Introduction to ANN, DTrees and Clustering

Khaled El-Wazan [1]

Department of Mathematics and Computer Science, Faculty of Science,
Alexandria University

[1] khaled_elwazan@alex-sci.edu.eg

Outlines

1: Artificial Neural Networks

Introduction

Roots of work on neural Networks (NN) are from:

- Neuro-biological studies (more than a century), e.g. how nerves behave when stimulated by different electric currents.
- Psychological studies, e.g. how animal learn, forget and recognize types and tasks.
- Psycho-physical experiments, and
- MaCulloch and Pitts, when they introduced their first mathematical model for single neurons.

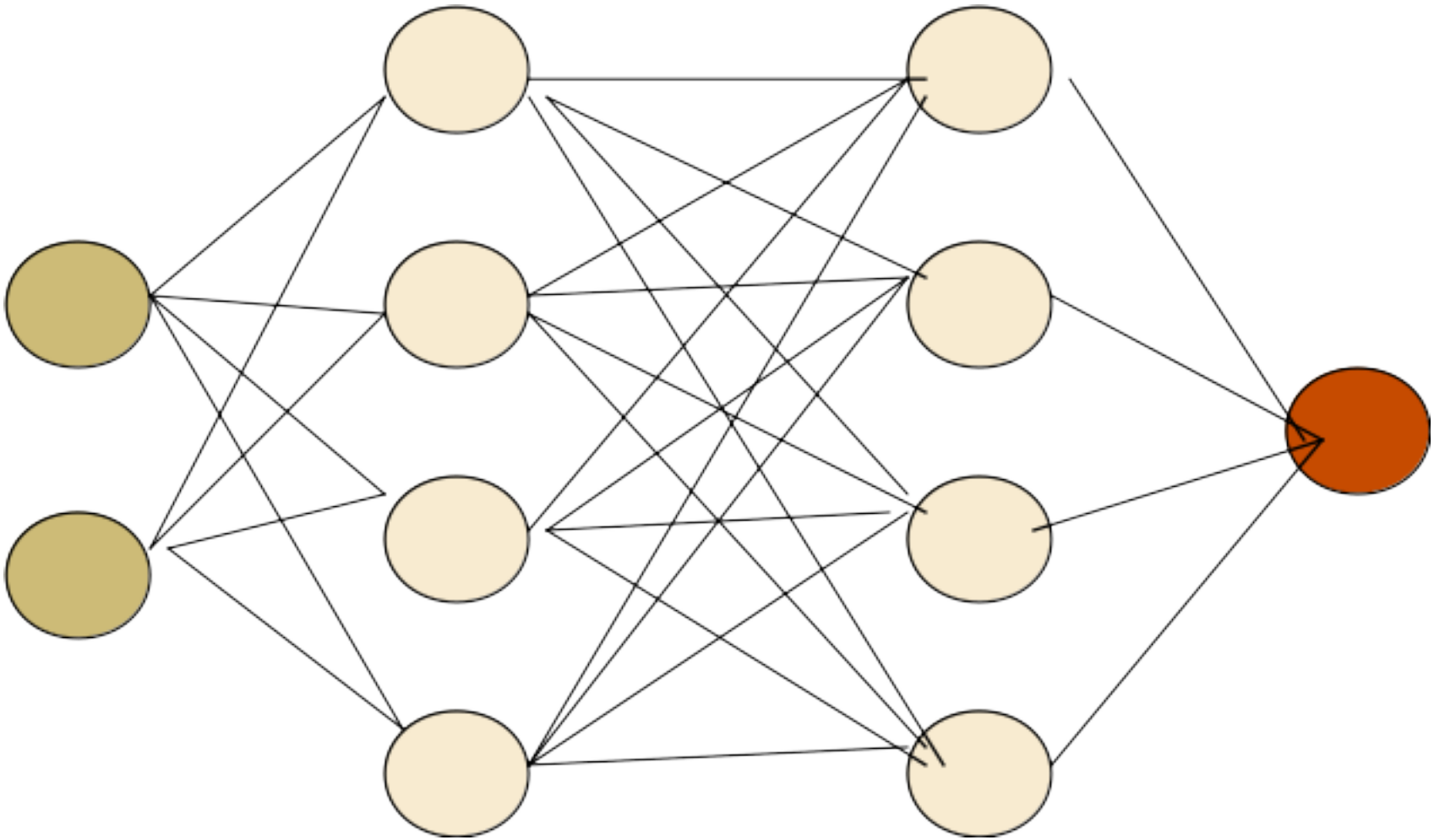
Applications

- Classification
  - Image recognition
  - Speech recognition
  - Diagnostics
  - Fraud detection
- Regression
  - Forecasting and prediction
- Pattern association
  - Retrieve an image from corrupted one
- Clustering
  - Client profiling
  - Disease subtypes

