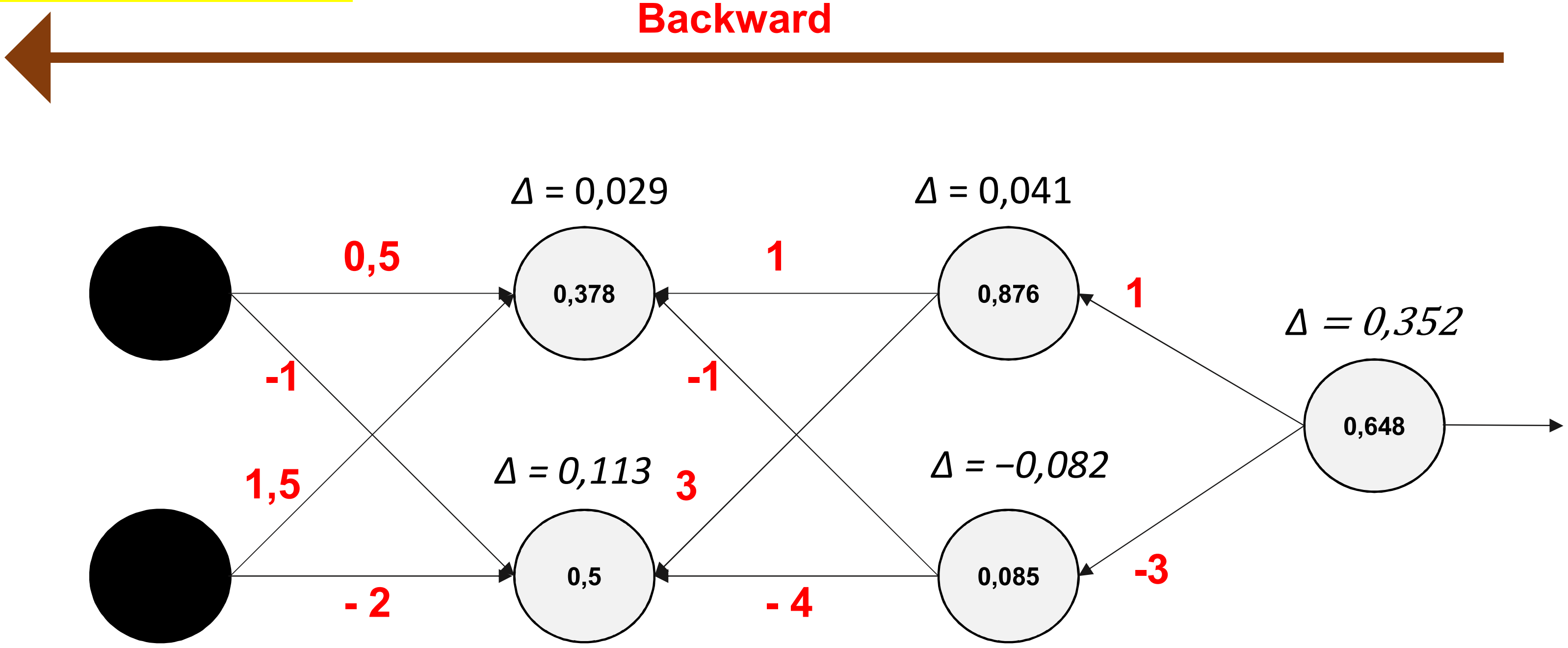


$$\Delta = 0{,}5 * (1 - 0{,}5) * [\, 3 * 0{,}041 + (-4) * (-0{,}082) \,] = 0{,}113$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1 **Logistic function =** $\frac{1}{1+e^{-x}}$

Step 3: Backward Pass



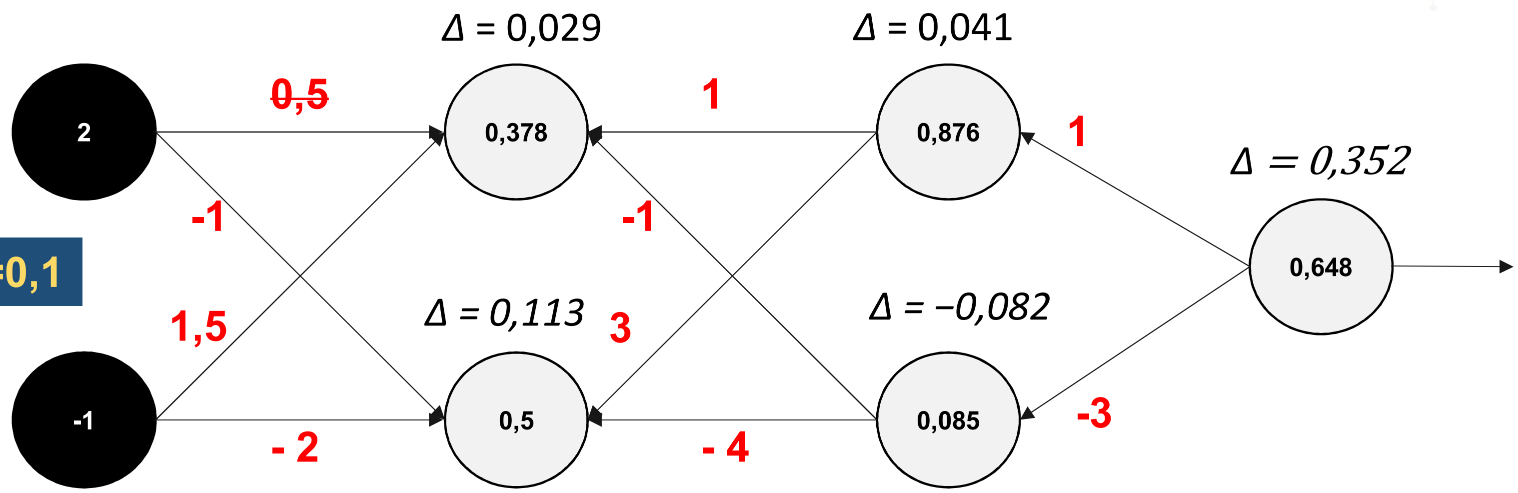
$$\Delta = 0{,}5 * (1 - 0{,}5) * [\, 3 * 0{,}041 + (-4) * (-0{,}082)\,] = 0{,}113$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1   **Logistic function =** $\frac{1}{1+e^{-x}}$

