Neural Network and Deep Learning

Adaline

YANIKA KONGSOROT

Outline

- Adaline
- Adaline Learning Algorithm

Adaline

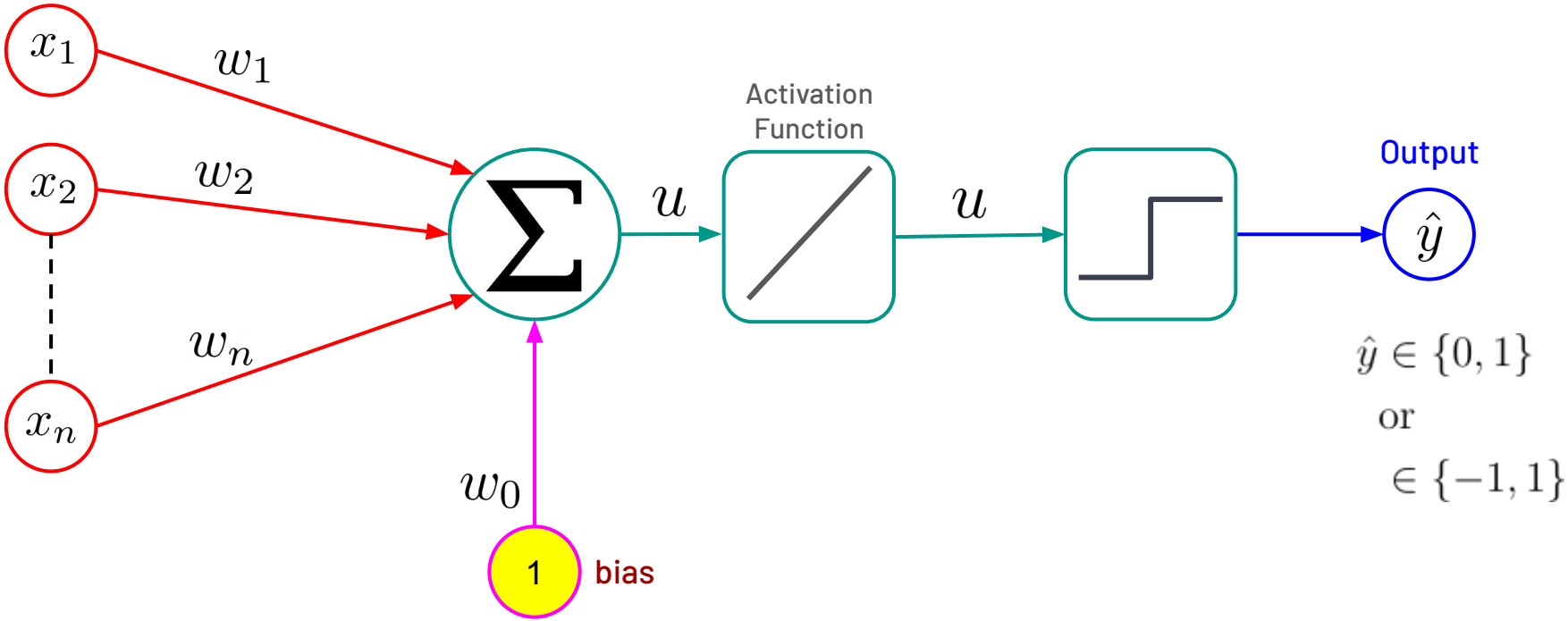
Adaptive **Li**near **Ne**uron

Adaptive **Lin**ear **E**lement

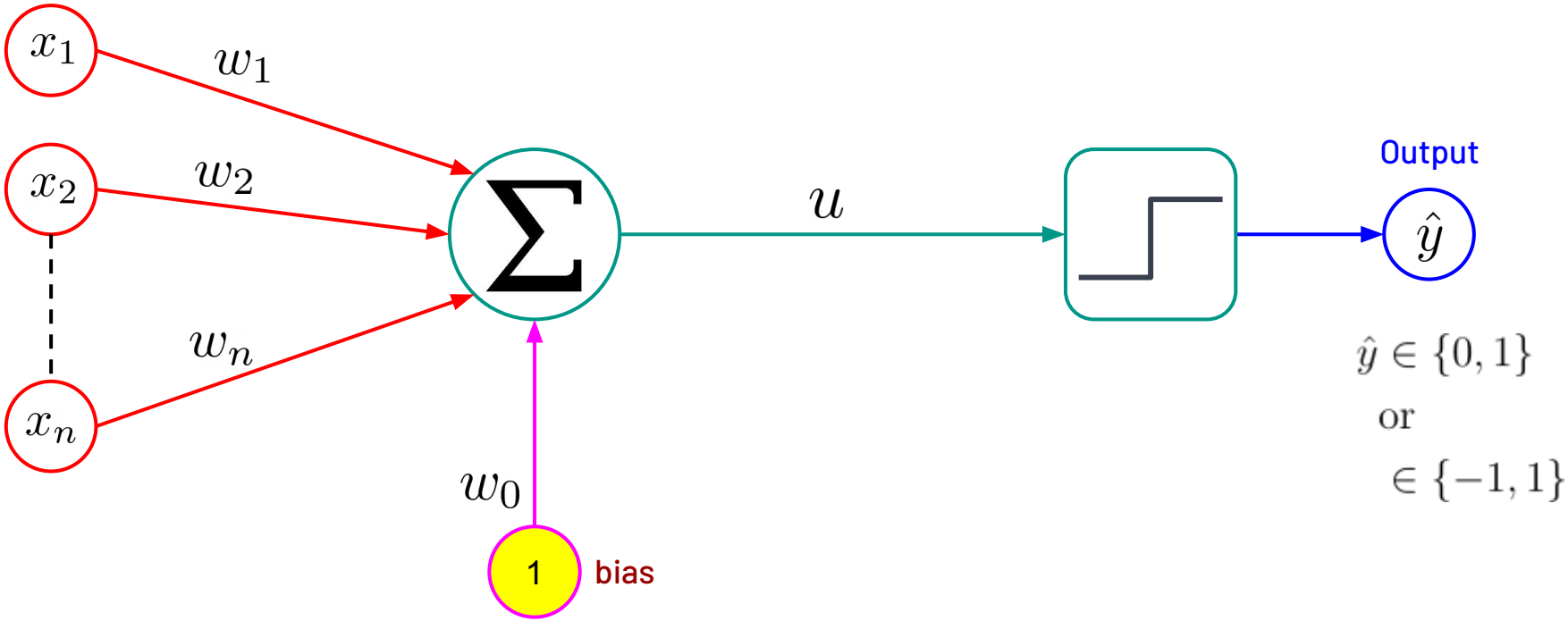
Adaline

- Adaline was proposed by **Bernard Widrow** and **Ted Hoff** in 1960.
- Adaline was developed based on the McCulloch-Pitts neuron.
- In the Adaline learning, the sum of product (wx) is passed to the *linear activation function* $\sigma(u)$ that is then used to make a prediction using a step function (threshold) as with the perceptron.
- A *key* difference with the perceptron is that
    - The *linear activation function* is used for learning the weights, whilst
    - The step function (threshold) is only used for making the prediction at the end.
    - The linear activation function is *differentiable* whilst the step function is not!

Adaline


x_1
w_1
x_2
w_2
w_n
x_n
w_0
1
bias
$\sum$
u
Activation Function
u
Output
$\hat{y}$
$\hat{y} \in \{0, 1\}$
or
$\in \{-1, 1\}$

