Introduction to Deep Learning Applications and Theory

Lecture 4
Artificial Neural Network (ANN)

Chhoeum Vantha, Ph.D.
Telecom & Electronic Engineering

Previous week: Properties of Brain Networks and Learning

1. Building Blocks of a Neural Network
2. Neural Net Workflow Steps

Outline

- Artificial Neural Networks
- Properties of ANNs
- Applications of ANNs
- Type of Artificial Neural Networks
- The Perceptron

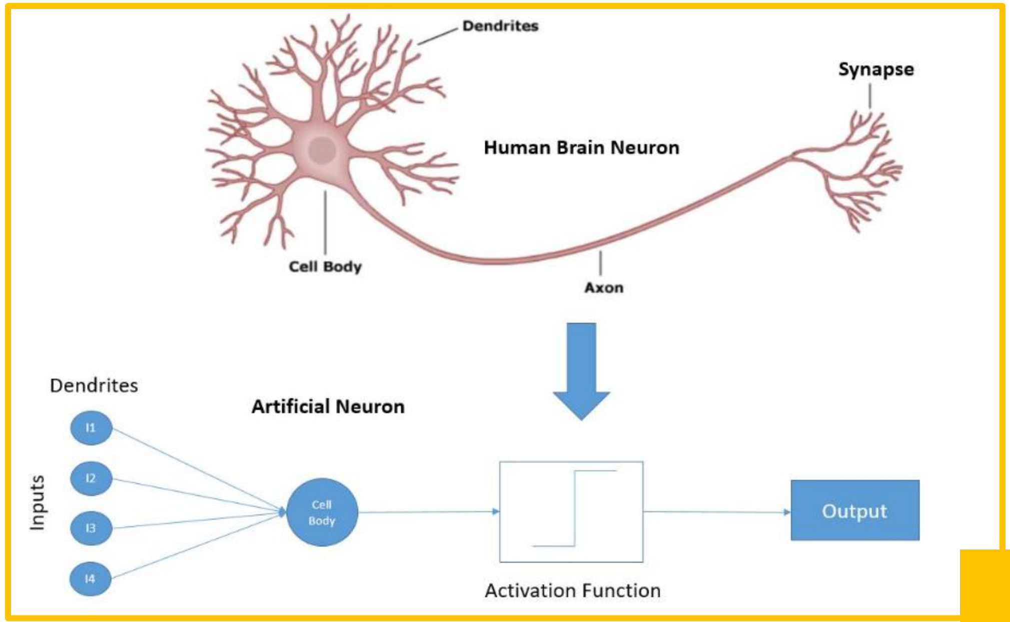
Artificial Neural Networks

Computational models inspired by the human brain:

- Massively parallel, distributed system, made up of simple processing units (neurons)
- Synaptic connection strengths among neurons are used to store the acquired knowledge.
- Knowledge is acquired by

  the network from its environment

  through a learning process



Properties of ANNs

Learning from examples/features

- labeled or unlabeled

Adaptivity

- changing the connection strengths to learn things

Non-linearity

- the non-linear activation functions are essential

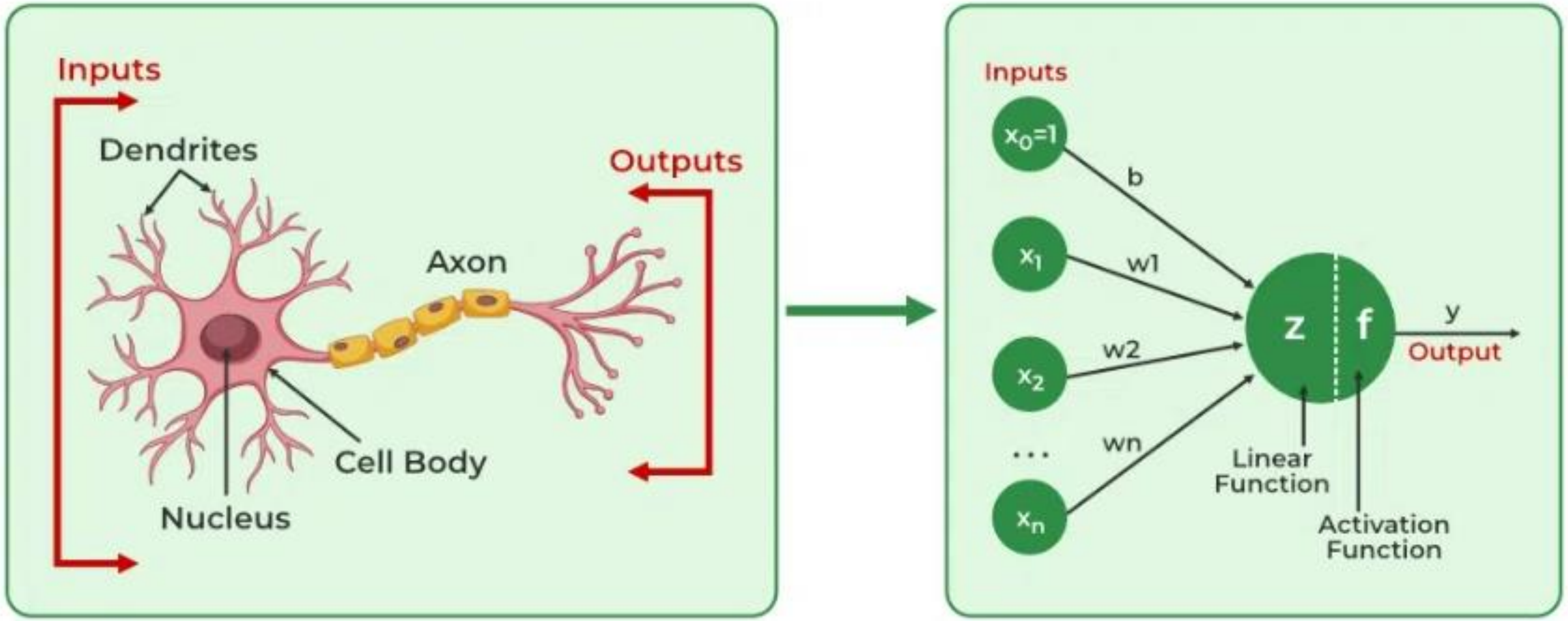
Fault tolerance

- if one of the neurons or connections is damaged, the whole network still works quite well