Step 4: Weights' Update

Learning Rate α=0,1



0,5 -> weight old value + α * neuron value * Delta of the next neuron

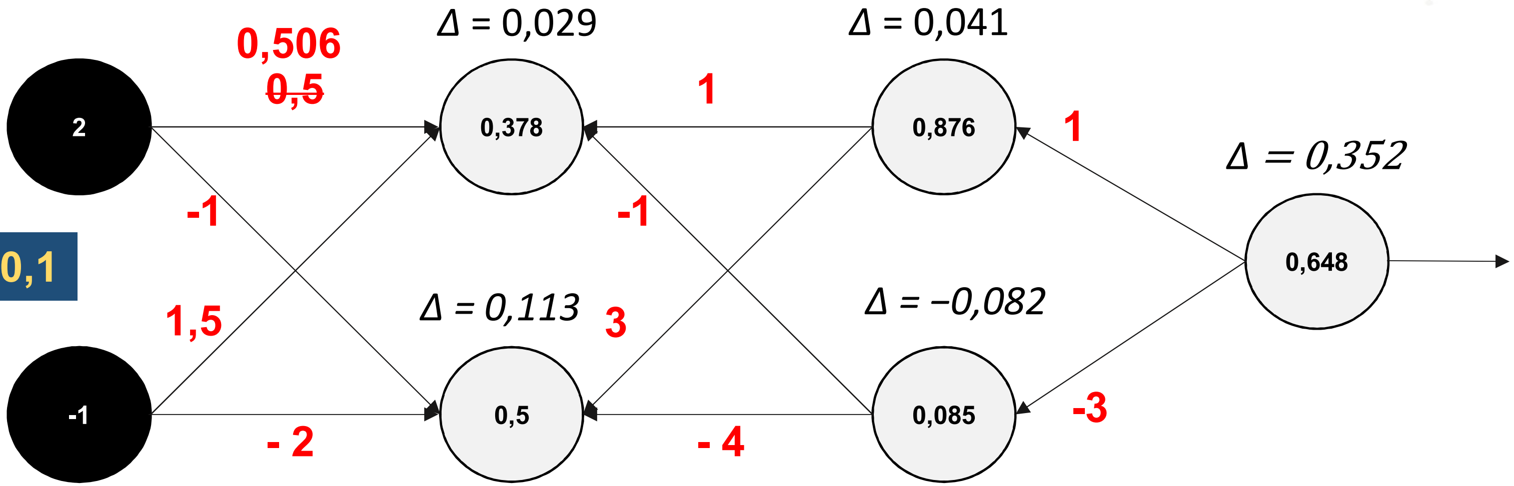
$$0{,}5 -> 0{,}5 + 0{,}1 * 2 * \mathbf{0{,}029} = \mathbf{0{,}506}$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1 **Logistic function =** $\frac{1}{1+e^{-x}}$

Step 4: Weights' Update

Learning Rate α=0,1



0,5 -> weight old value + α * neuron value * Delta of the next neuron

$$0{,}5 \rightarrow 0{,}5 + 0{,}1 * 2 * \mathbf{0{,}029} = \mathbf{0{,}506}$$

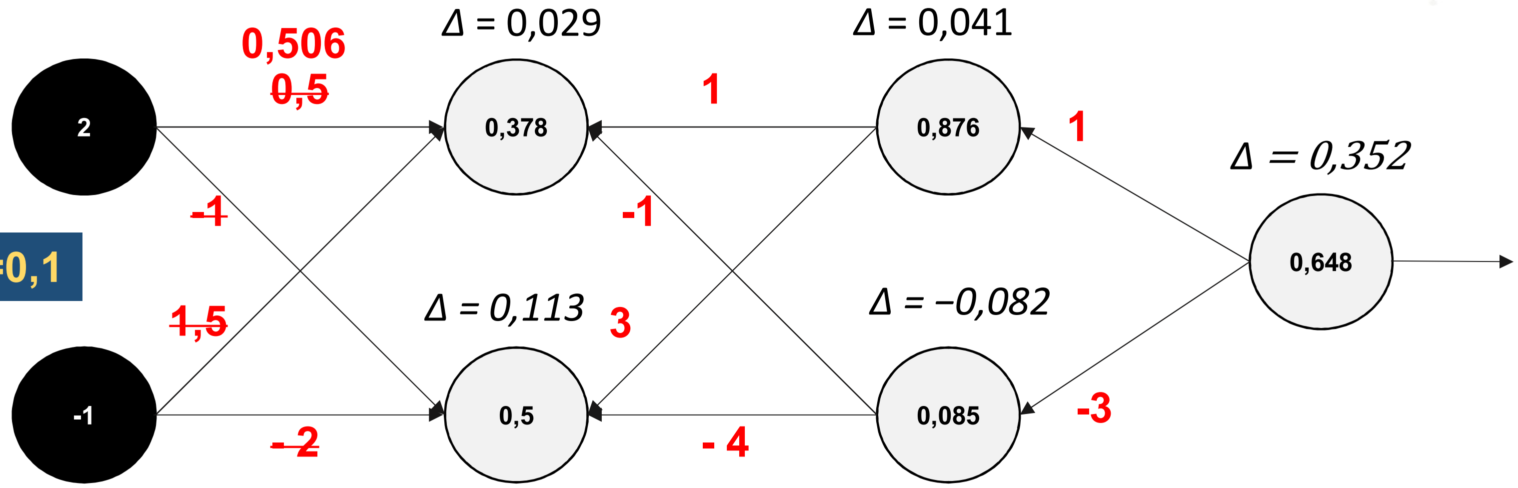
4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