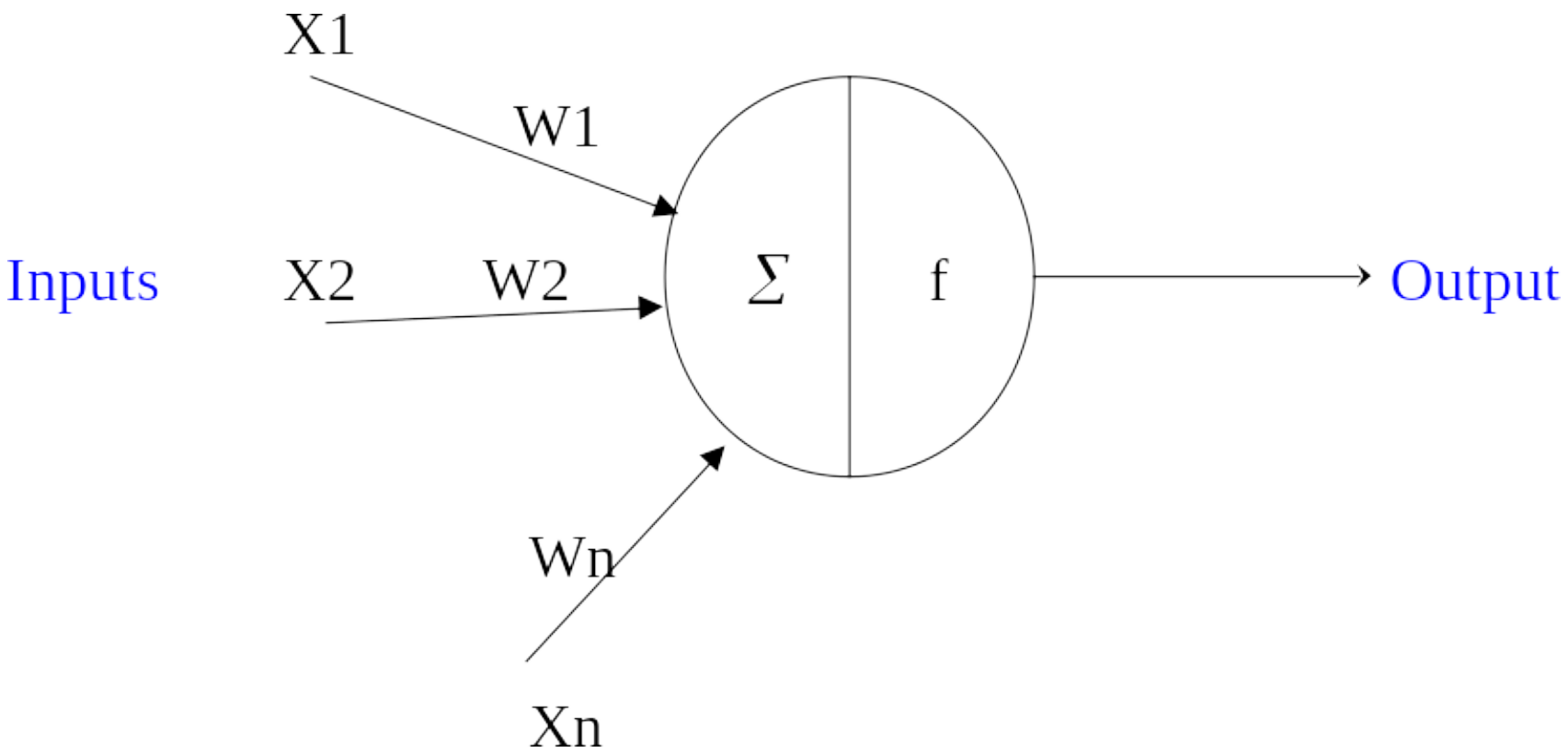
Real Neural Learning

- Synapsis change size and strength with experience.
- When 2 connected neurons are firing at the same time, the strength of the synapse between them increase, i.e. Hebbian Learning.
- *"Neurons that fire together, wire together."*