Adaline



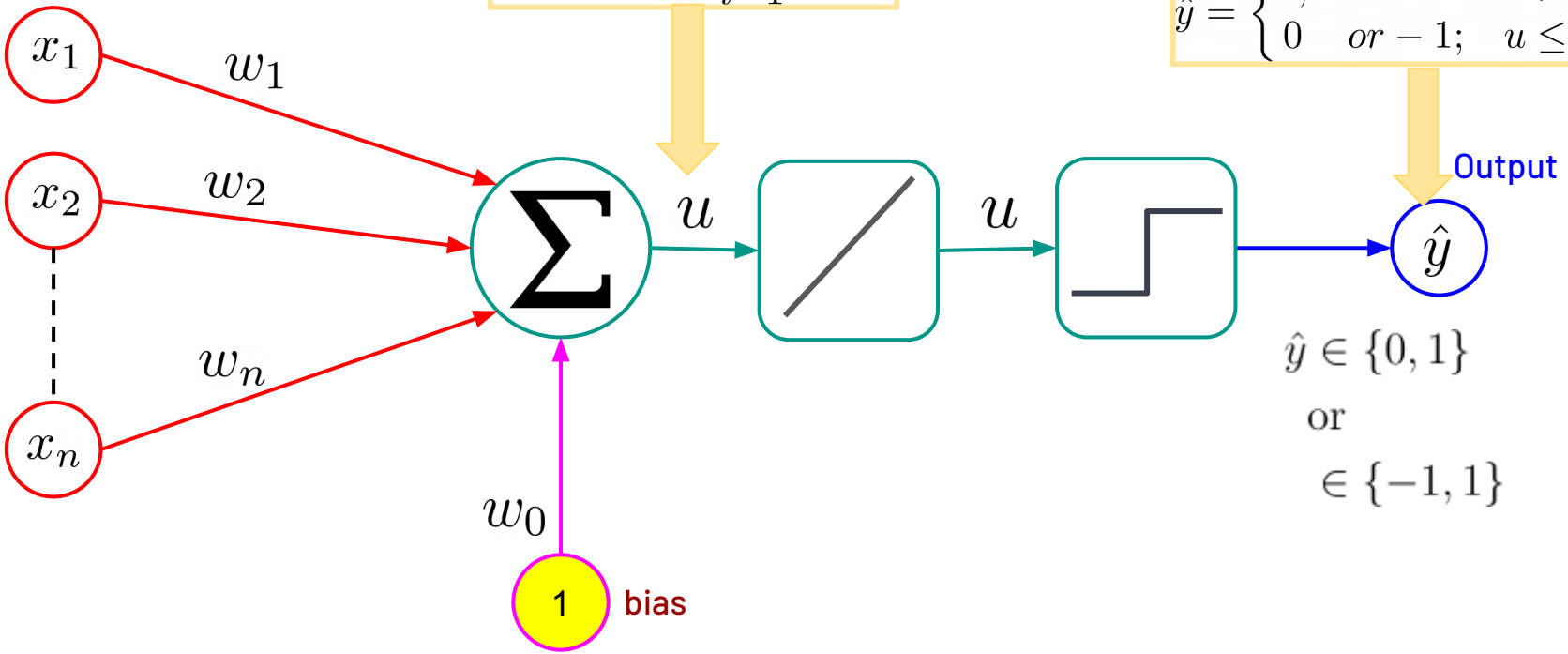
Adaline Learning Algorithm

Adaline

$$u = w_0 + \sum_{i=1}^{n} w_i x_i$$

threshold = 0

$$\hat{y} = \begin{cases} 1; & u > theshold \\ 0 \quad or -1; & u \leq theshold \end{cases}$$