Step 4: Weights' Update

Learning Rate α=0,1



$$0{,}5 \rightarrow 0{,}5 + 0{,}1 * 2 * \mathbf{0,029} = \mathbf{0,506}$$

$$-1 \rightarrow -1 + 0{,}1 * 2 * \mathbf{0,113} = \mathbf{-0,977}$$

$$1{,}5 \rightarrow 1{,}5 + 0{,}1 * (-1) * \mathbf{0,029} = \mathbf{1,497}$$

$$-2 \rightarrow -2 + 0{,}1 * (-1) * \mathbf{0,113} = \mathbf{-2,011}$$

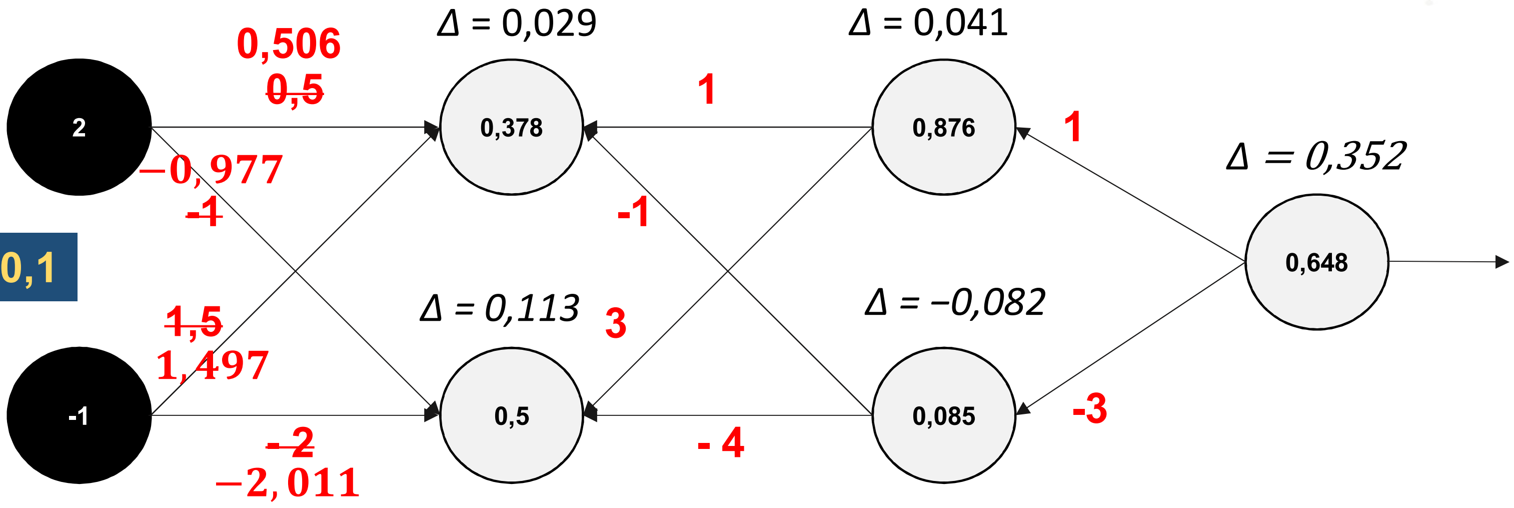
4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

Step 4: Weights' Update

Learning Rate α=0,1



$$1 \rightarrow 1 + 0{,}1 * 0{,}378 * 0{,}041 = 1{,}002$$

$$-1 \rightarrow -1 + 0{,}1 * 0{,}378 * (-0{,}082) = -1{,}003$$

$$3 \rightarrow 3 + 0{,}1 * 0{,}5 * 0{,}041 = 3{,}002$$

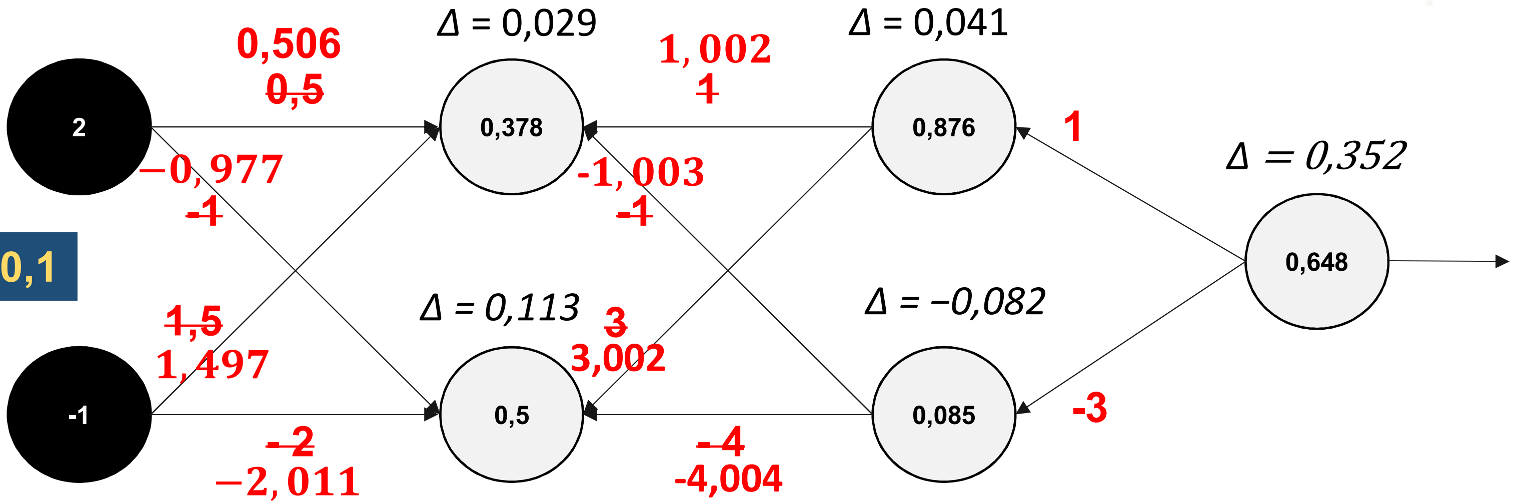
$$-4 \rightarrow -4 + 0{,}1 * 0{,}5 * (-0{,}082) == -4{,}004$$

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1 **Logistic function =** $\frac{1}{1+e^{-x}}$

Step 4: Weights' Update

Learning Rate α=0,1



$$1 -> 1 + 0{,}1 * 0{,}876 * \mathbf{0,352} = \mathbf{1,031}$$

$$-3 -> -3 + 0{,}1 * 0{,}085 * \mathbf{0,352} = \mathbf{-2,997}$$

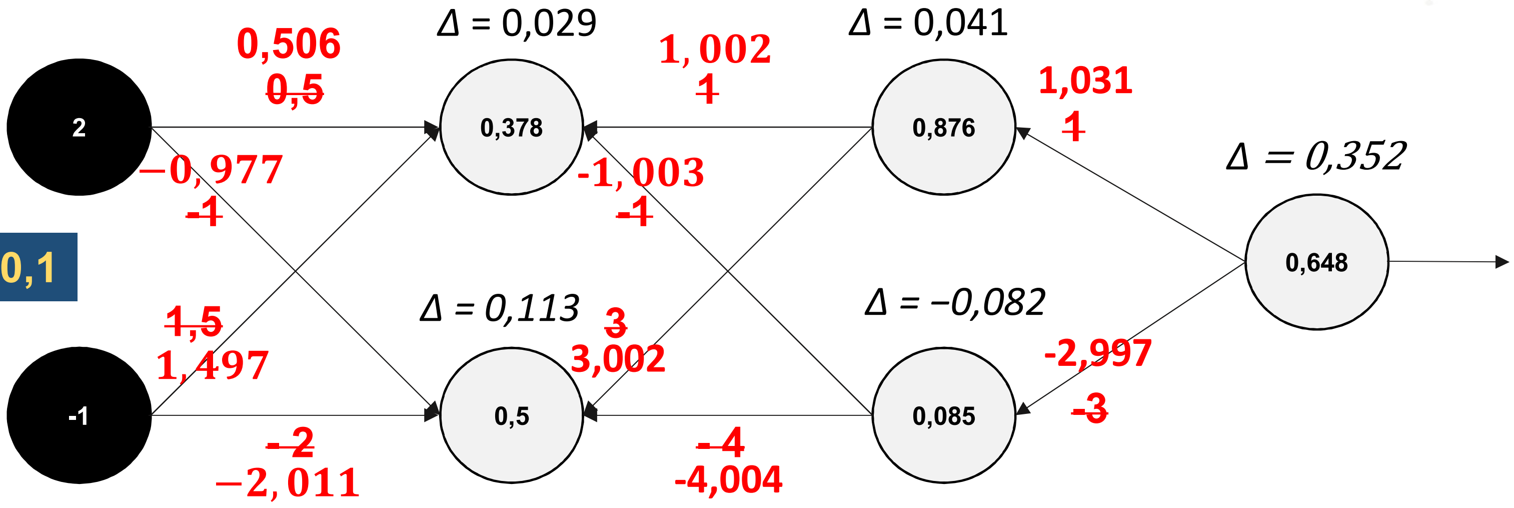
4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