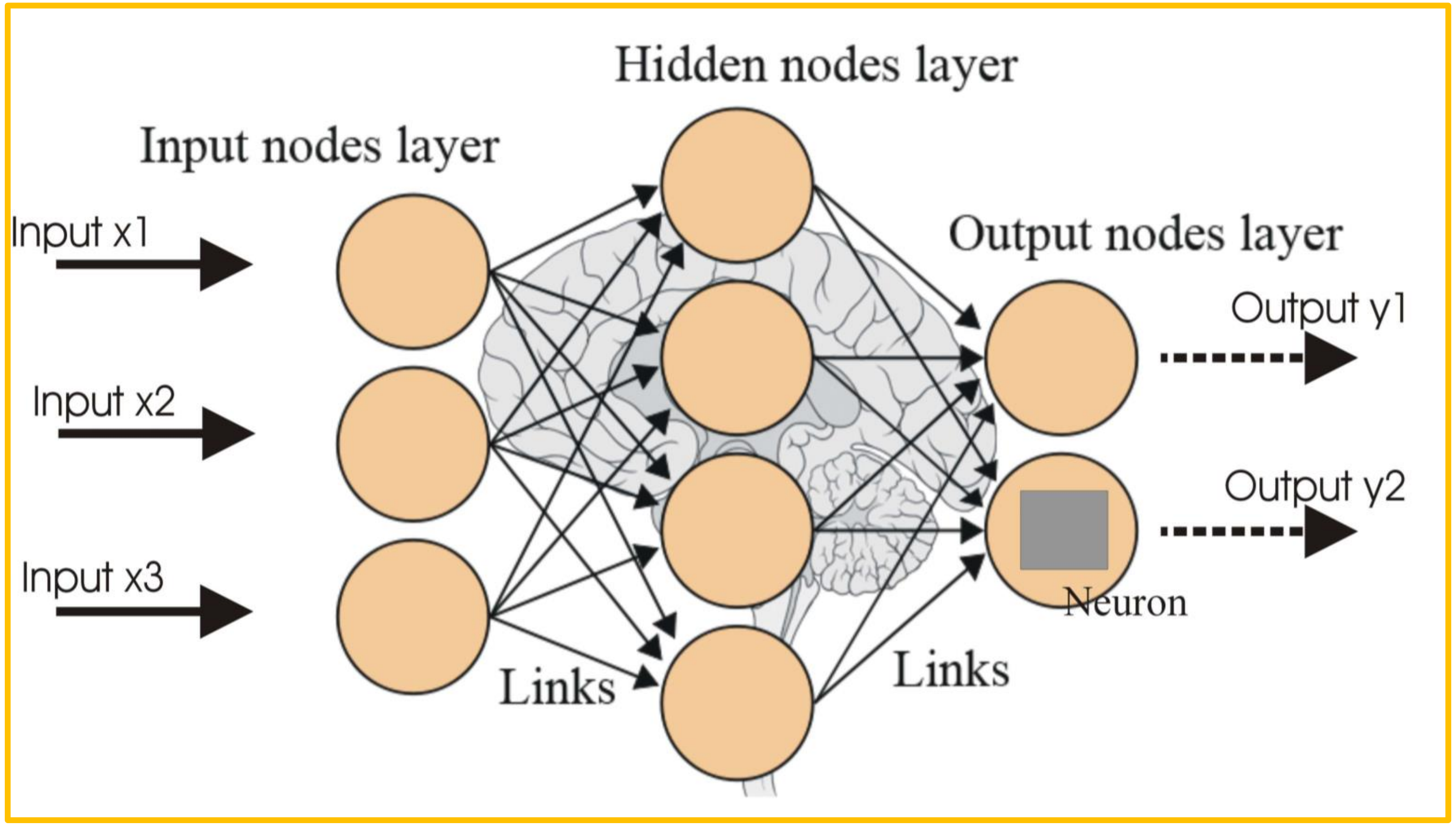
Properties of ANNs

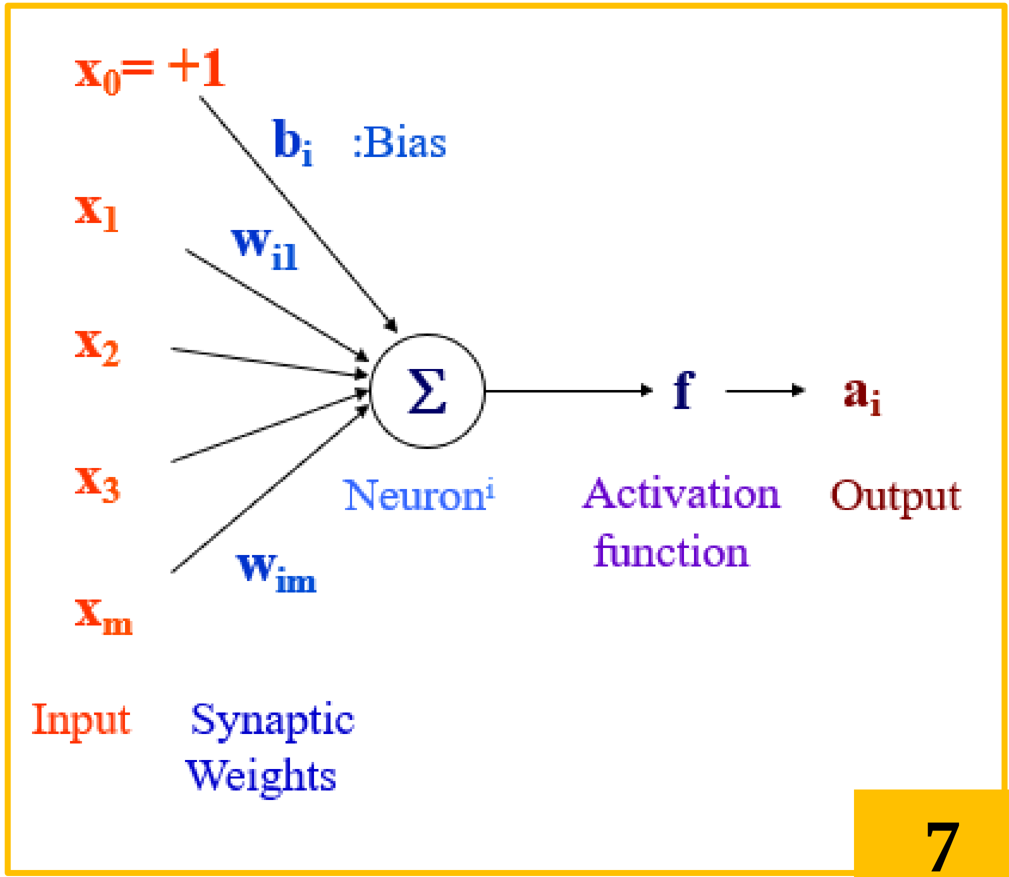
ANN might be better alternatives than classical solutions for problems characterized by:

- high dimensionality, noisy, imprecise or imperfect data
- And a lack of a clearly stated mathematical solution or algorithm



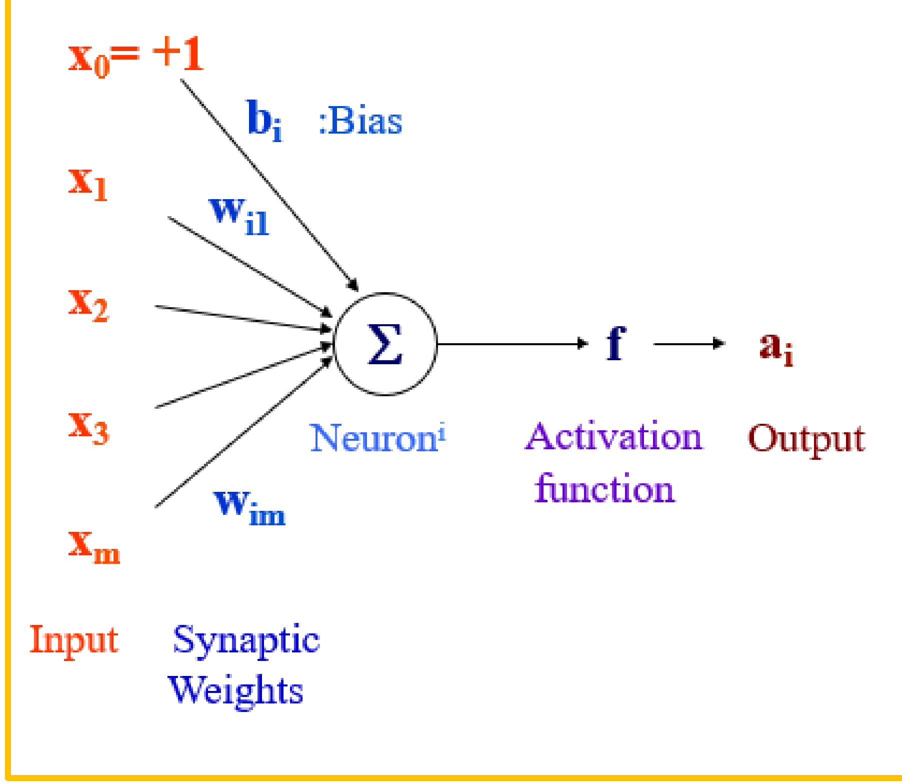
Properties of ANNs





Properties of ANNs

$$a_i = f(n_i) = f\left(\sum_{j=1}^{n} w_{ij}x_j + b_i\right)$$



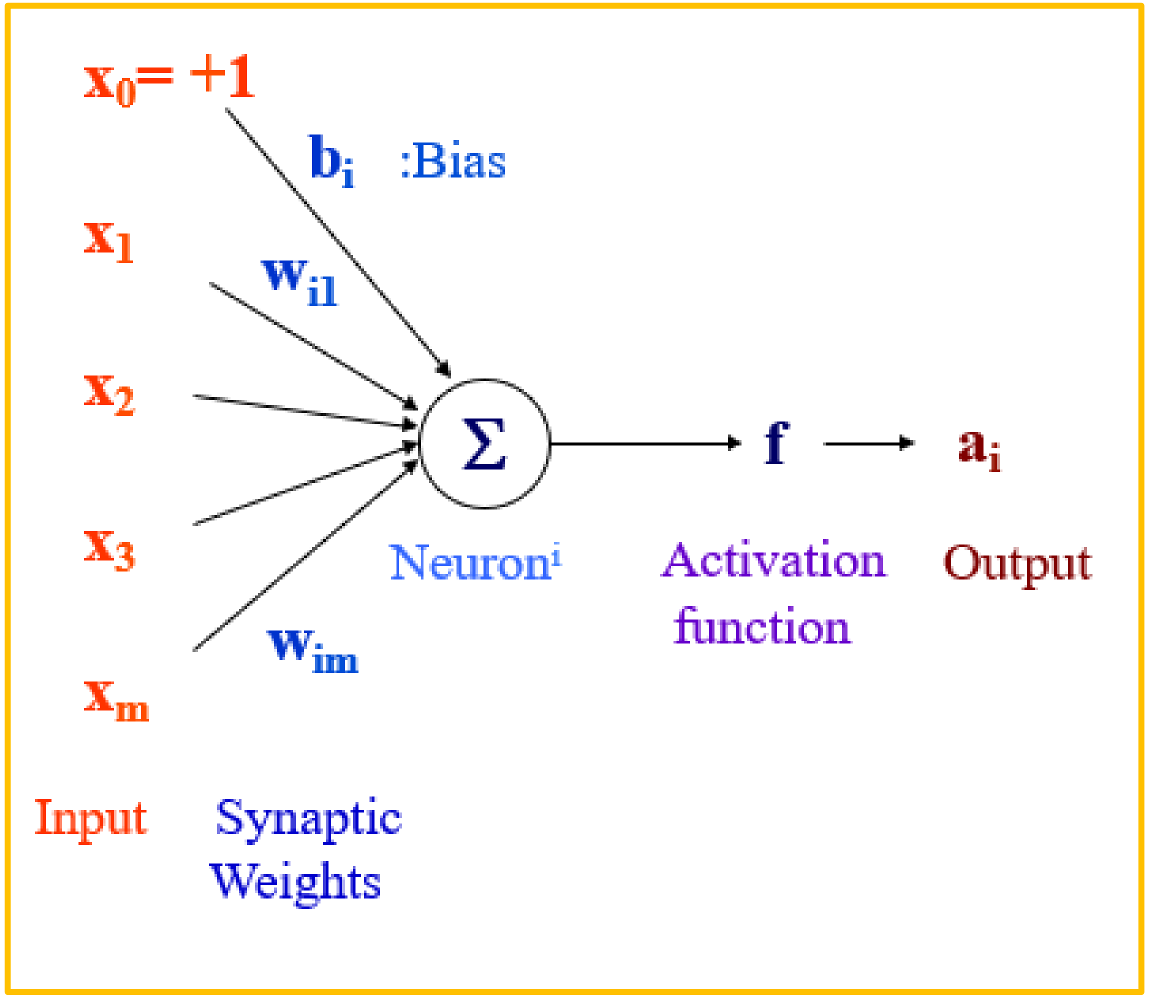
An artificial neuron:

- computes the weighted sum of its input (called its **net input**)
- adds its bias
- passes this value through an activation function

We say that the neuron “fires” (i.e. becomes active) if its output is above zero.

Weight

- Weights are **numerical values** associated with the connections between neurons.
- They **determine the strength** of these connections and, in turn, the influence that one neuron's output has on another neuron's input
- Each input feature is assigned a weight that determines its influence on the output.
- These weights are adjusted during training to find the optimal values.

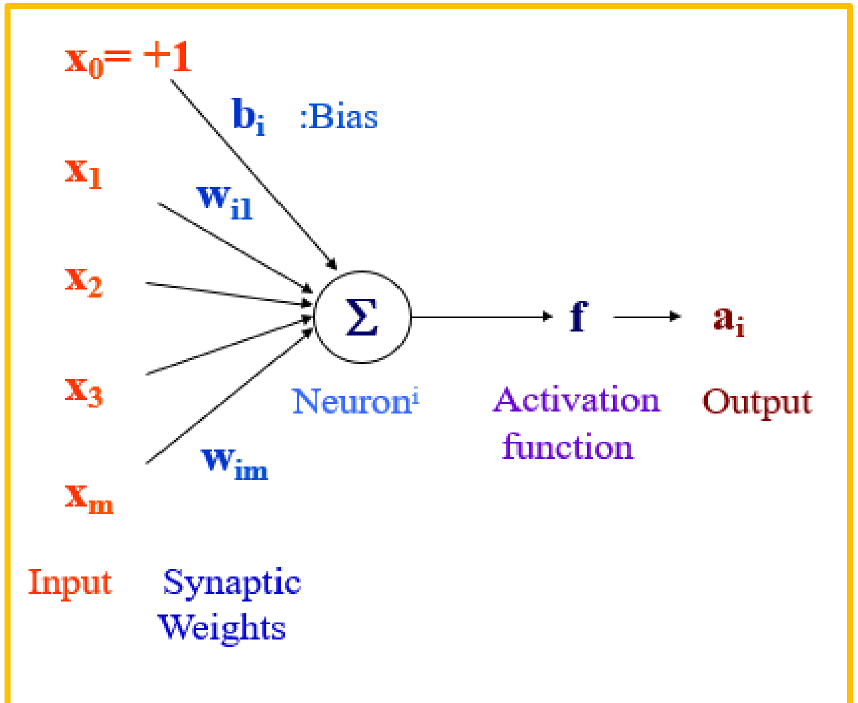


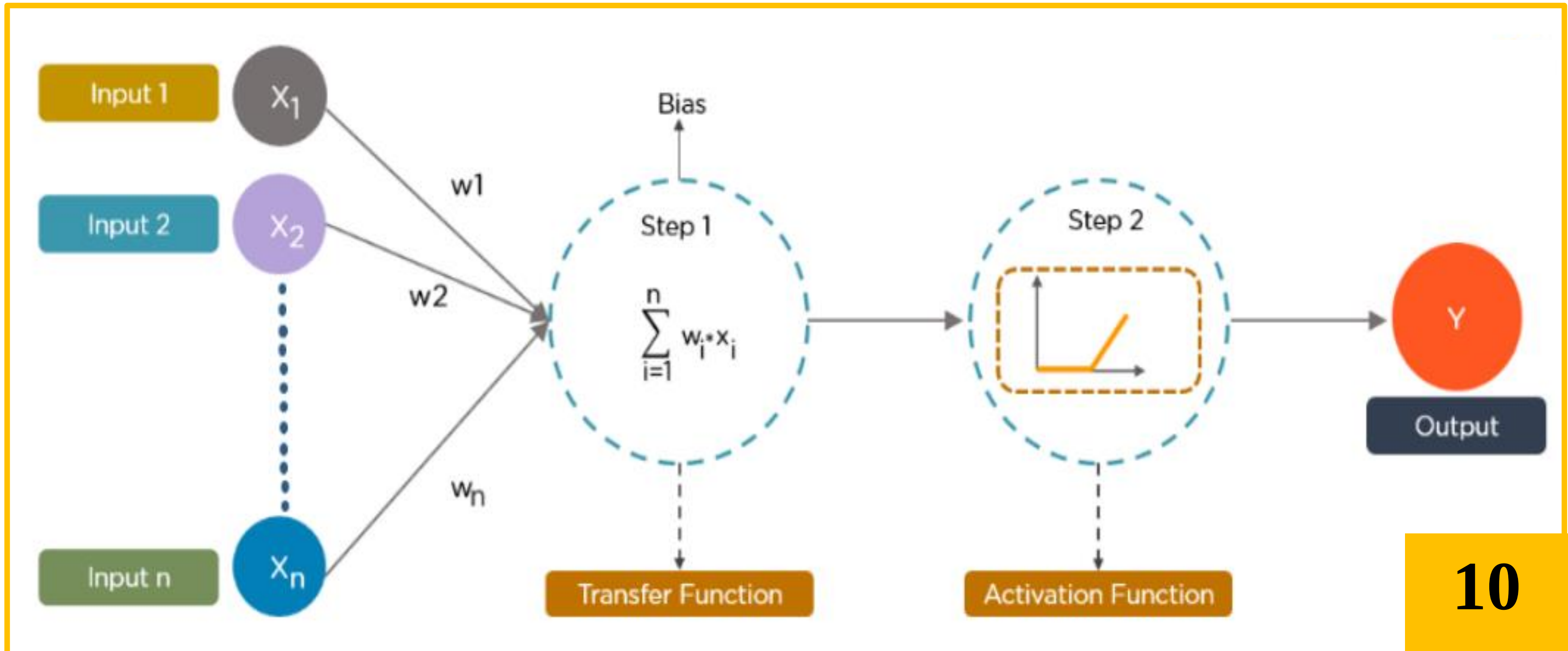
Bias

Bias can be incorporated as another weight clamped to a fixed input of +1.0

This extra free variable (bias) makes the neuron more powerful.

$$a_i = f(n_i) = f\left(\sum_{j=0}^{n} w_{ij} x_j\right) = f(\mathbf{w}_i . \mathbf{x}_j)$$





Applications of ANNs

ANNs have been widely used in various domains for:

- Regression
- Pattern recognition
- Image recognition,
- Speech recognition,
- Machine translation,
- Medical diagnosis
- ...

