Neural Network and Deep Learning

Perceptron Learning

YANIKA KONGSOROT

Outline

- Learning rule
- Perceptron Learning Rule
- Perceptron Learning Algorithm

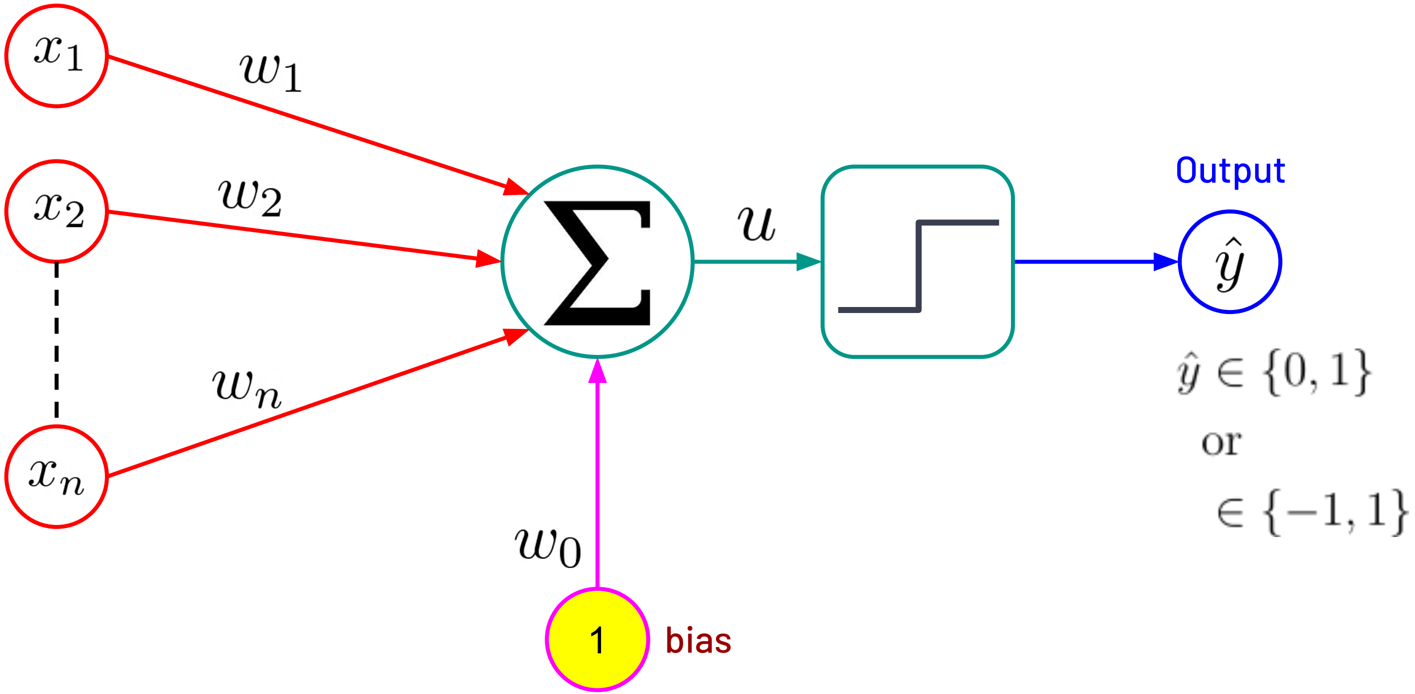
Learning rule

Learning rule

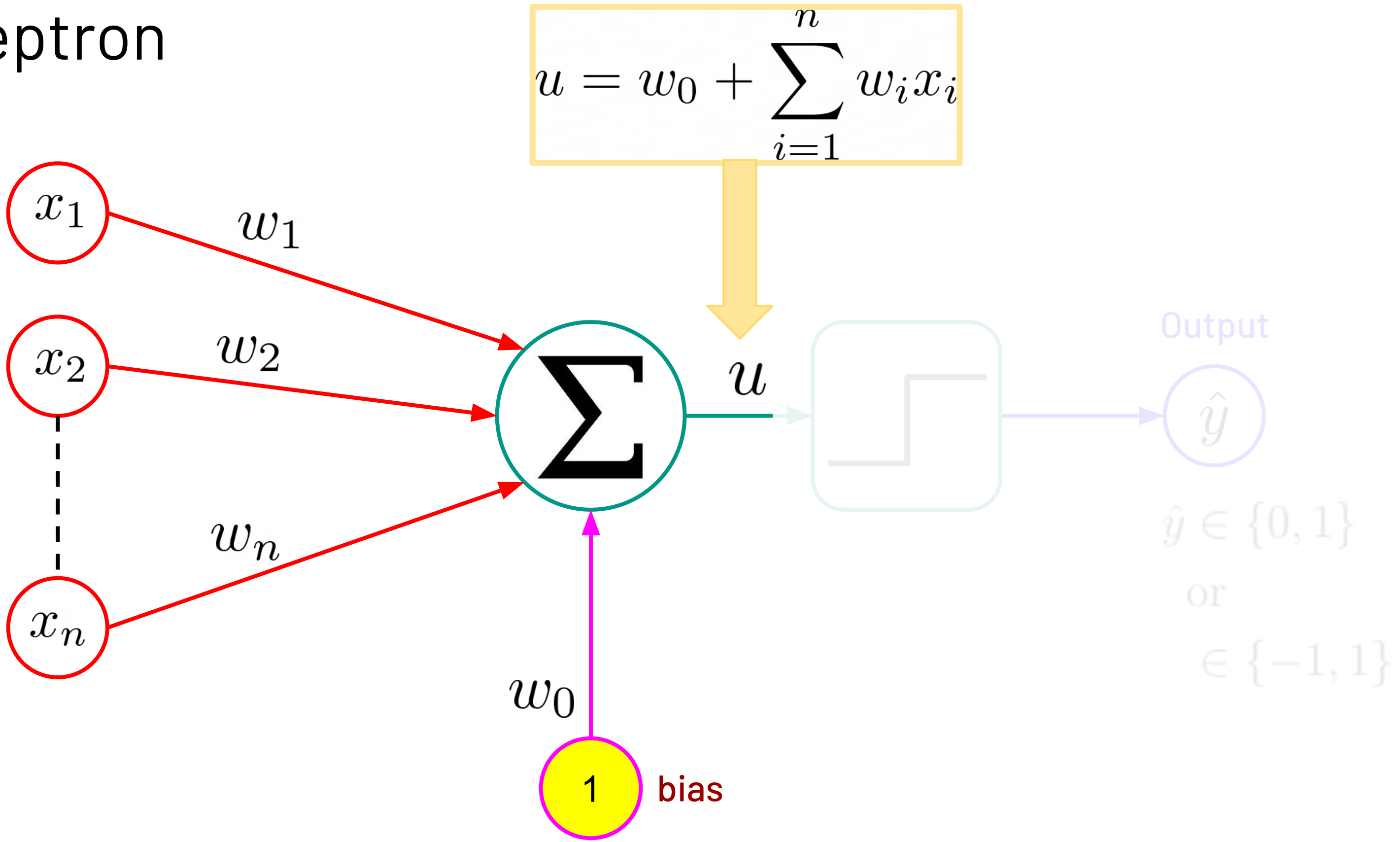
- Neural networks learn by mimicking the human brain's learning process, which is capable of ***adapting and changing behavior*** *in response to environmental stimuli*.
- Neural networks learning rules can be defined as the algorithm called ***Learning Algorithm***.