Step 4: Weights' Update

Learning Rate α=0,1



Step 5: Repeat Step 1 to Step 4 for all train vectors

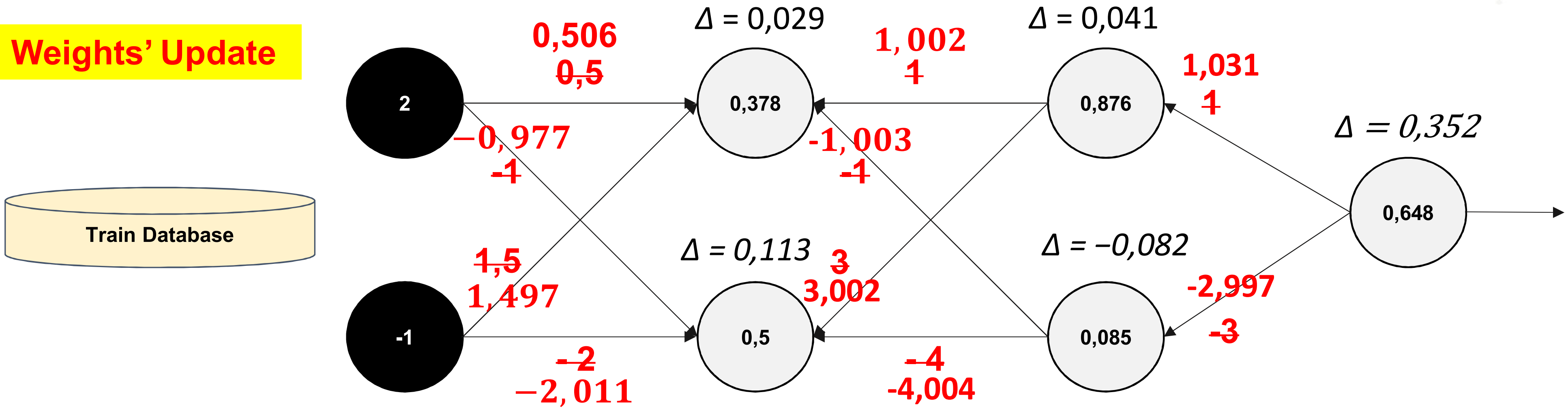
Train Database

4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

Step 4: Weights' Update



Step 5: Repeat Step 1 to Step 4 for all train vectors

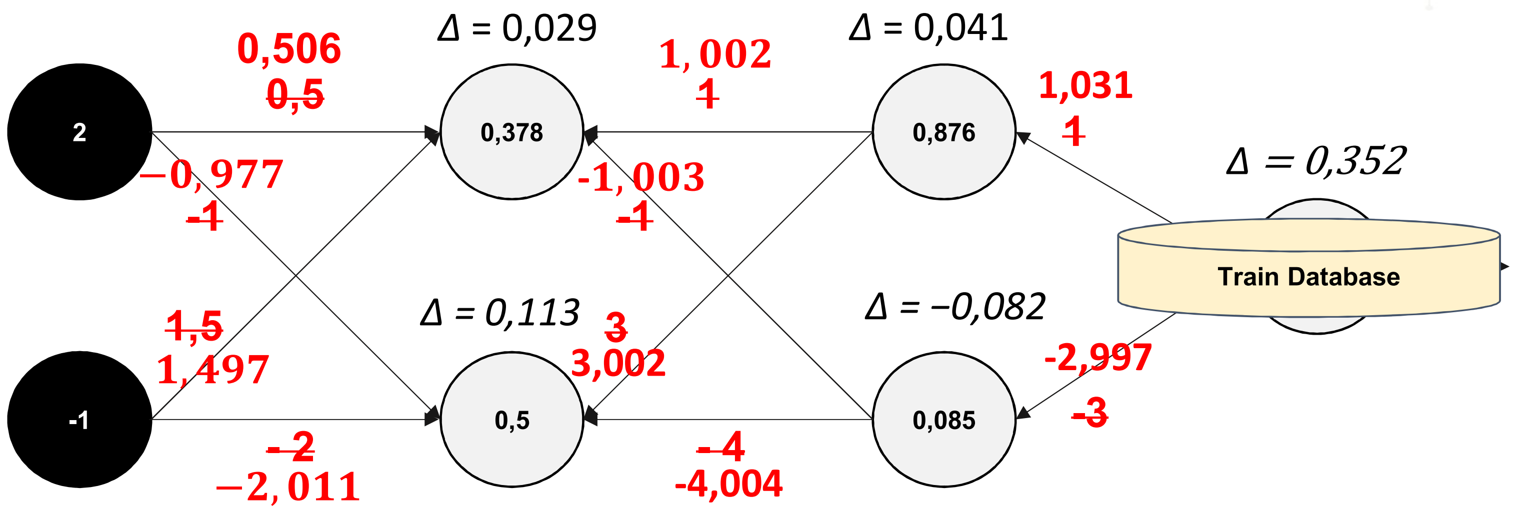
4. Neural Network: Theory and Application

Train Vector = [2 , -1] ; Train Label = 1

Logistic function = $\frac{1}{1+e^{-x}}$

Step 4: Weights' Update

Epoch



Step 5: Repeat Step 1 to Step 4 for all train vectors

5. Naïve Bayes: Theory and Application

Activity

Color	Type	Origin	Stolen
Red	Sport	Domicile	Yes
Red	Sport	Domicile	No
Red	Sport	Domicile	Yes
Yellow	Sport	Domicile	No
Yellow	Sport	Importation	Yes
Yellow	Classic	Importation	No
Yellow	Classic	Importation	Yes
Yellow	Classic	Domicile	No
Red	Classic	Importation	No
Red	Sport	Importation	Yes

$$P(Yes) = \frac{5}{10}$$

$$P(No) = \frac{5}{10}$$

Learning

Color

$$P(Red/Yes) = \frac{3}{5} \qquad P(Yellow/Yes) = \frac{2}{5}$$

$$P(Red/No) = \frac{2}{5} \qquad P(Yellow/No) = \frac{3}{5}$$

Type

$$P(Sport/Yes) = \frac{4}{5} \qquad P(Classic/Yes) = \frac{1}{5}$$

$$P(Sport/No) = \frac{2}{5} \qquad P(Classic/No) = \frac{3}{5}$$

Origin

$$P(Domicile/Yes) = \frac{2}{5} \qquad P(Importation/Yes) = \frac{3}{5}$$

$$P(Domicile/No) = \frac{3}{5} \qquad P(Importation/No) = \frac{2}{5}$$

5. Naïve Bayes: Theory and Application

Sample X= <Red, Classic, Domicile>

$$P(X, Yes) = P(Red/Yes) \text{ x } P(Classic/Yes) \text{ x } P(Domicile/Yes) \text{ x } P(Yes)$$
$$= \frac{3}{5} * \frac{1}{5} * \frac{2}{5} * \frac{5}{10}$$

$$P(X, No) = P(Red/No) \text{ x } P(Classic/No) \text{ x } P(Domicile/No) \text{ x } P(No)$$
$$= \frac{2}{5} * \frac{3}{5} * \frac{3}{5} * \frac{5}{10}$$