Type of Artificial Neural Networks

Early ANN Models:

– Perceptron (Frank Rosenblatt in 1957) , ADALINE, Hopfield Network

Current Models:

– Deep Learning Architectures

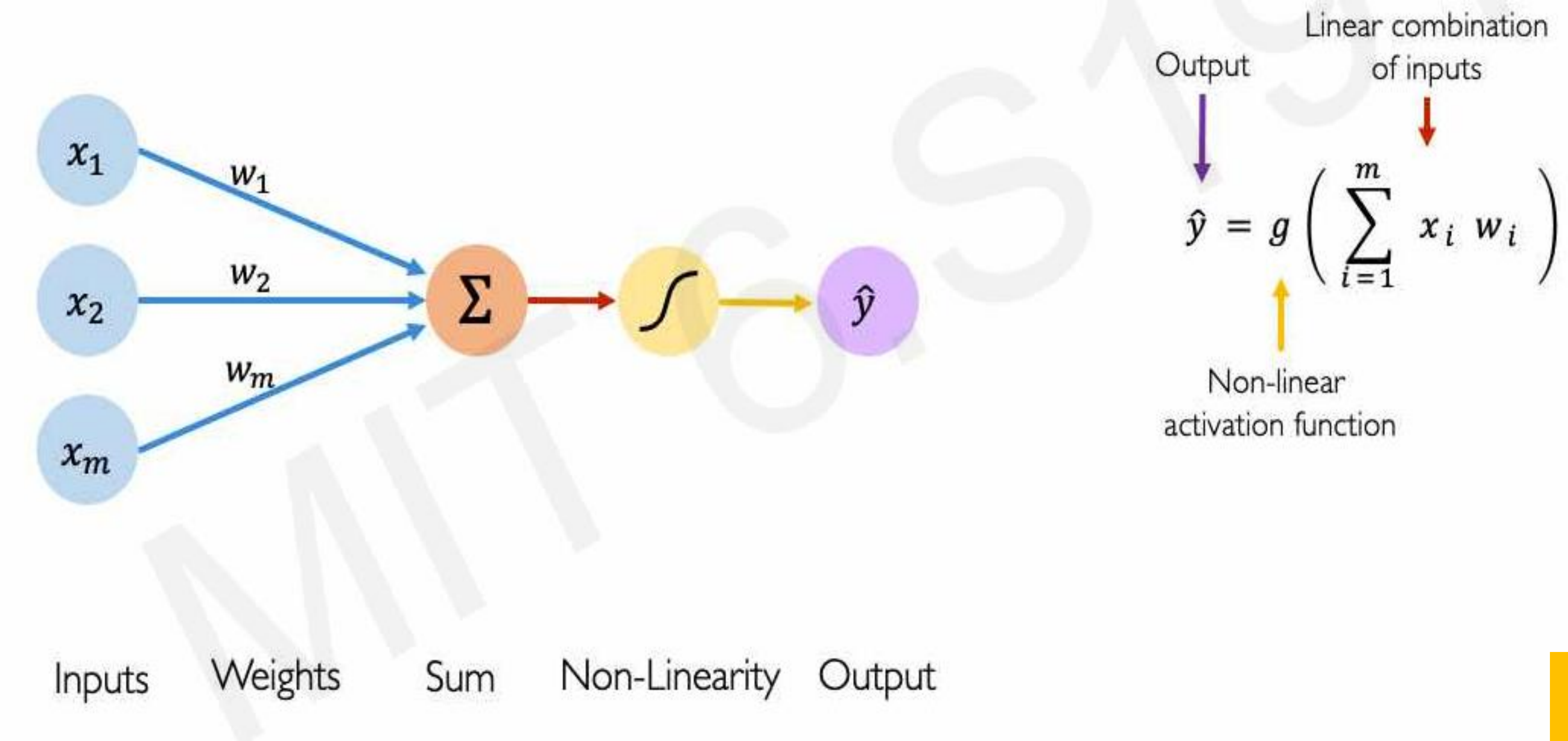
– Multilayer feedforward networks (Multilayer perceptrons)

– Radial Basis Function networks

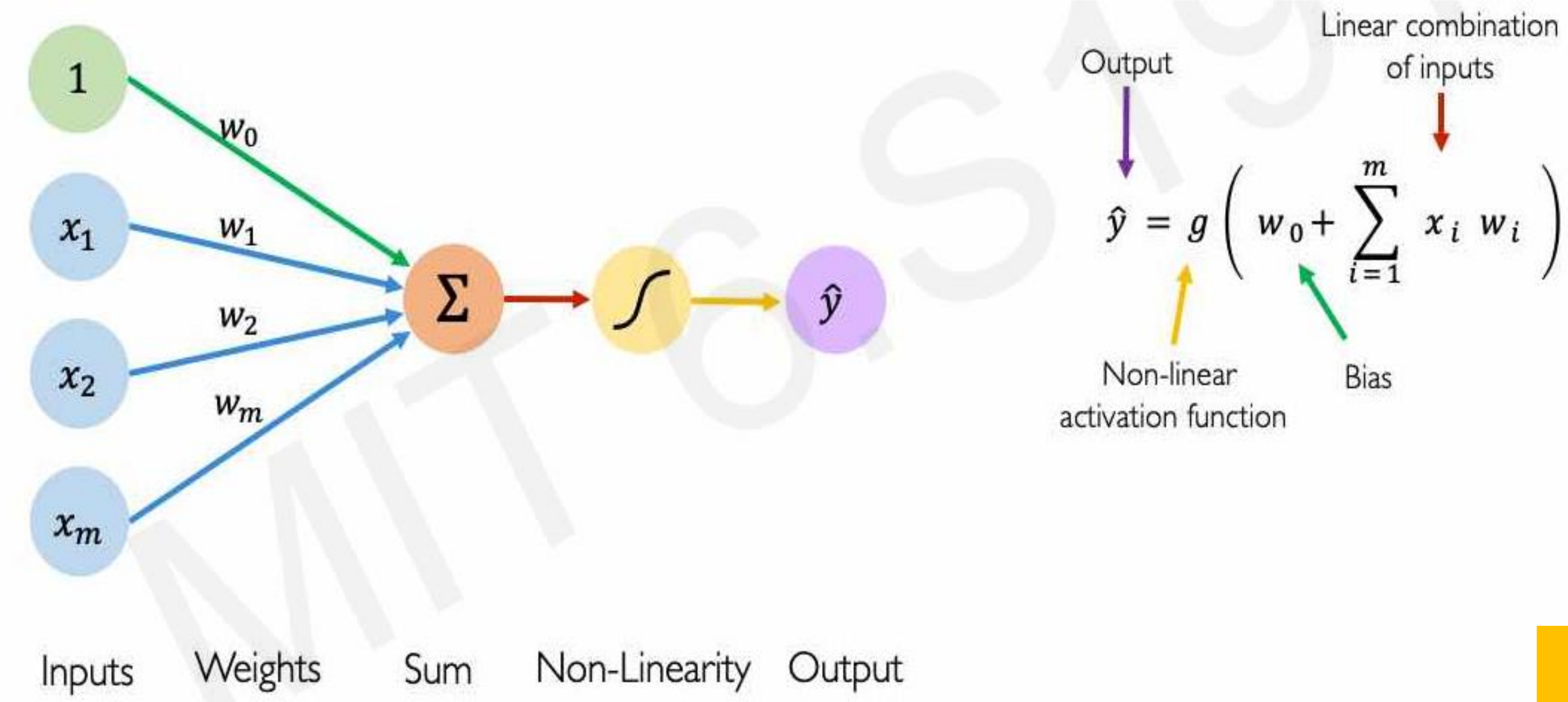
– Self Organizing Networks

– ...