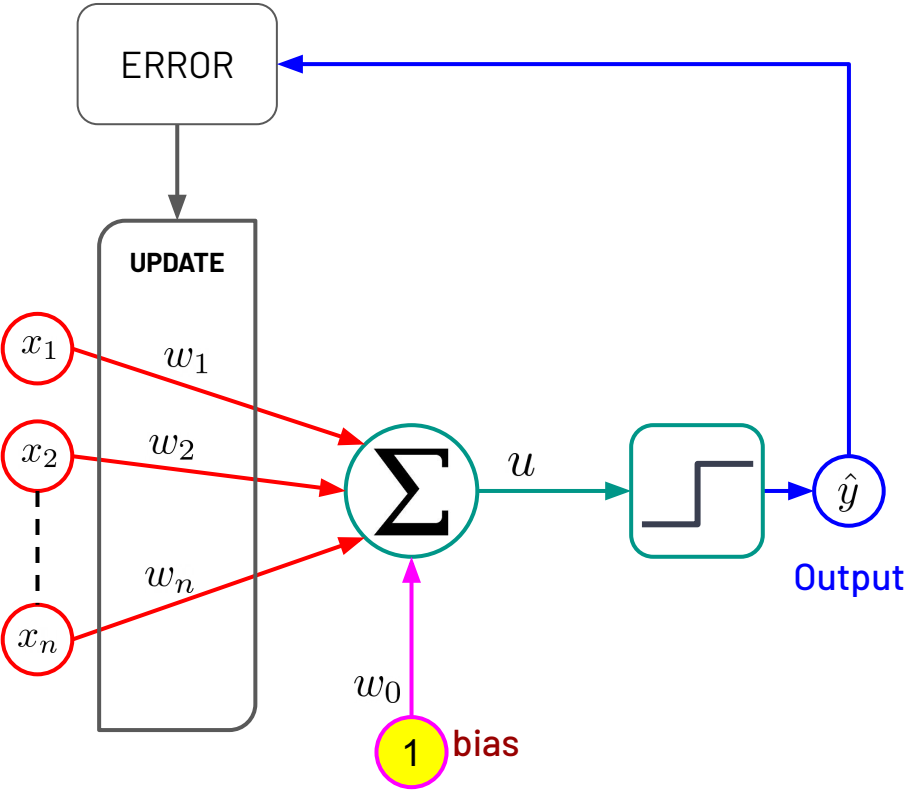


$\hat{y} \in \{0, 1\}$

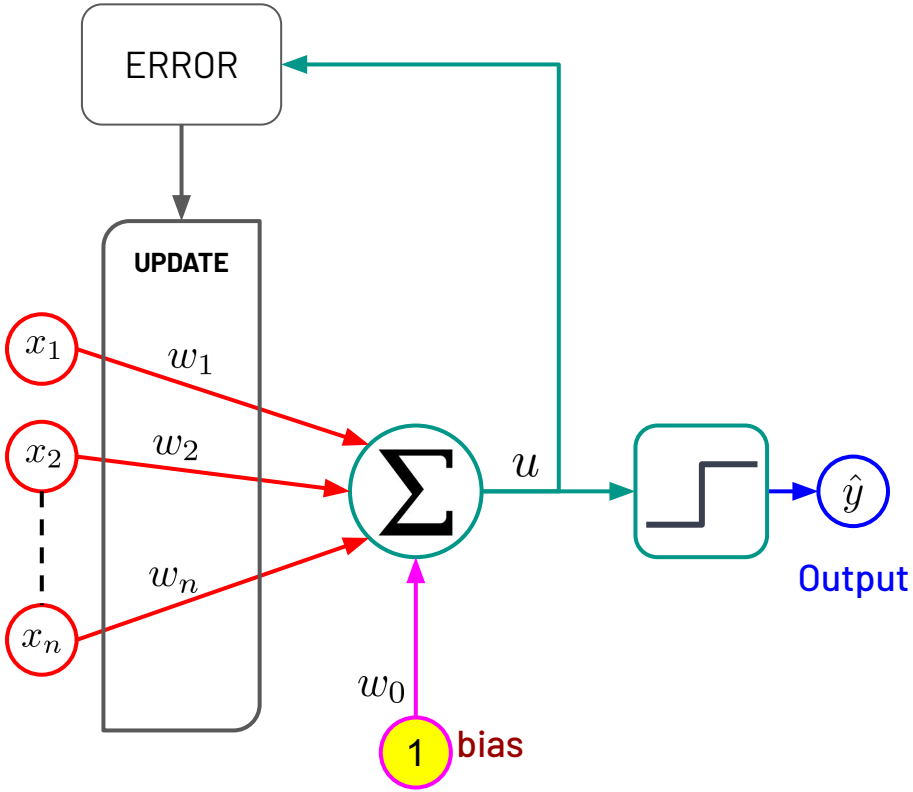
or

$\in \{-1, 1\}$

Perceptron **vs** Adaline



Perceptron



Adaline

Adaline Learning

- In Adaline Learning, when the **learning rate** η is defined, the weights are adjusted as

$$w_i(t+1) = w_i(t) - \eta(2(u-y))x_i$$

- where $0 < \eta \leq 1$

Adaline Learning Algorithm

Step 1:

- Initially, **random** the weights with small value
- Define the value of **learning rate** $\eta = (0, 1]$
- Define the stopping criteria i.e. *number of round*

Step 2:

- Check the stopping criteria
  - If meet the criteria, then stop
  - If far from the criteria, go to **step 3**

Adaline Learning Algorithm

Step 3: Train model

- For each data point ($\mathbf{x}$)
  - **Step 3.1:** Calculate **sum-of-product** between input and weight $$u = w_0 + \sum_{i=1}^{n} w_i x_i$$
  - **Step 3.2:** Calculate the **output** of model $$\hat{y} = \begin{cases} 1; & u > theshold \\ 0 \quad or -1; & u \leq theshold \end{cases}$$
  - **Step 3.3: Update Weights** $$w_i(t+1) = w_i(t) - \eta(2(u-y))x_i$$

Step 4:

- Go to **step 2**

Hands On