Neural Network



Simple Aritifial Neuron



Neuron Model

- A neuron has more than one input, $x_1, x_2, \cdots, x_n$.
- Each input is associated with a weight $w_1, w_2, \cdots, w_n$.
- The neuron has a bias b.
- The net input of the neuron is calculated as follows:

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{1}$$

- The neuron output is

$$y = f(z), \tag{2}$$

where f is called **transfer function**.

Transfer Function

We have 3 common transfer functions:

- Hard limit transfer function

$$y = f(z) = \begin{cases} 1, & z \geq \tau \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

  where τ is a certain desired threshold.

- Linear transfer function,

$$y = f(z) = a \times z + b, \tag{4}$$

  where a and b are constants, and

- Sigmoid transfer function

$$f(z) = \frac{1}{1 + e^{-z}}. \tag{5}$$

Architecture of ANN

- *Feed-forward network.* Allow the signals to travel one way from the input layer to the output layer, in case of multilayer ANN.
- *Feed-back networks.* The signal travels as loops in the network, the output is connected to the input of the network.

Perceptron

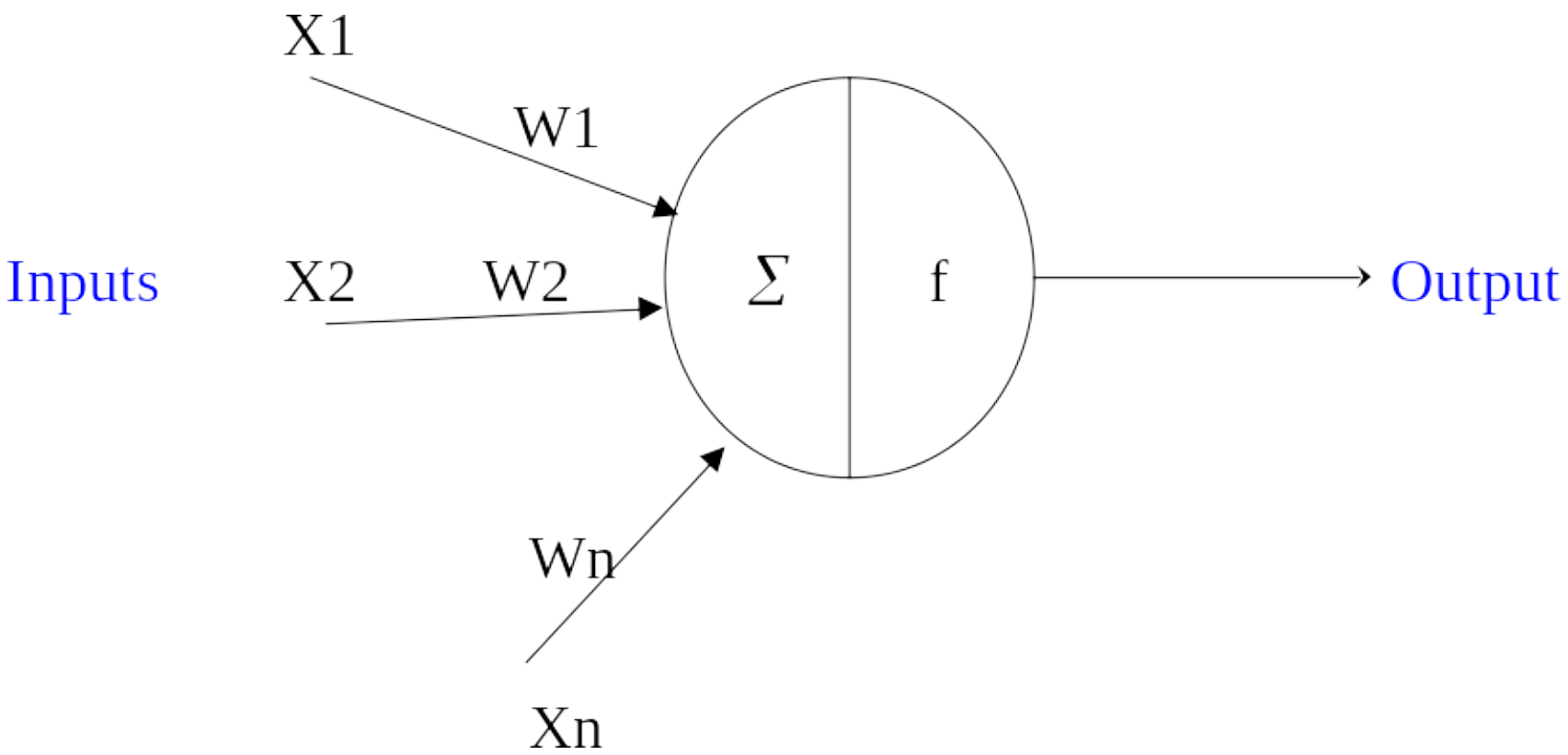


Figure 1: It is a network of one neuron and hard limit transfer function

Perceptron Learning Rule

$$W_{new} = W_{old} + (t - a)X, \tag{6}$$

where W_{new} is the new weight,
W_{old} is the old value of the weight,
X is the input value,
t is the desired output, and
a is the actual value of the output.