Perceptron Learning Rule

Perceptron

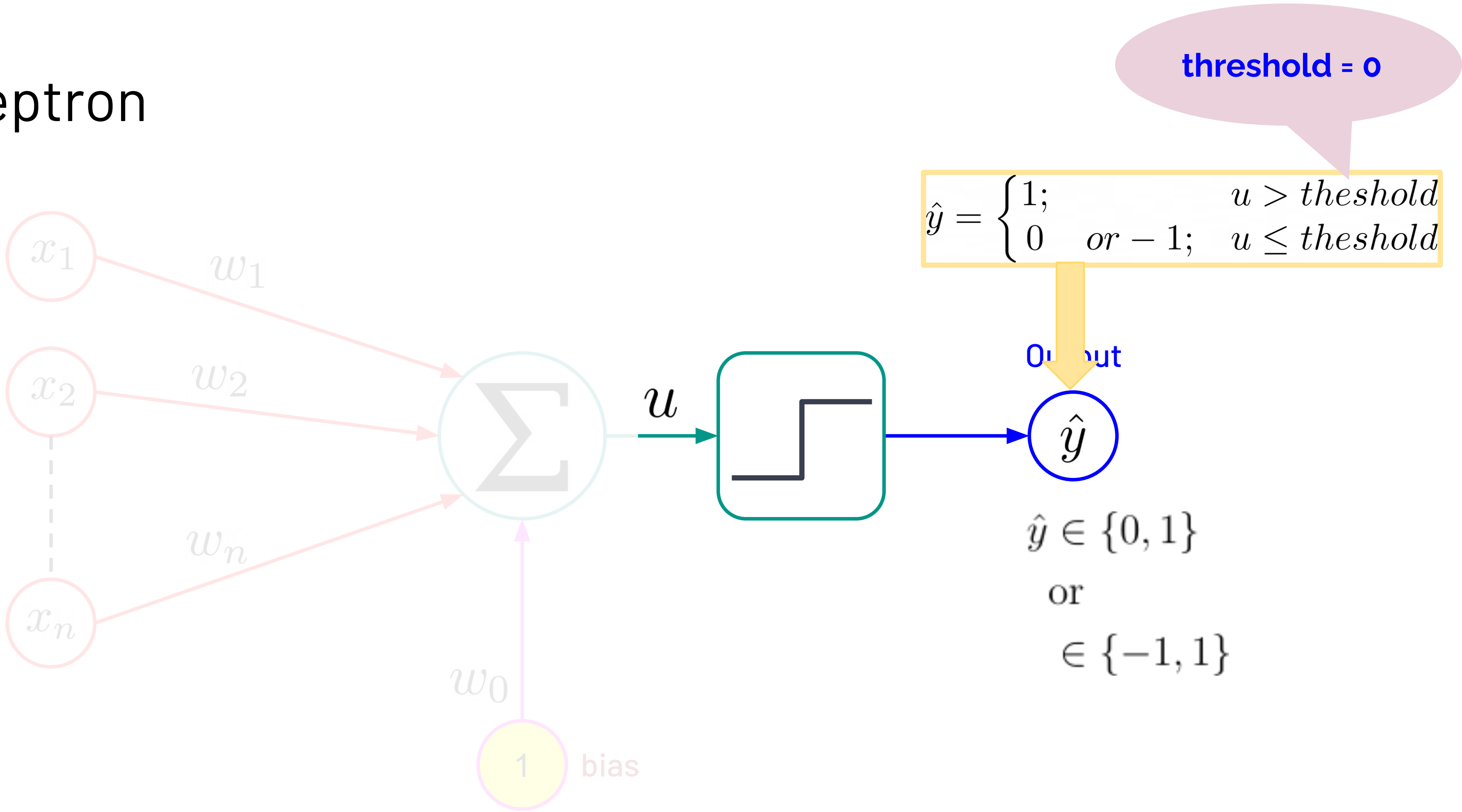


Perceptron



$u = w_0 + \sum_{i=1}^{n} w_i x_i$
x_1
x_2
x_n
w_1
w_2
w_n
Σ
u
w_0
1
bias
Output
$\hat{y}$
$\hat{y} \in \{0, 1\}$
or
$\in \{-1, 1\}$

Perceptron


threshold = 0
$\hat{y} = \begin{cases} 1; & u > theshold \\ 0 \quad or -1; & u \leq theshold \end{cases}$
Output
x_1
x_2
x_n
w_1
w_2
w_n
w_0
Σ
u
$\hat{y}$
1
bias
$\hat{y} \in \{0, 1\}$
or
$\in \{-1, 1\}$

Perceptron Learning Rule

- **Frank Rosenblatt** published the first concept of the **perceptron learning rule** based on the McCulloch-Pitts neuron model in 1957
- In the perceptron learning rule, **weight adjustments** are made through iterative operations.
- According to ***convergence theory***, every round, the weights of the perceptron must be adjusted to the proper weights; that is, the adjusted weights should produce an output that is as close to the actual value as possible.