Testing

$$P(No) = \frac{5}{10} \qquad P(Yes) = \frac{5}{10}$$

Color

$$P(Red/Yes) = \frac{3}{5} \qquad P(Yellow/Yes) = \frac{2}{5}$$
$$P(Red/No) = \frac{2}{5} \qquad P(Yellow/No) = \frac{3}{5}$$

Type

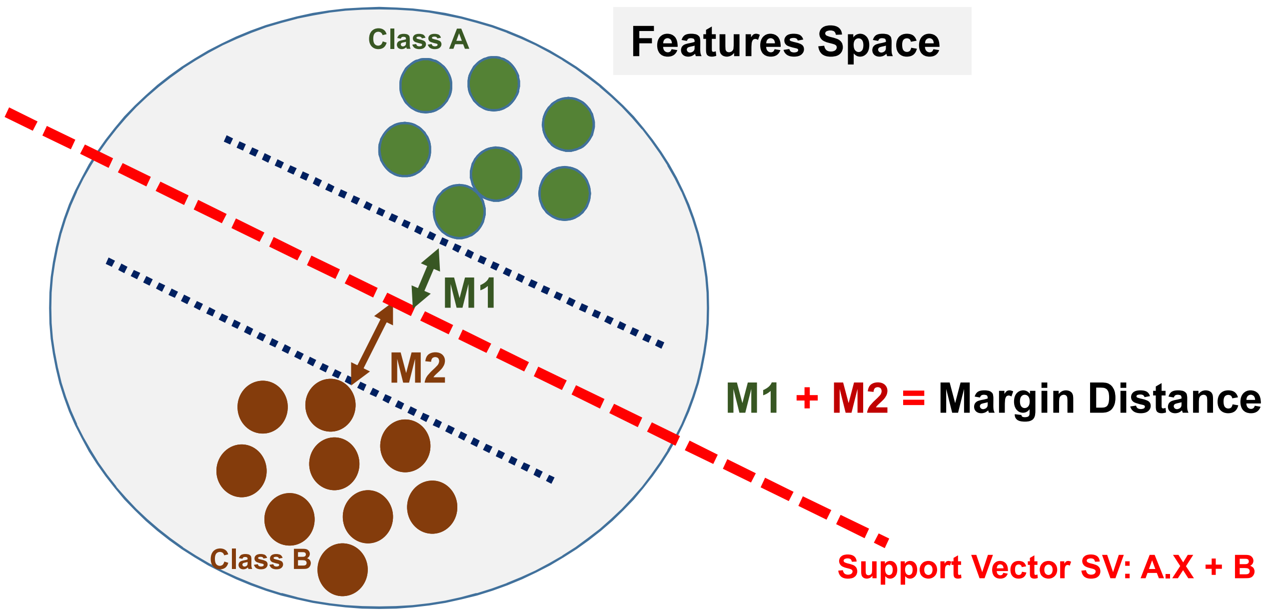
$$P(Sport/Yes) = \frac{4}{5} \qquad P(Classic/Yes) = \frac{1}{5}$$
$$P(Sport/No) = \frac{2}{5} \qquad P(Classic/No) = \frac{3}{5}$$

Origin

$$P(Domicile/Yes) = \frac{2}{5} \qquad P(Importation/Yes) = \frac{3}{5}$$
$$P(Domicile/No) = \frac{3}{5} \qquad P(Importation/No) = \frac{2}{5}$$

6. Support Vector Machines: Theory and Application

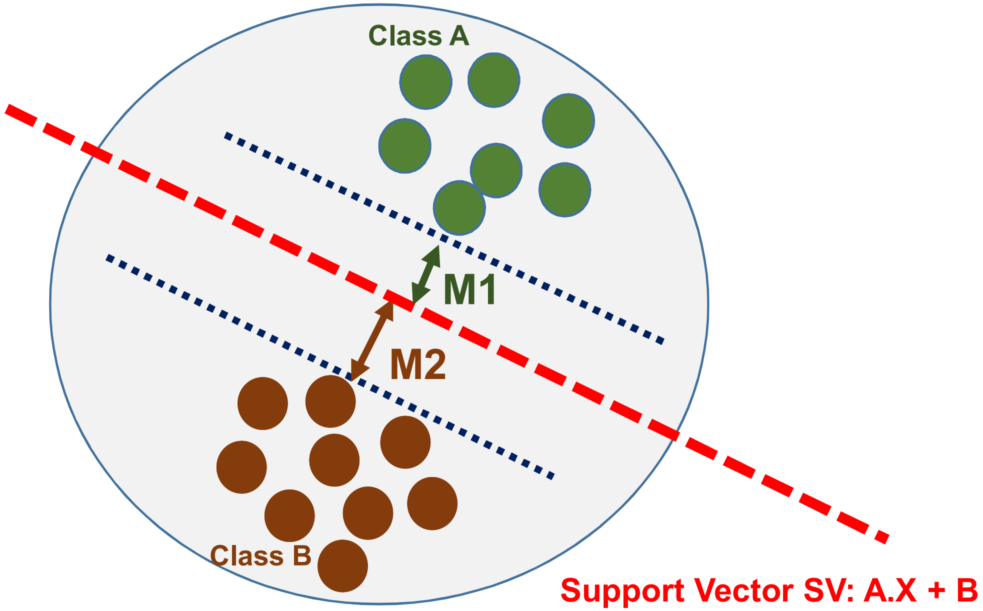
Basic Idea: Find the appropriate Support Vector which maximize Margin Distance