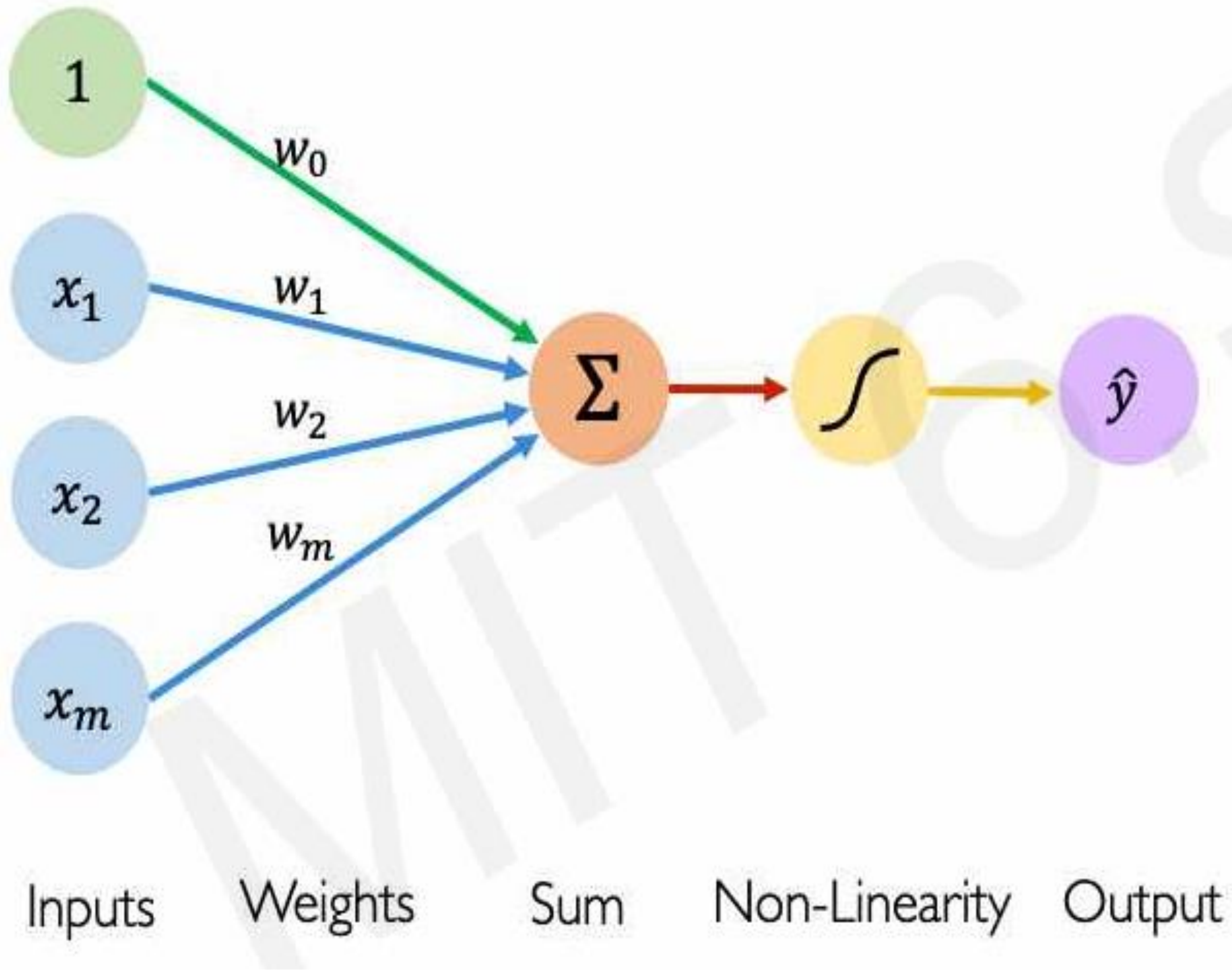
Perceptron: Forward Propagation



Perceptron: Forward Propagation



Perceptron: Forward Propagation

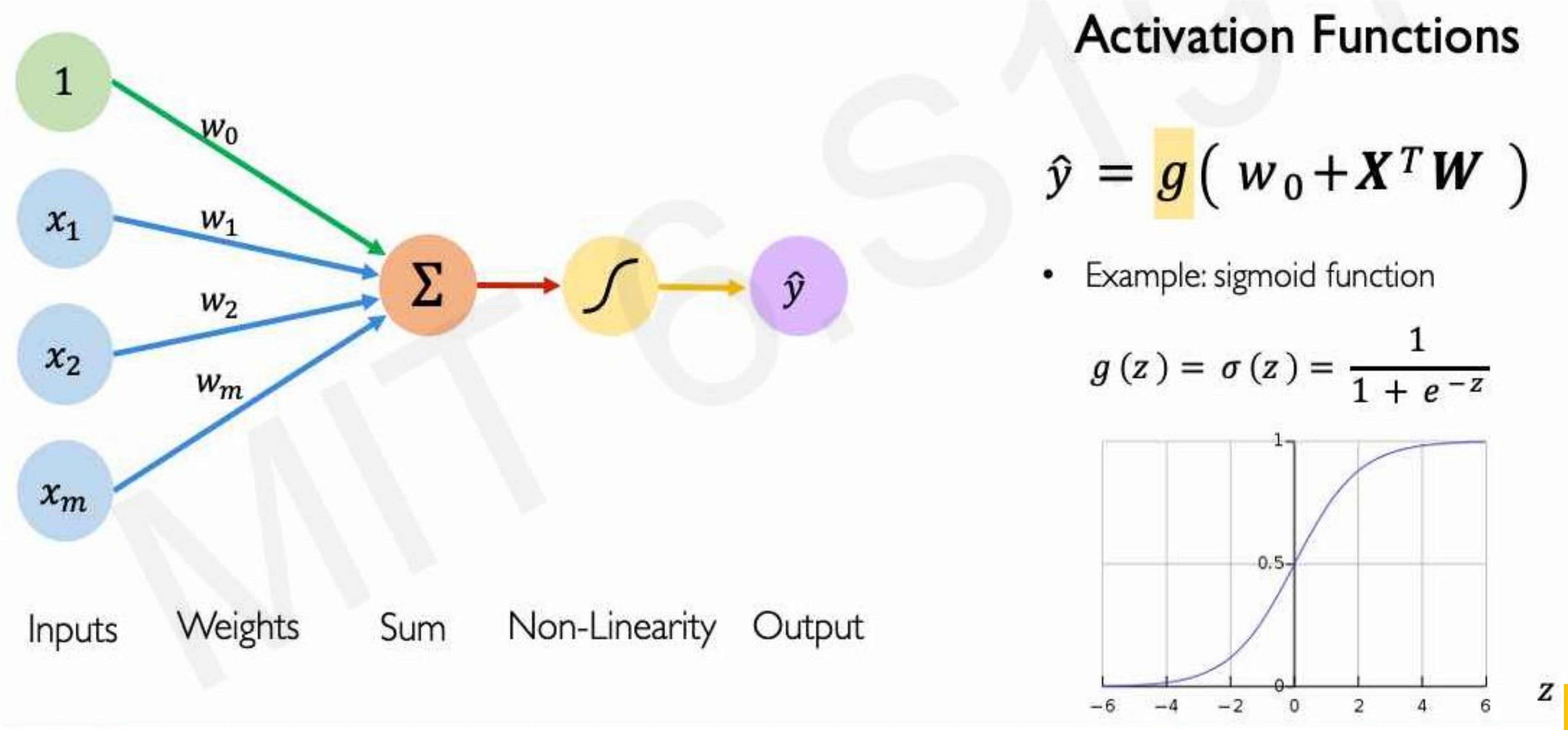


$$\hat{y} = g\left(w_0 + \sum_{i=1}^{m} x_i\, w_i \right)$$

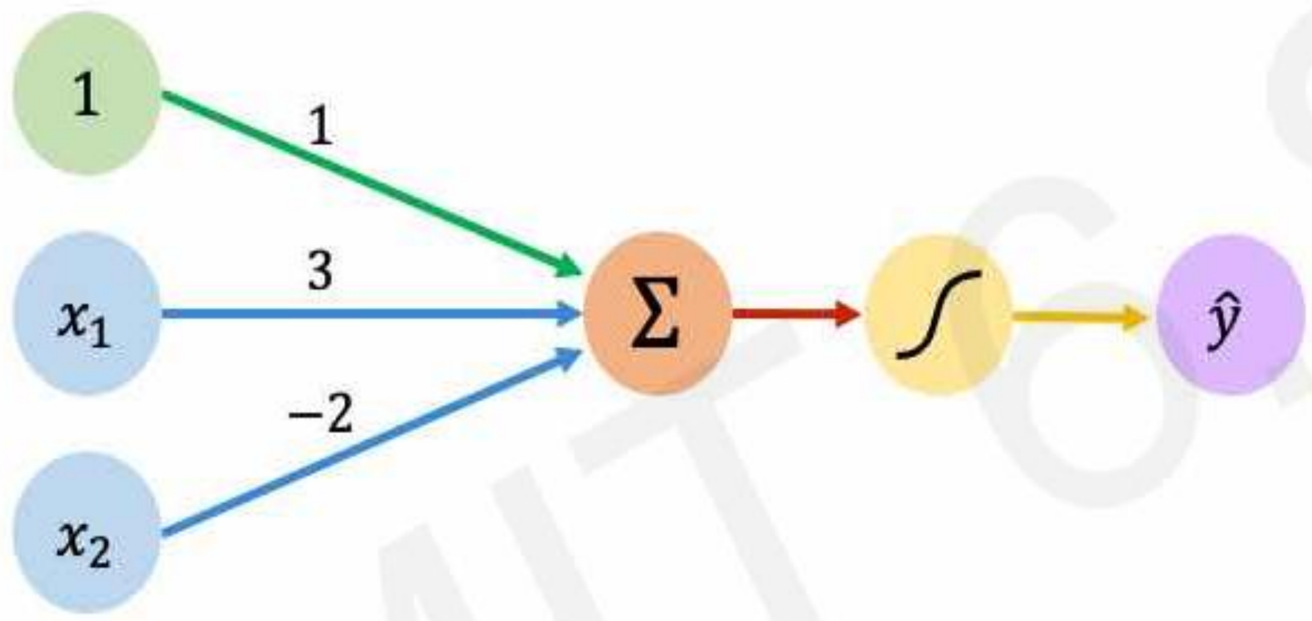
$$\hat{y} = g\left(w_0 + \boldsymbol{X}^T \boldsymbol{W} \right)$$

where: $\boldsymbol{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$ and $\boldsymbol{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$