Perceptron Learning Rule

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

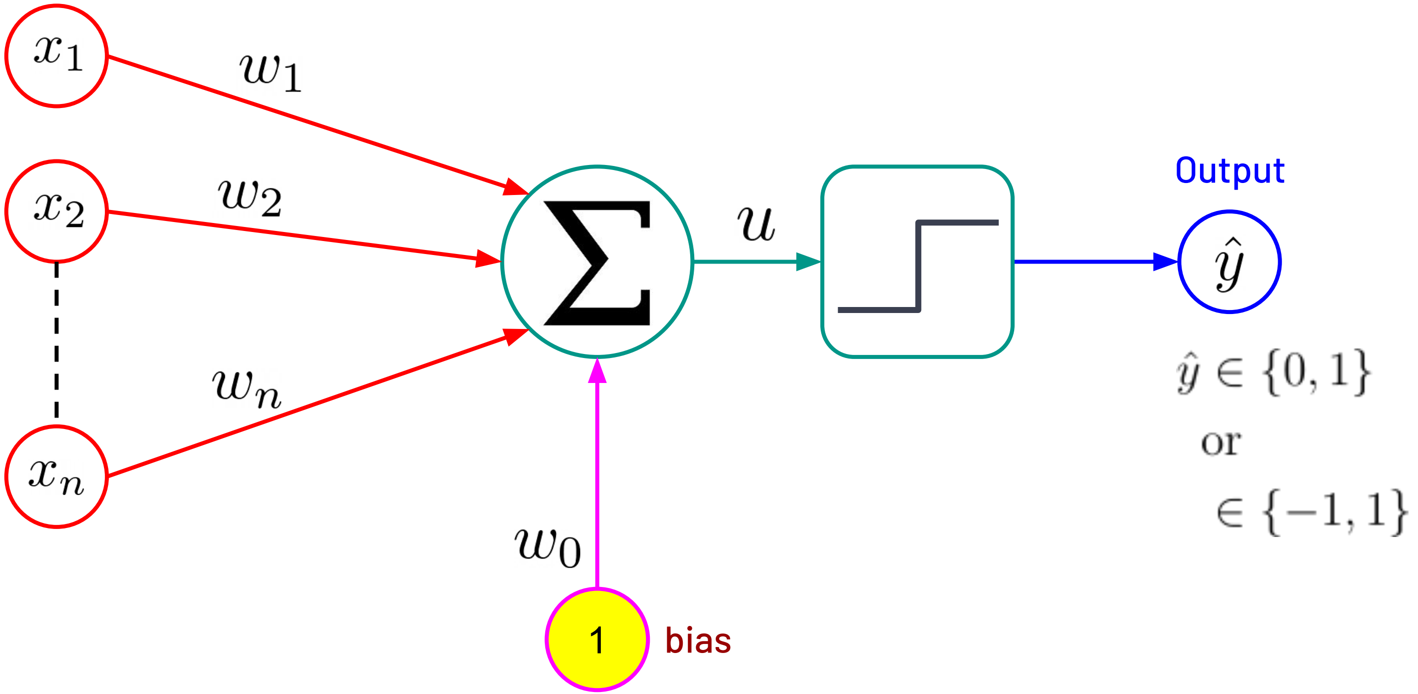
$$\boxed{w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i}$$