Perceptron Example

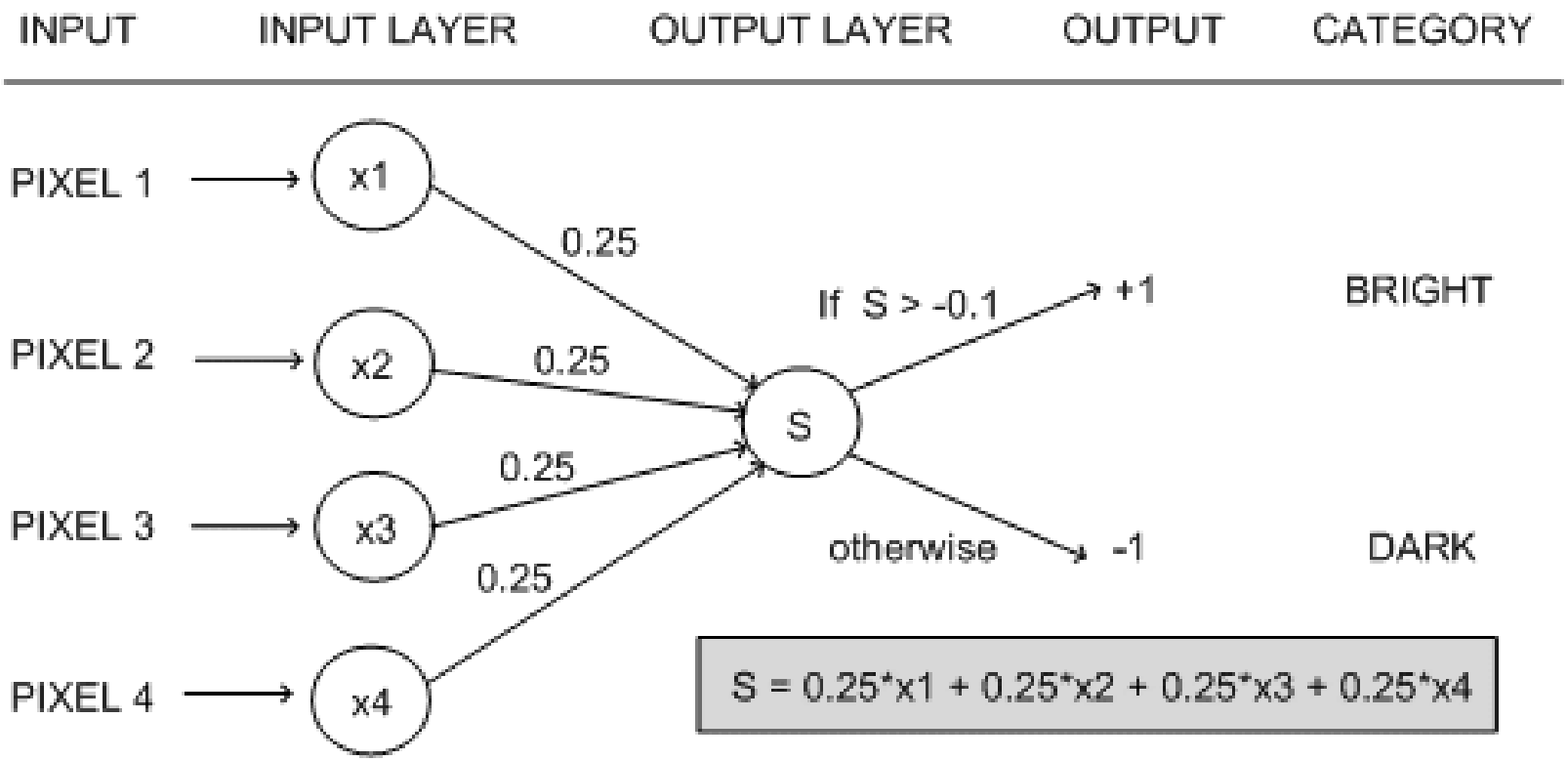


Figure 2: A perceptron working example, categorizing image to either: *bright* or *black*.

Learning Rule I

- The perceptron learning rule can not be applied to multi-layer network, so we use back propagation algorithm for the learning process.

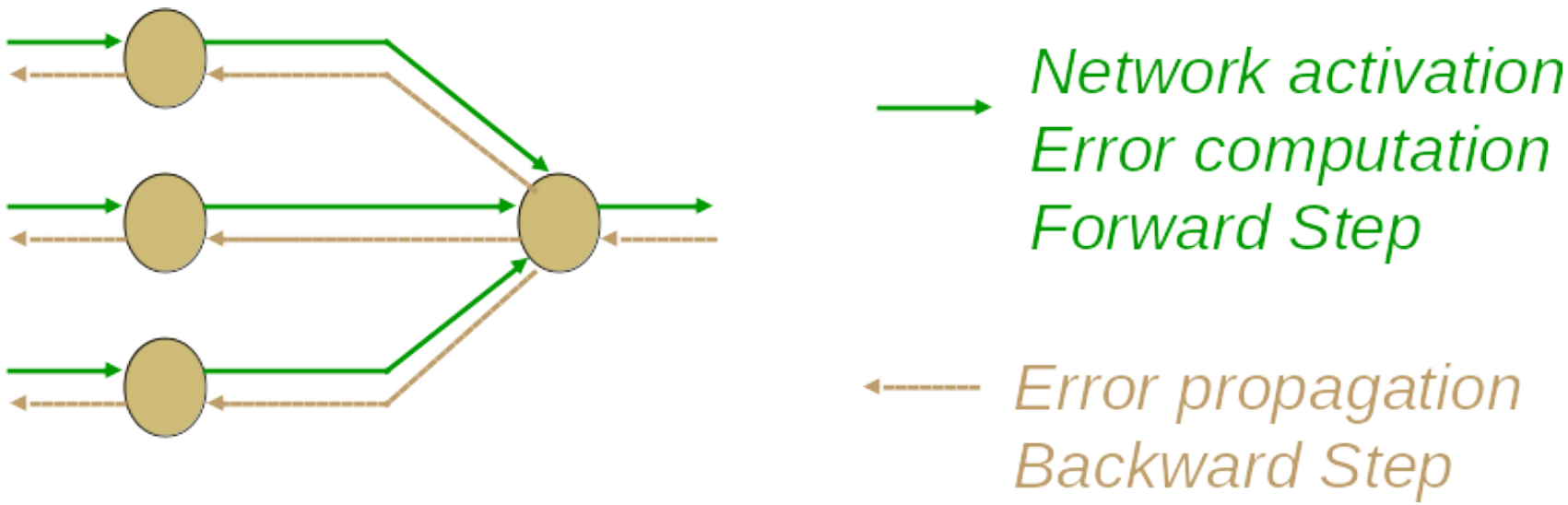


Figure 3: The wights are adjusted to minimize the mean squared error.

Learning Rule II

- The weight change rule is:

$$\omega_{ij}^{new} = \omega_{ij}^{old} + \alpha \times \text{error} \times f^{'}(\text{input}), \tag{7}$$

where α is the learning factor,
error the the difference between the actual data and the trained data, and
$f^{'}$ is the derivative of sigmoid function.