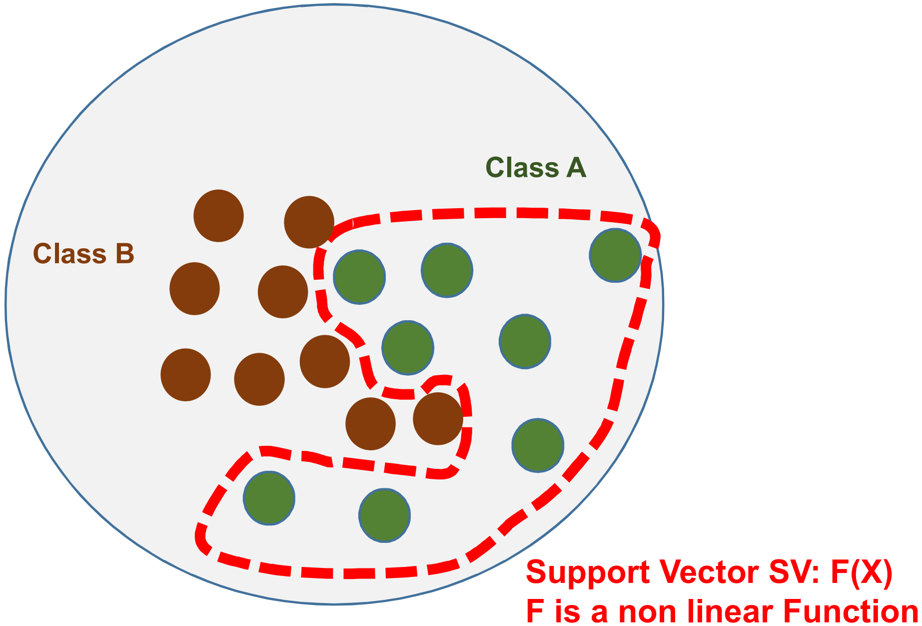
6. Support Vector Machines: Theory and Application

Basic Idea: Find the appropriate Support Vector which maximize Margin Distance

Case 1: Linear Separation



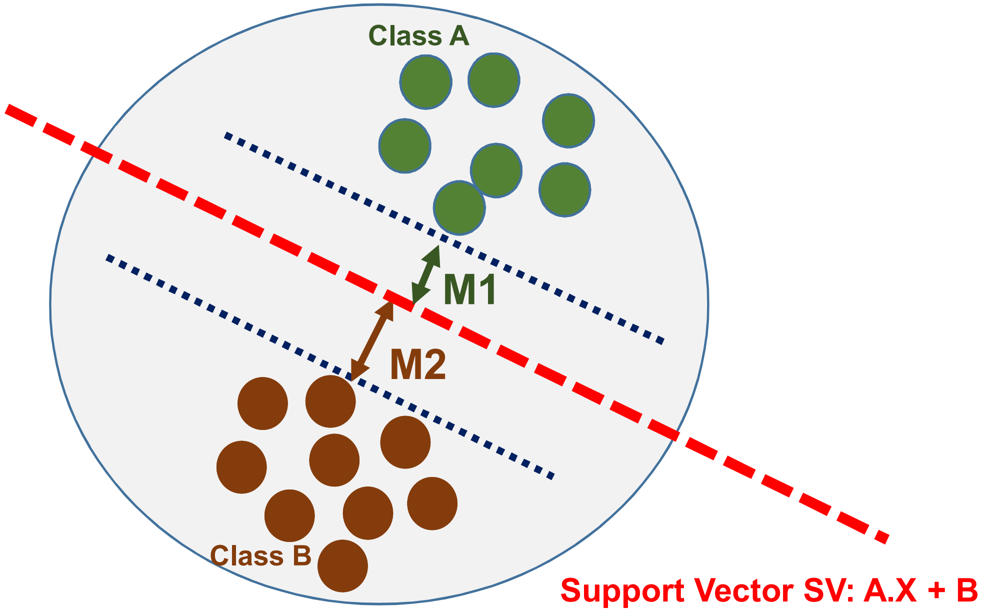
Case 2: Non Linear Separation



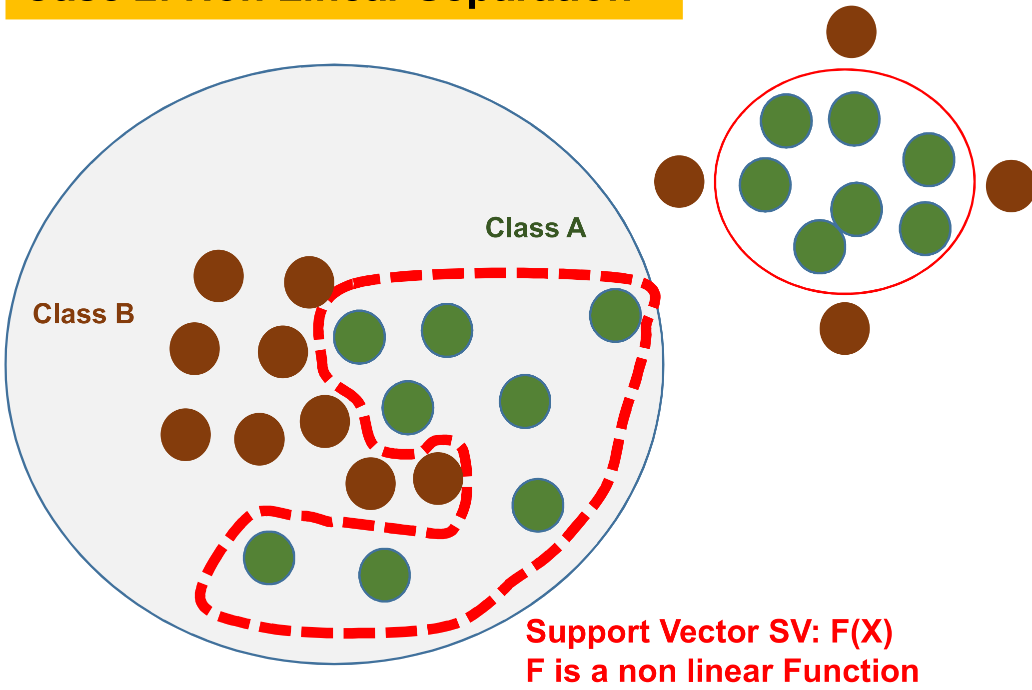
6. Support Vector Machines: Theory and Application