Perceptron: Forward Propagation



Perceptron: Example



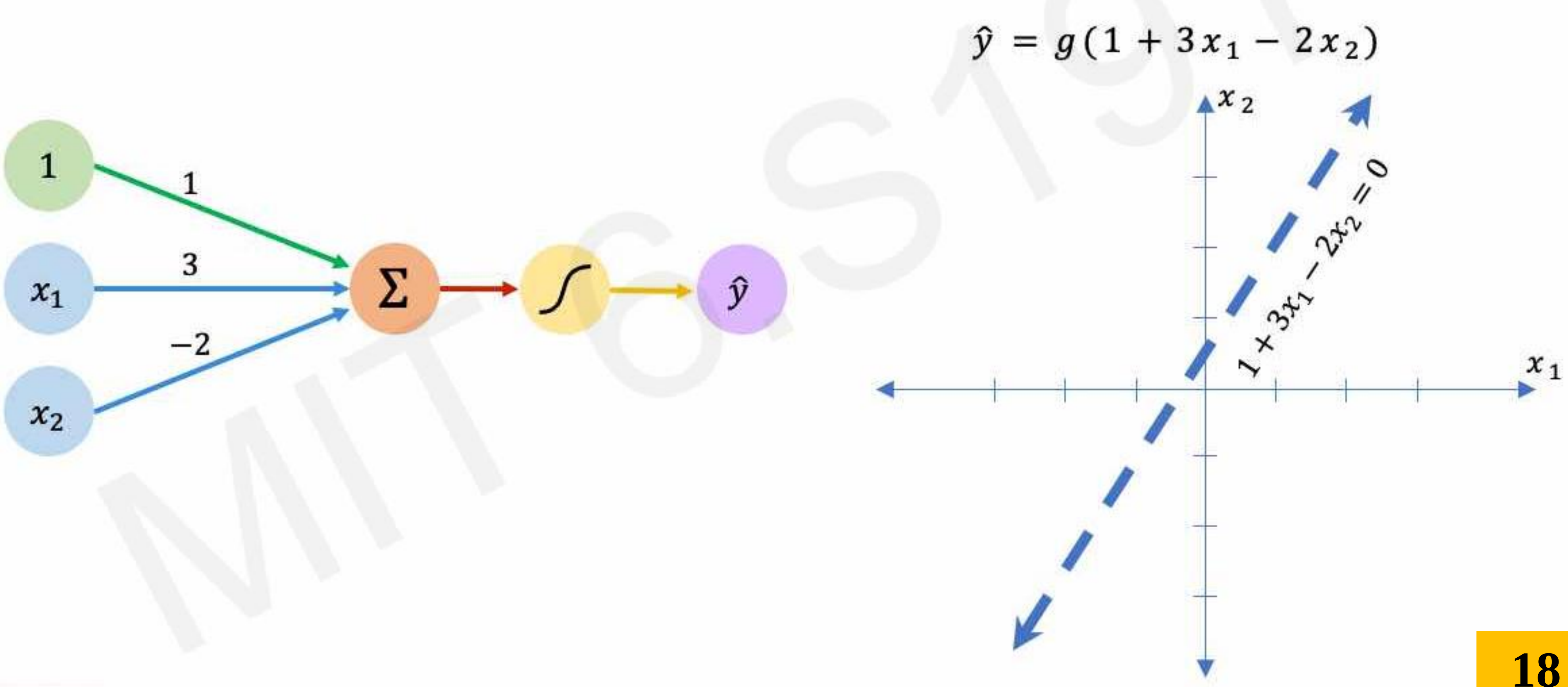
We have: $w_0 = 1$ and $\boldsymbol{W} = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$

$$\hat{y} = g\left(w_0 + \boldsymbol{X}^T \boldsymbol{W} \right)$$

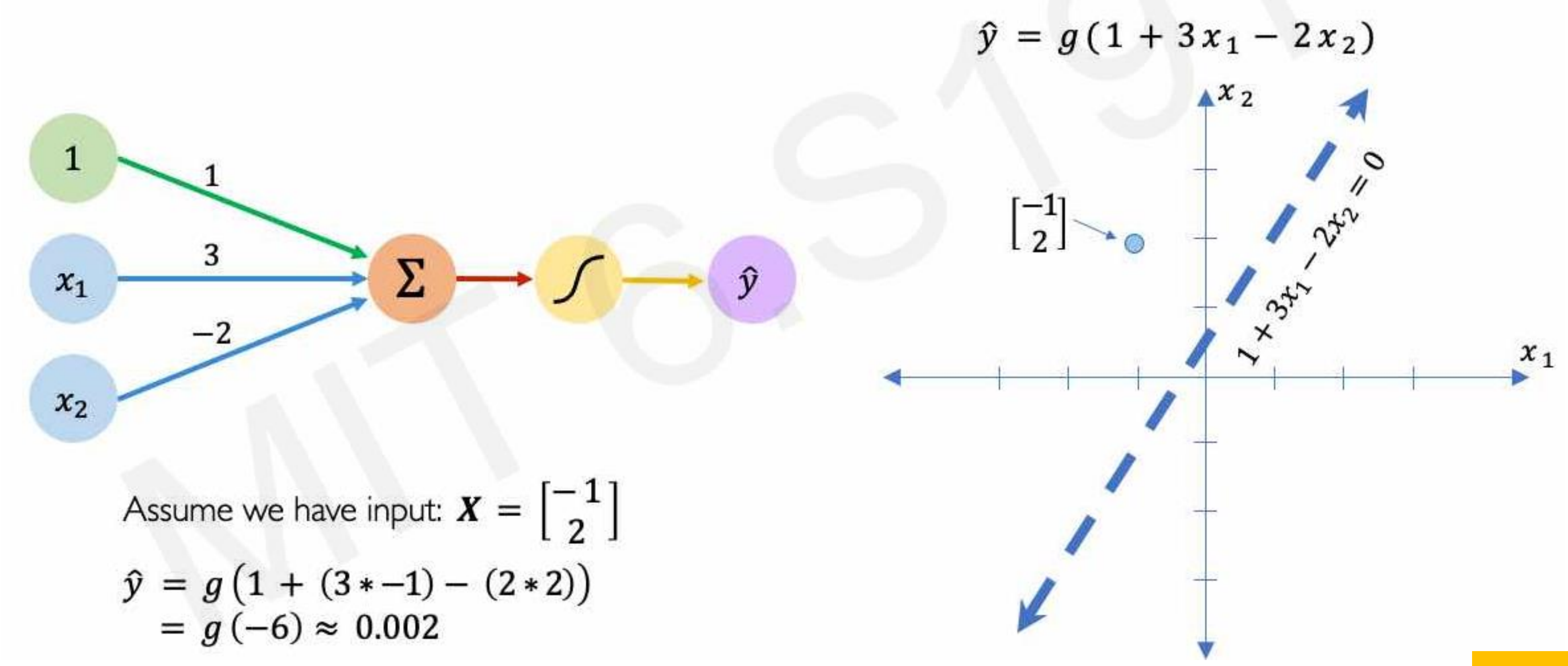
$$= g\left(1 + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 3 \\ -2 \end{bmatrix} \right)$$

$$\hat{y} = g(\underbrace{1 + 3x_1 - 2x_2}_{\text{This is just a line in 2D!}})$$