Building ANN

How to build an ANN:

1. *Define the problem in terms of neurons.* Think in terms of the layers.
2. *Represent the information as neurons.* Operationalize neurons, select their data types and locate data for testing and learning.
3. *Define the network.*
4. *Train the network.*
5. *Test the network*

2: Decision Trees

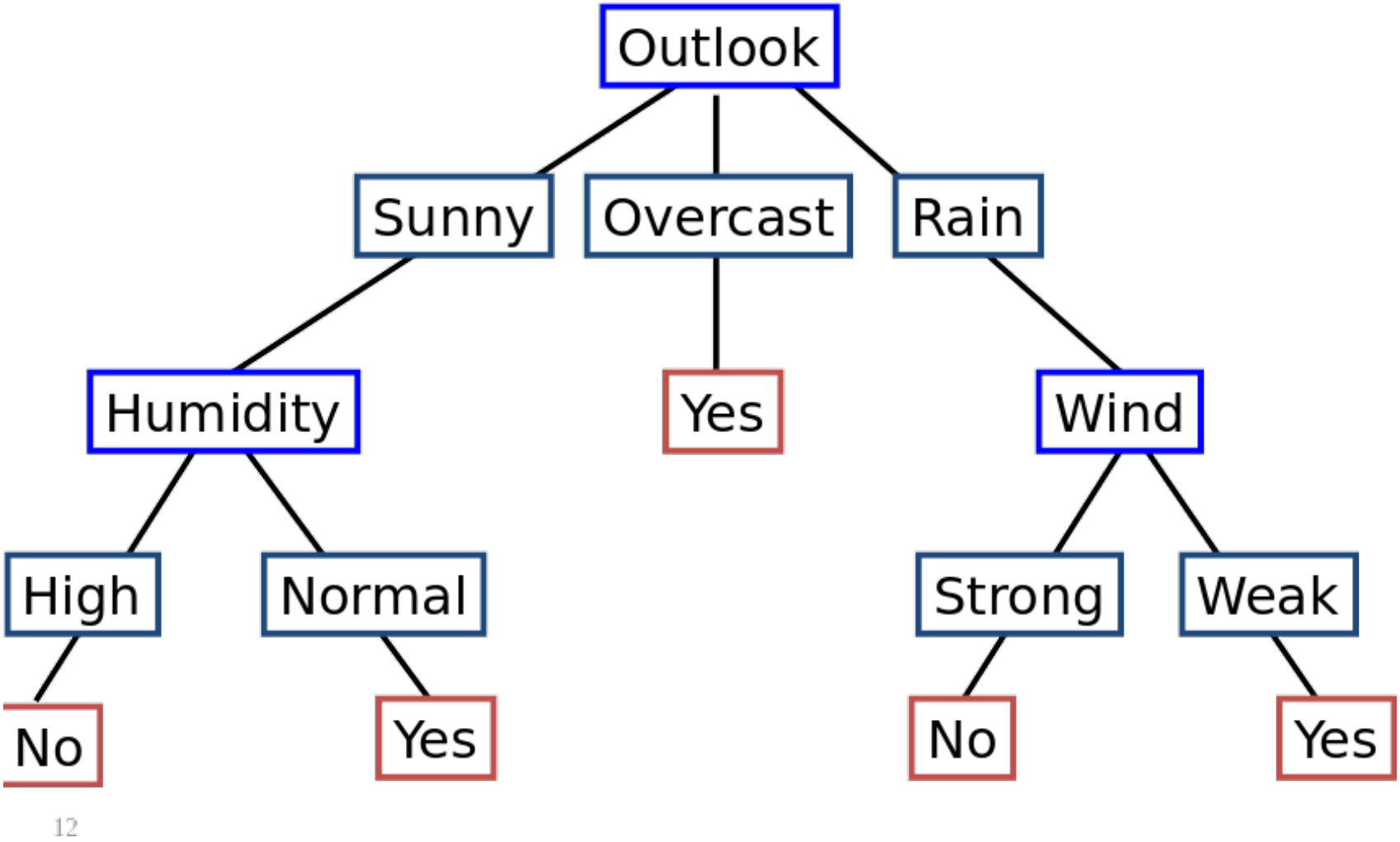
Descision Trees

What is a decision tree?