Basic Idea: Find the appropriate Support Vector which maximize Margin Distance

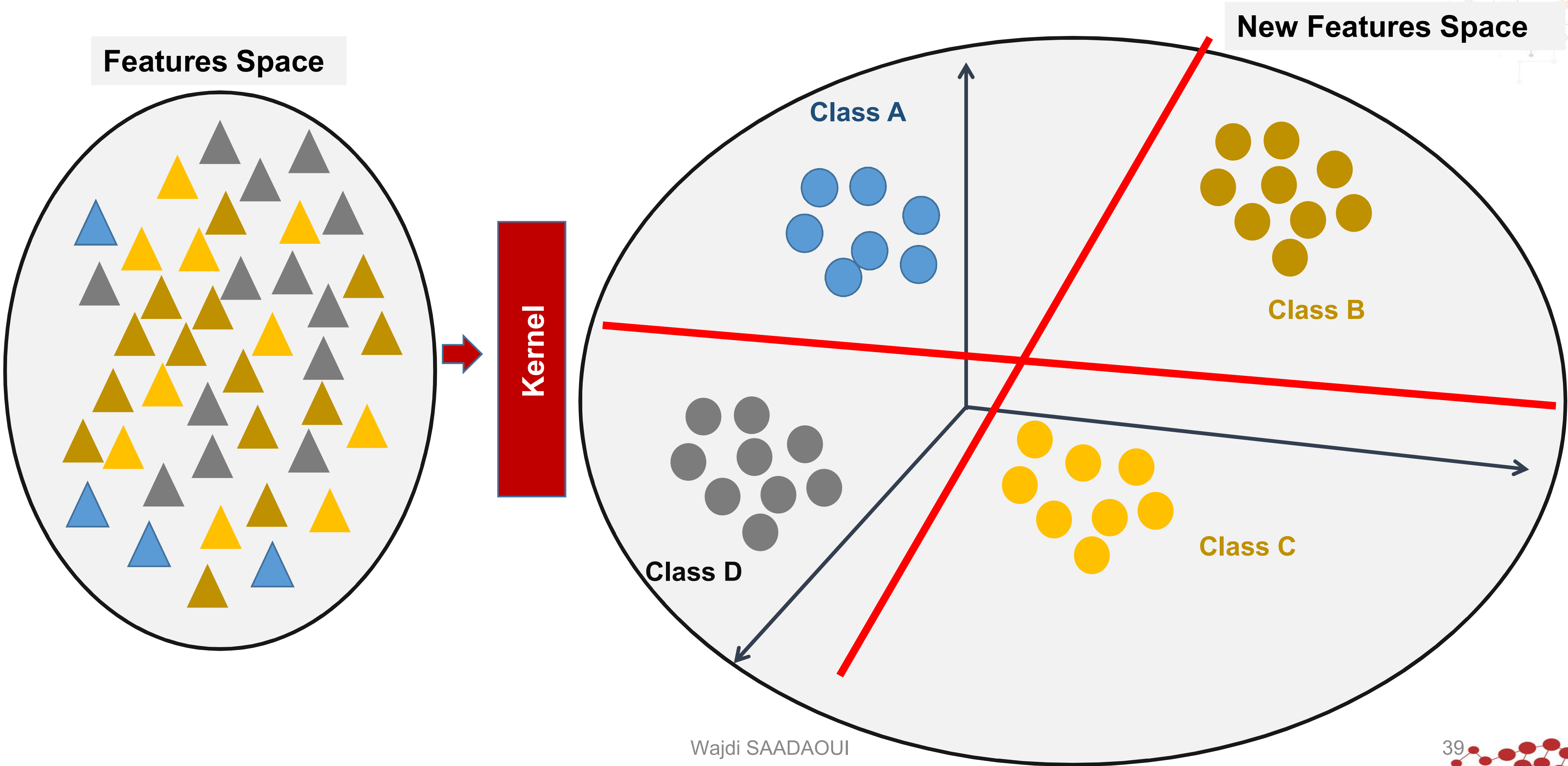
Case 1: Linear Separation



Case 2: Non Linear Separation



6. Support Vector Machines: Theory and Application



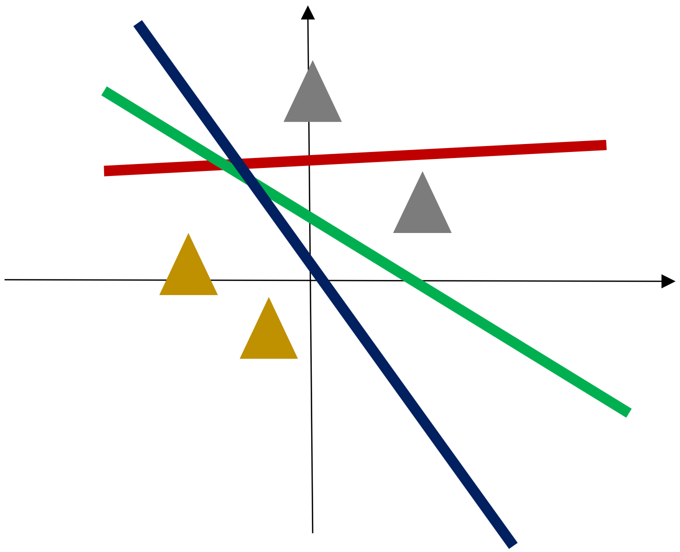
6. Support Vector Machines: Theory and Application

Kernel

	Kernel function	Expression
1	Liner kernel function	$K(x_i, x_j) = x_i \cdot x_j$
4	Polynomial kernel function	$K(x_i, x_j) = \left(x_i \cdot x_j + 1\right)^d$
2	Radial basis function (RBF) kernel function	$K(x_i, x_j) = \exp\left(-\gamma \left\Vert x_i - x_j \right\Vert^2\right)$
3	Sigmoid kernel function	$K\left(x_i, x_j\right) = \tanh\left(b\left(x_i, x_j\right) + c\right)$

6. Support Vector Machines: Theory and Application

Case 1: Linear Separation

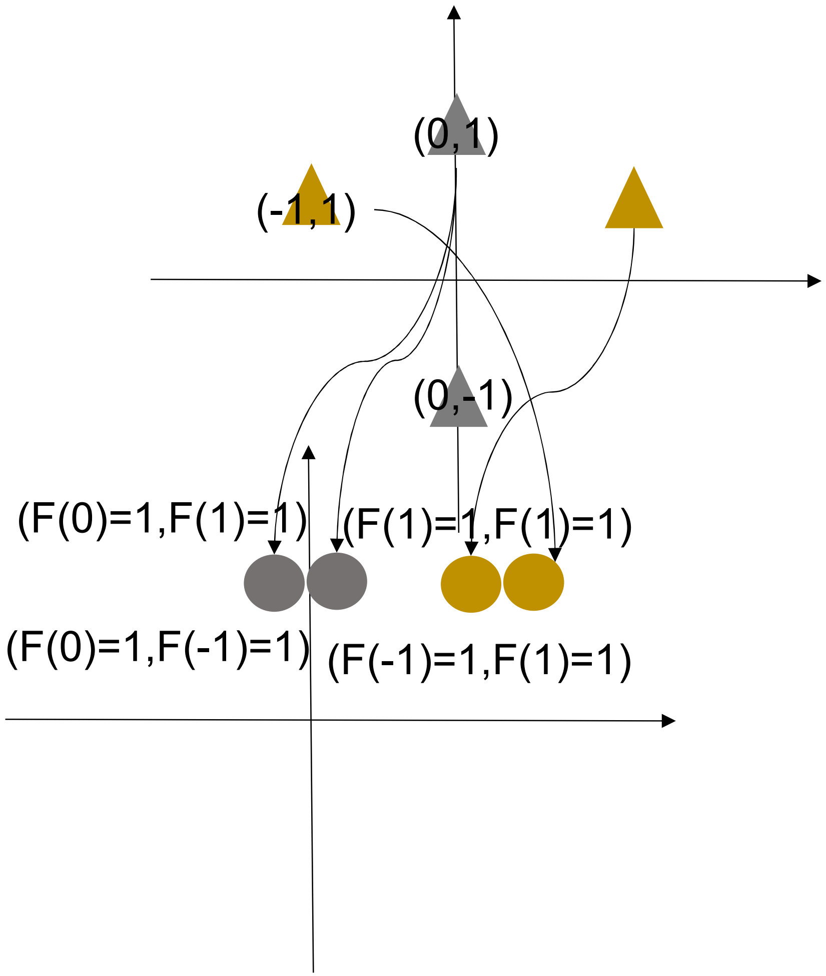


1. K Support Vectors = N-1 where N is the number of samples (K=3)
2. Initialize 3 linear support vectors
   - D1= A1*X+B1
   - D2= A2*X+B2
   - D3= A3*X+B3
3. Compute the accuracy of SVs
   - D1 (75%)
   - D2 (100%)
   - D3 (100%)
4. Thresholding Acc>85%
   - D2 (100%)
   - D3 (100%)
4. Compare Margin Distance
   - M2 (100%)
   - M3 (100%)