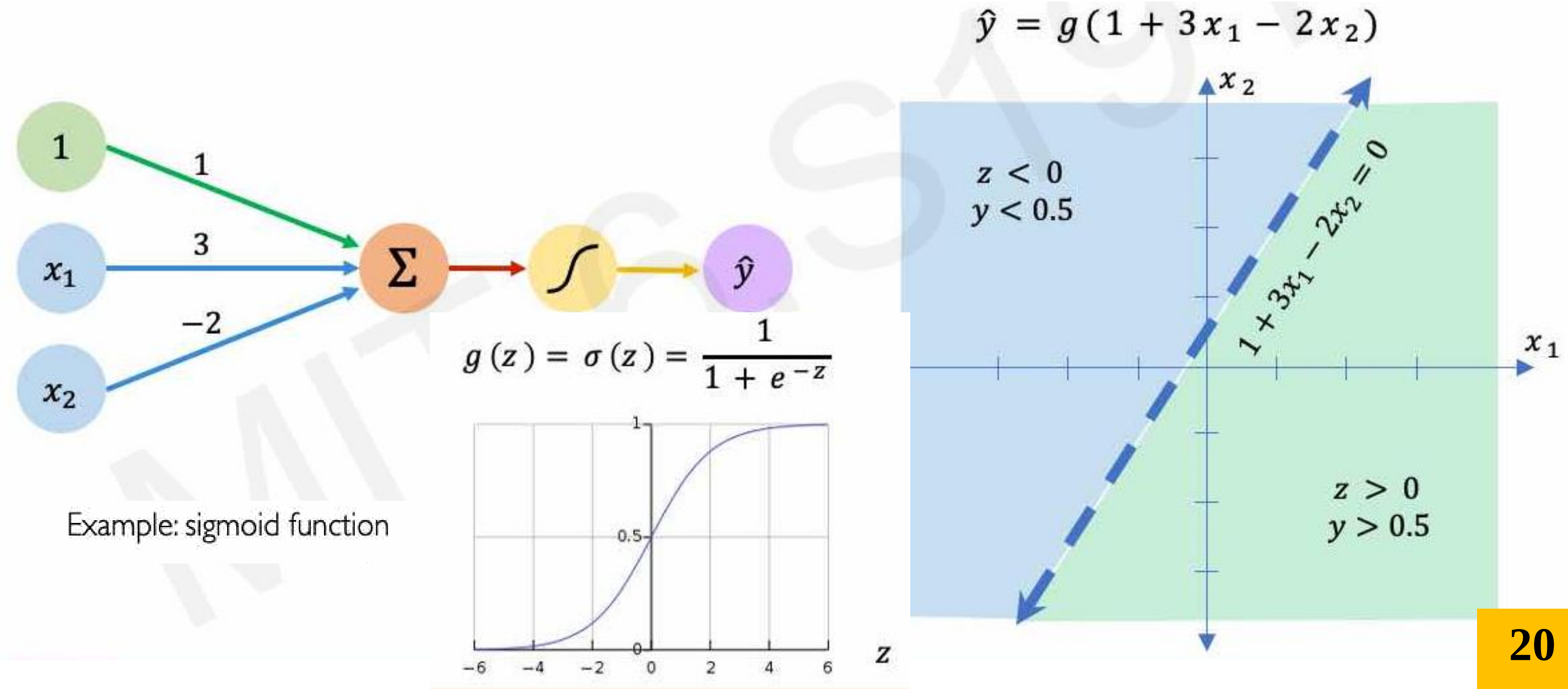
Perceptron: Forward Propagation



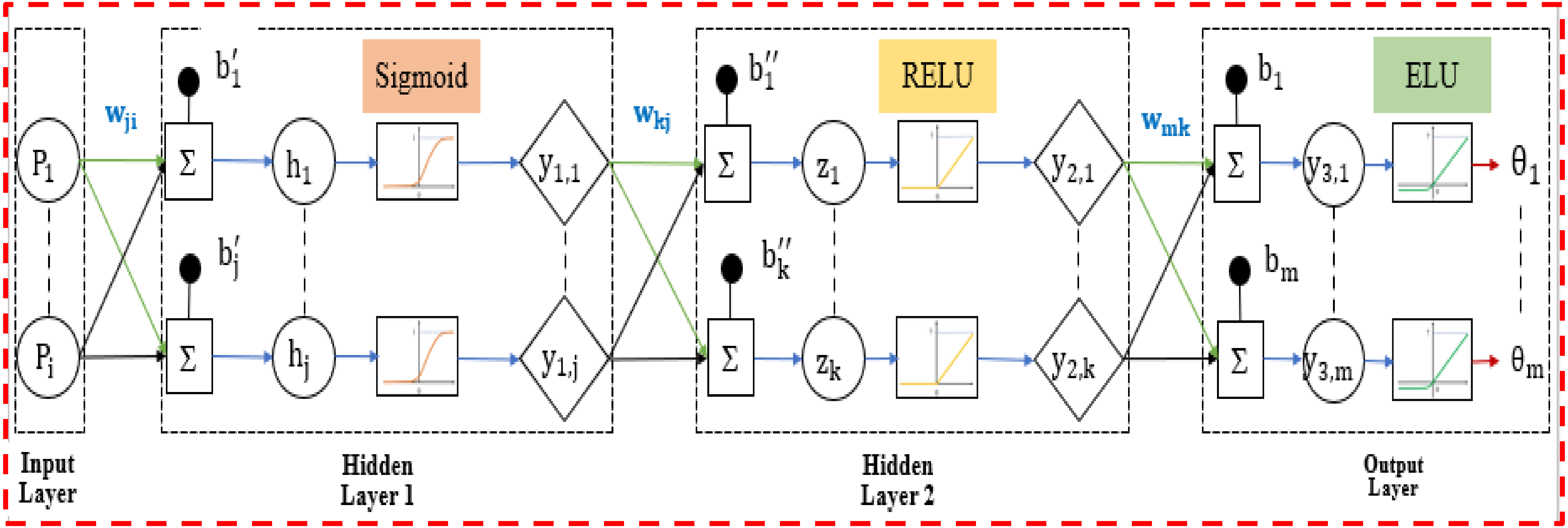
Perceptron: Forward Propagation



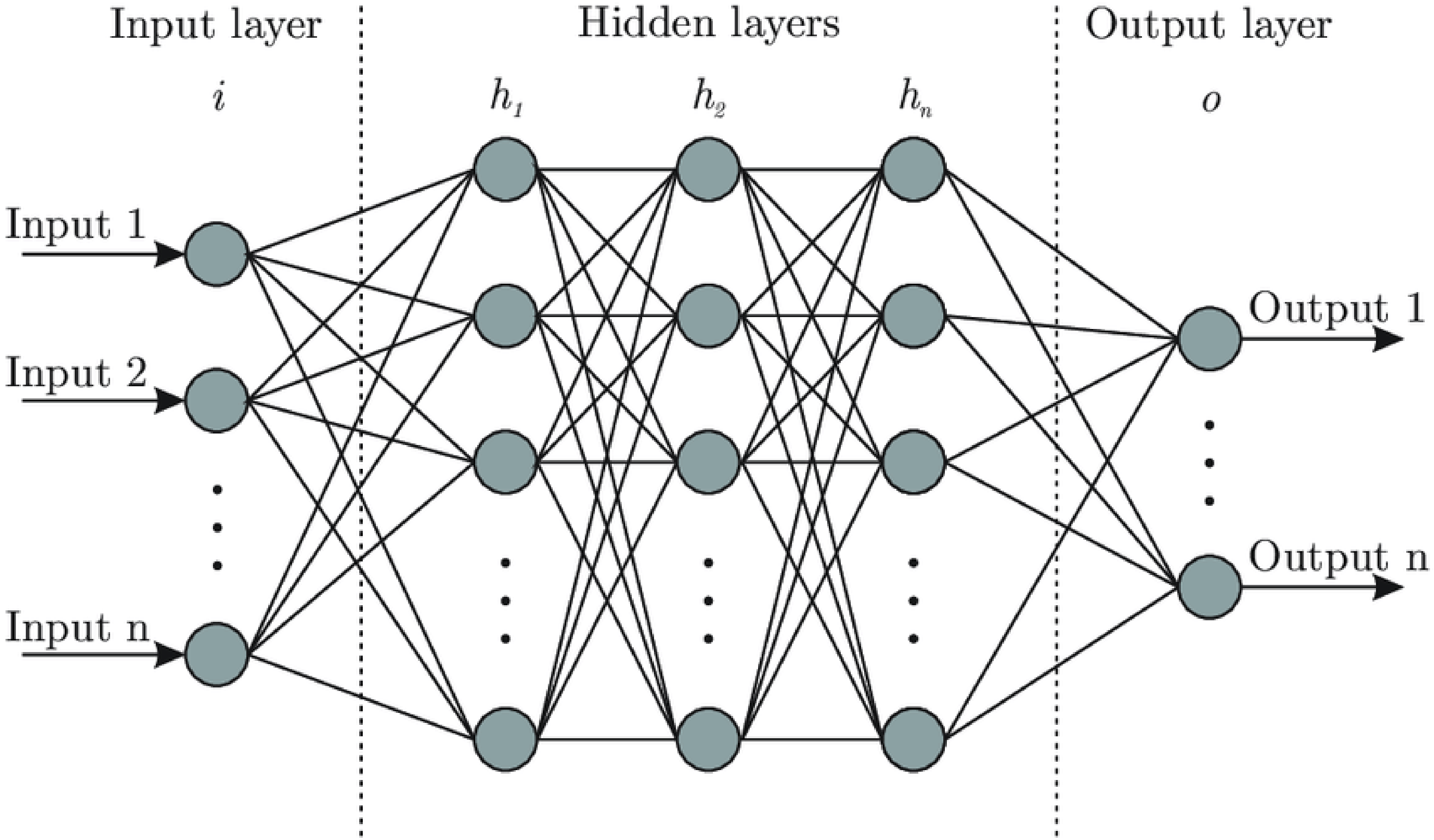
Perceptron: Forward Propagation



Artificial Neuron Model with Multiple hidden layers



Artificial Neuron Model with Multiple hidden layers



Practice 4

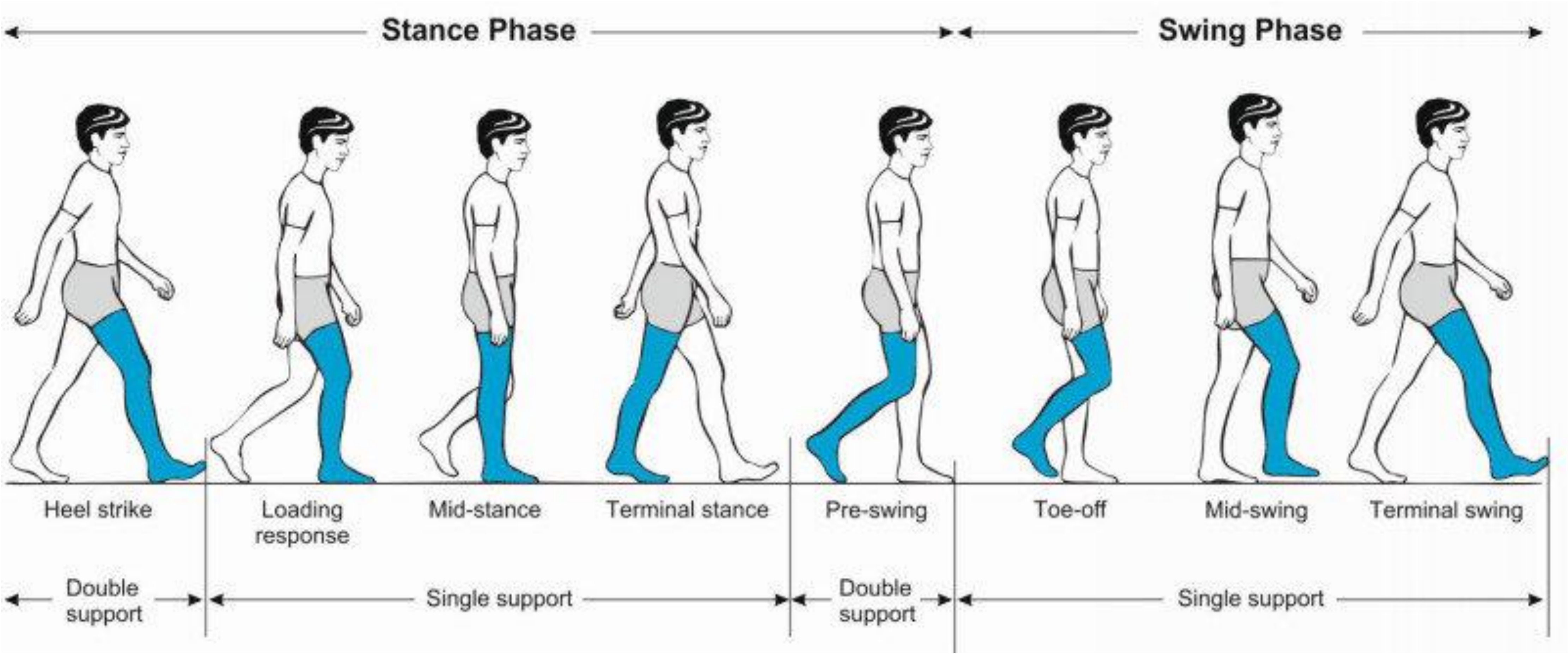
Implementation of ANN in Regression

Prediction the knee joint angle activation base on foot pressure data

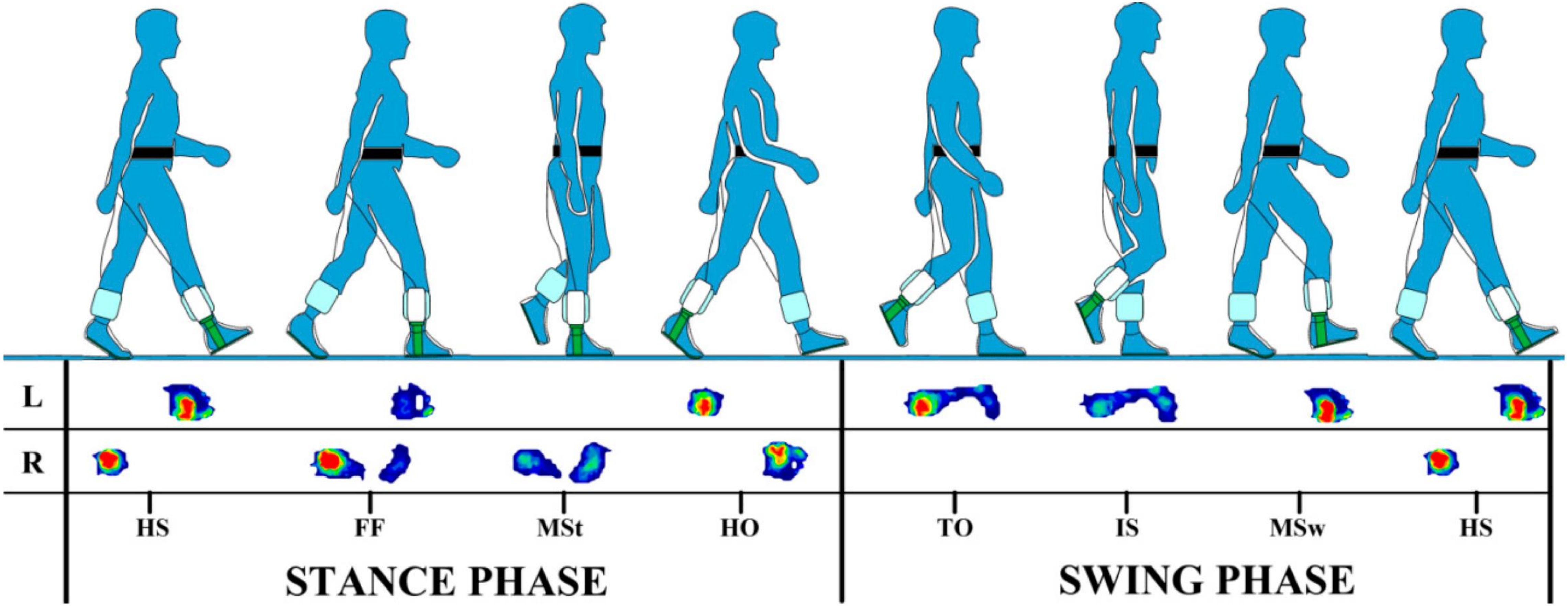
Check Dataset Pattern:

- A gait cycle
- Foot plantar pressure
- Knee joint angle

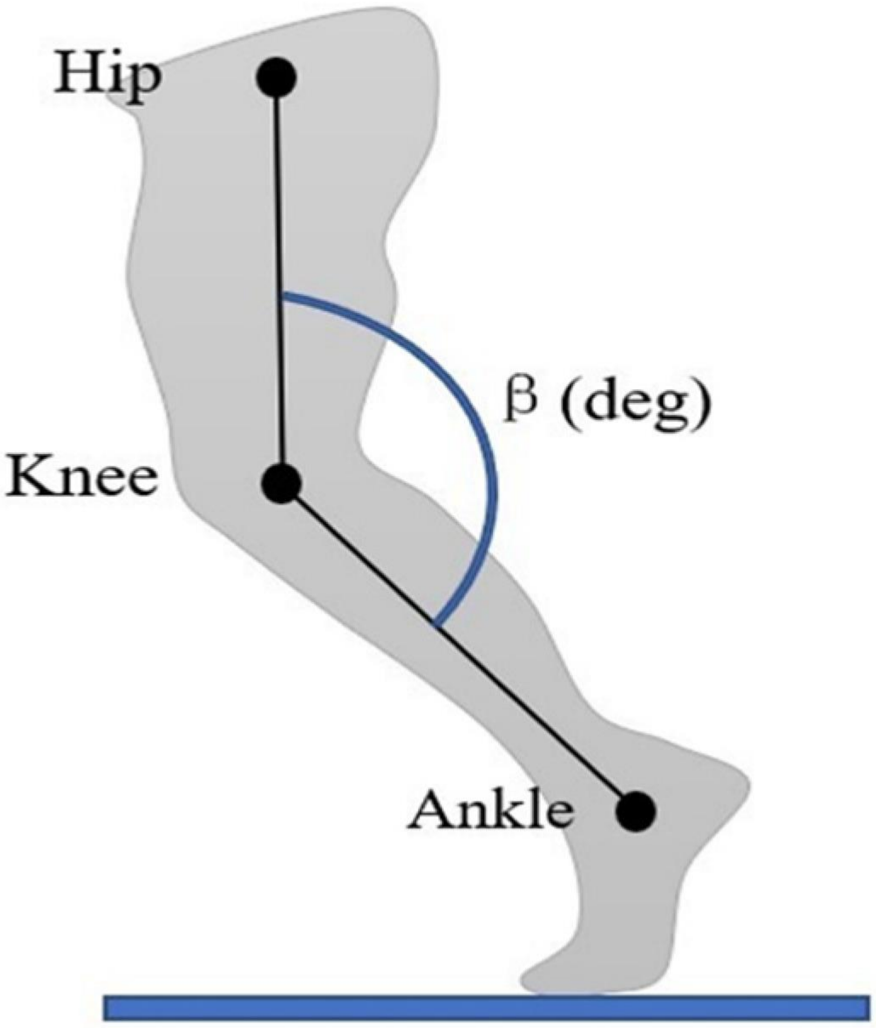
A Gait Cycle


Stance Phase
Swing Phase
Heel strike
Loading response
Mid-stance
Terminal stance
Pre-swing
Toe-off
Mid-swing
Terminal swing
Double support
Single support
Double support
Single support

Input Dataset: Foot plantar pressure



Target dataset: Knee joint angle



Neural Net Workflow Steps

- Prepare Data
- Select Hyperparameter
- Define Model
- Identify Tracked Values
- Train and Validate Model
- Visualization and Evaluation

- Data Preparation
  - Define batch size
  - Split train/val/test sets
  - Migrate to Tensors
  - Additional pre-processing (normalization, encoding, etc.)
  - One-hot encoding?

Neural Net Workflow Steps

- Prepare Data
- Select Hyperparameter
- Define Model
- Identify Tracked Values
- Train and Validate Model
- Visualization and Evaluation

- Hyperparameter Selection
  - Network size and type
  - Learning Rate
  - Regularizers and strength
  - Define loss function and optimizer
  - Other hyperparameters

Neural Net Workflow Steps

- Prepare Data
- Select Hyperparameter
- Define Model
- Identify Tracked Values
- Train and Validate Model
- Visualization and Evaluation

- Model Definition
  - Network Type
  - Network Parameters/Layers
  - Output value(s) and dimensions
  - Forward() Function

Thanks!