- A flow-chart-like tree structure.
- Internal nodes denotes an attribute
- Branch represents the values of the node attributes.
- Leaf nodes represent class labels or class distribution.

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Figure 4: A table represent decisions based on certain attributes.



12

Figure 5: A decision tree based on the given data provided in Figure 4.

Information Gain

$Gain(S, A)$: expected reduction in entropy due to sorting S on attribute A.

$$Gain(S, A) = entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} entropy(S_v). \quad (8)$$

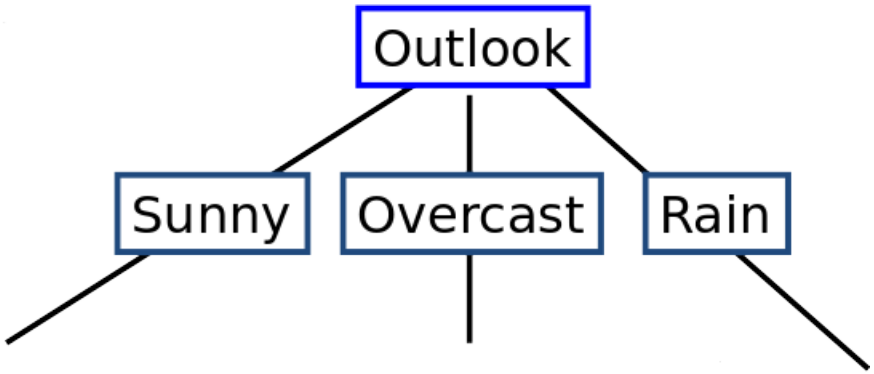
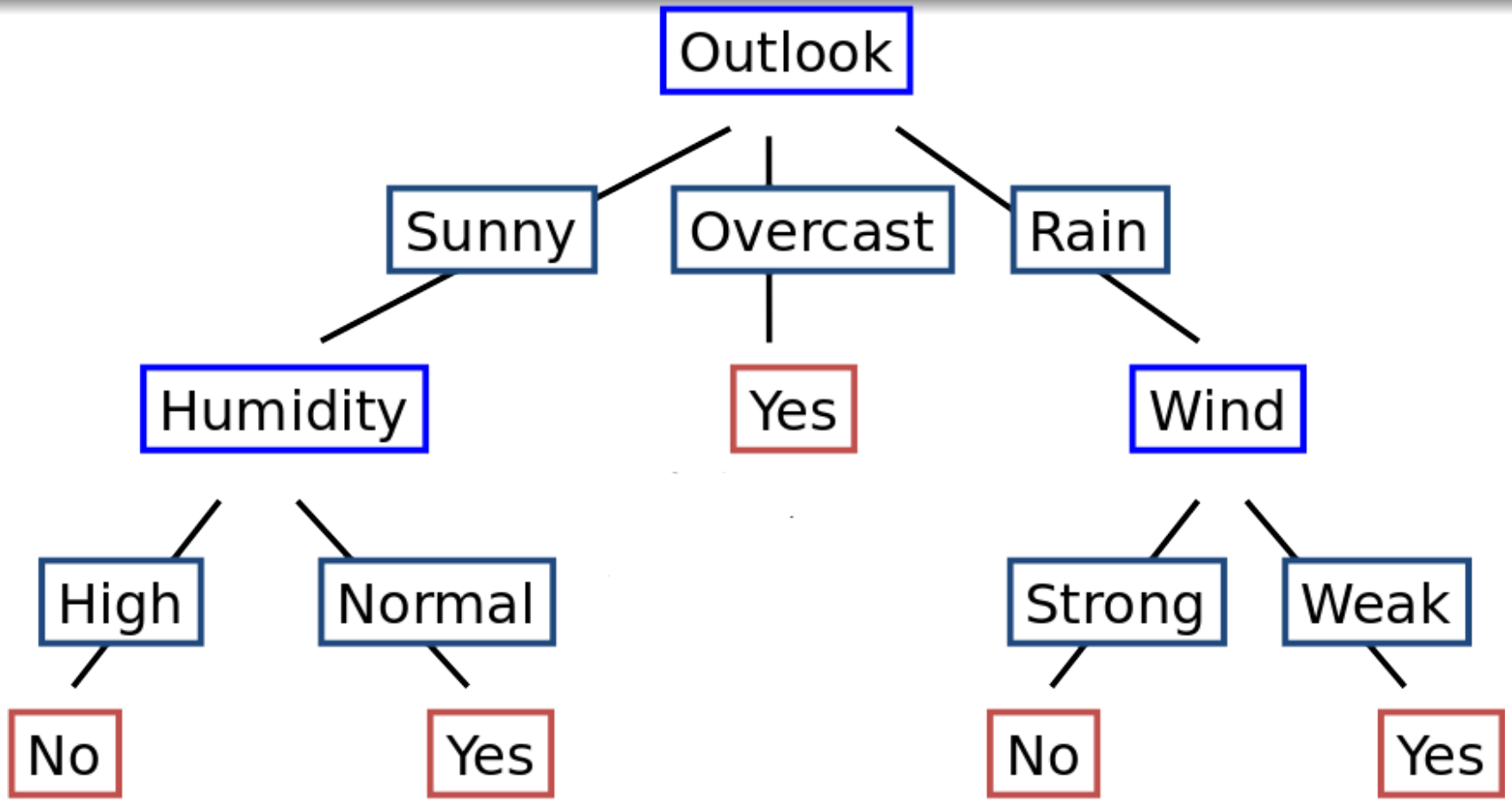