- We keep the highest one

6. Support Vector Machines: Theory and Application

Case 2: Non Linear Separation



1. Use a Kernel Function F=x^2
2. Transform Vectors using F function
3. Apply Linear separability
4. K Support Vectors = N-1 where N is the number of samples (K=3)
5. Initialize 3 linear support vectors
   - D1= A1*X+B1
   - D2= A2*X+B2
   - D3= A3*X+B3

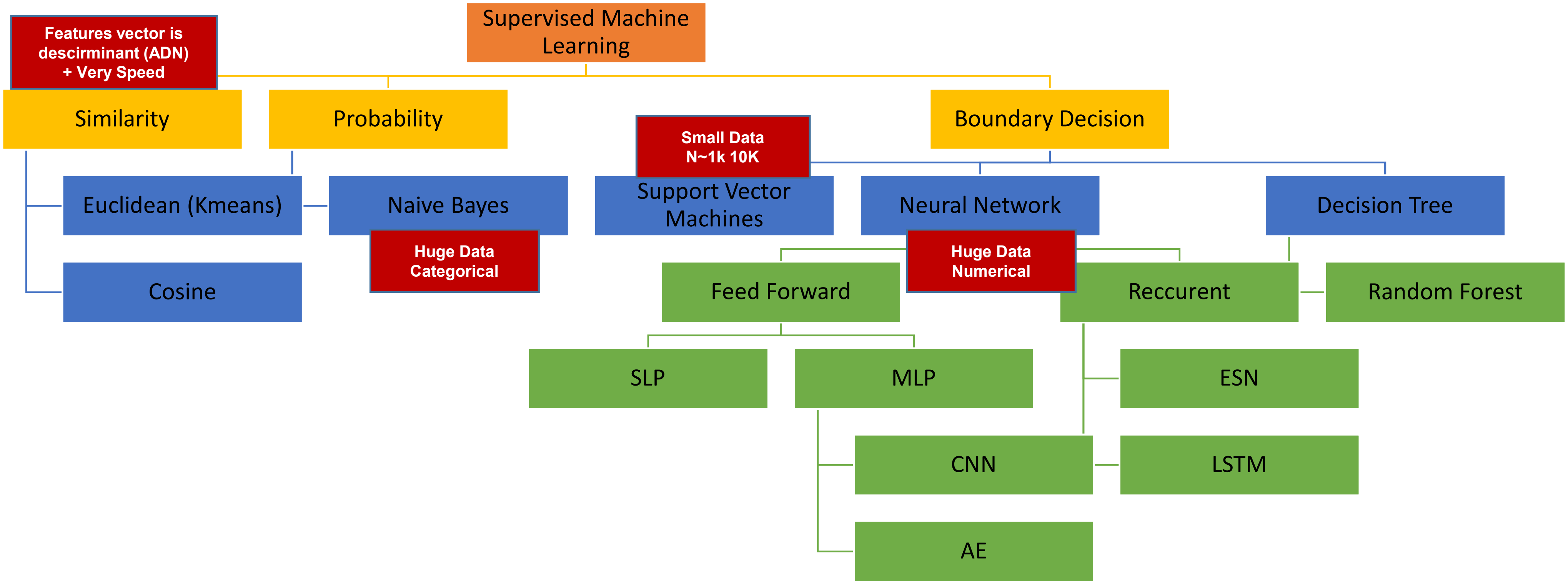
3. Compute the accuracy of SVs
   - D1 (75%)
   - D2 (100%)
   - D3 (100%)

4. Thresholding Acc>85%
   - D2 (100%)
   - D3 (100%)

4. Compare Margin Distance
   - M2 (100%)
   - M3 (100%)

- We keep the highest one

7. How to select the appropriate Machine Learning

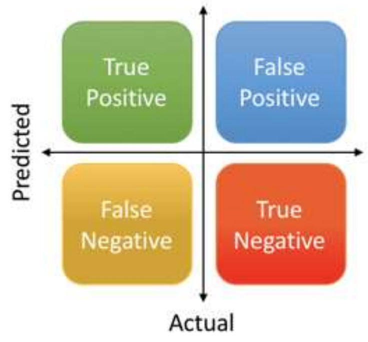


8. How to evaluate the Machine Learning Performance

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\textbf{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$



		Predicted Class	
		Spam	Non-Spam
Actual Class	Spam 65	TP=45	FN=20
	Non-Spam 35	FP=5	TN=30

$$F1-score = 2 * \frac{Recall \; * Precision}{Recall + Precision}$$

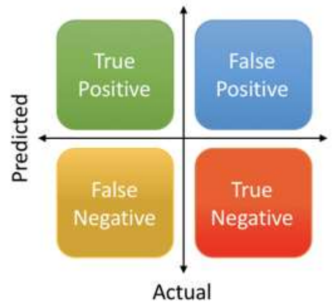
$$g_{mean} = \sqrt{(Recall \; * \; Precision)}$$

8. How to evaluate the Machine Learning Performance

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\textbf{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$



$$F1-score = 2 * \frac{Recall \; * Precision}{Recall + Precision}$$

$$g_{mean} = \sqrt{(Recall \; * \; Precision)}$$

Activity

After training your model, you have 300 samples for test & validation to evaluate your model portioned as follow:

- 150 Class 1
- 150 Class 2

After testing your model well Classify 100 from Class 1 and 140 from Class 2

Accuracy = (100+140)/300 %

	Class 1	**Class 2**
Class 1 (150)	100	50
Class 2 (150)	10	140

Precision = 100/150

Recall = 100/100+10

7. How to select the appropriate Machine Learning

Summary

Machine Learning	Model	Projection	Efficiency
Similarity	Dataset	Similarity Measures	Discriminant Features vector
Naive Bayes	Set of probabilities	Compute Probability of each class	Huge Categorical Data
Support Vector Machines	Set of Support Vectors	Compute marginal distance	Small Numerical data
Neural Network	Weight Matrix	Scalar Product	Huge Numerical Data