Figure 6: First step i building a decision tree: testing attributes and entropy.

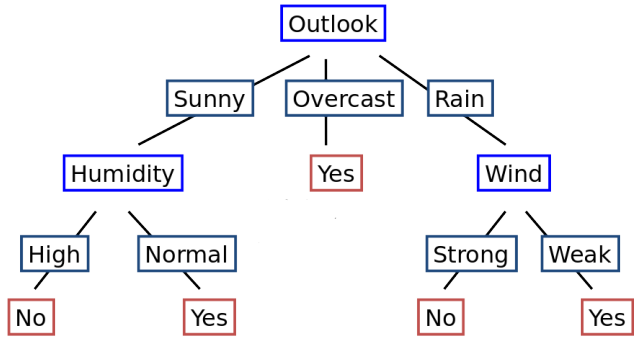
$$Gain(S_{sunny}, Humidity) = 0.97$$
$$Gain(S_{sunny}, Temp) = 0.57$$
$$Gain(S_{sunny}, Wind) = 0.019$$



21

Figure 7: The full tree of the given example.

Converting a Tree to Rules



- if(Outlook=Sunny) and (Humidity=High), then PlayTennis=No
- if(Outlook=Sunny) and (Humidity=Normal), then PlayTennis=Yes
- if(Outlook=Overcast), then PlayTennis=Yes
- if(Outlook=Rain) and (Wind=Strong), then PlayTennis=Normal
- if(Outlook=Rain) and (Wind=Weak), then PlayTennis=Yes

3: Clustering Analysis

Definition

- Clustering is the task of grouping a set of objects such that all the collected object together are similar.
- Clustering techniques is applied when prediction classes are no defined or known.
- Unsupervised learning.

Algorithms

There are several clustering algorithm which can be *"clustered"* together based on their model, e.g.

- Centroid-base,
- Grid-based,
- Denisty-based,
- Distribution-based, and
- Connectivity-based

Centroid-based algorithm (k-means) I

The algorithm is carried as follows:

1. Partition objects into k nonempty subsets
2. Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., mean point, of the cluster)
3. Assign each object to the cluster with the nearest seed point
4. Go back to Step 2, stop when the assignment does not change.

4: Measuring the Performance of a Classifier

Confusion Matrix

		Predicted data	
		+	−
Actual data	+	TP	FN
	−	FP	TN

- *True positive (TP).* The number of positive instances that are classified as positive.
- *False positive (FP).* The number of negative instances that are classified as positive.
- *False negative (FN).* The number of positive instances that are classified as negative.
- *True negative (TN).* The number of negative instances that are classified as negative.

Types of Errors

- *False positive.* Occurs when instances that should be classified as negative are classified as positive.
- *False negative.* Occurs when instances that should be classified as positive are classified as negative.

Summary

Questions?