- where $0 < \eta \leq 1$

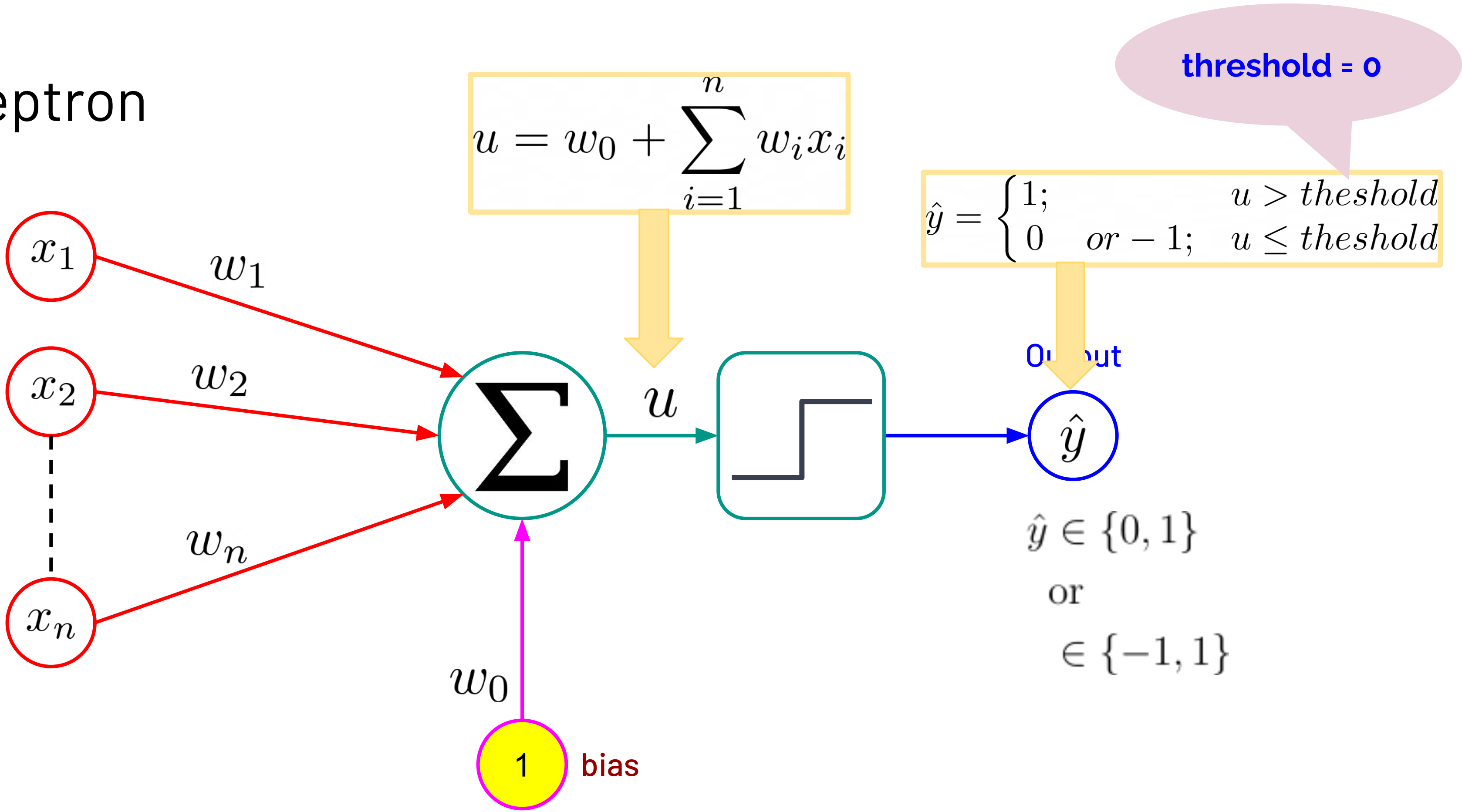
- According to **convergence theory**, **if the input vector and target values *can be linearly separated***, then when using the perceptron learning algorithm, **the weights should be obtained within a finite number of rounds**.

Perceptron Learning Algorithm

Perceptron



Perceptron



threshold = 0
$u = w_0 + \sum_{i=1}^{n} w_i x_i$
$\hat{y} = \begin{cases} 1; & u > theshold \\ 0 \quad or -1; & u \leq theshold \end{cases}$
x_1
x_2
x_n
w_1
w_2
w_n
Σ
u
$\hat{y}$
$\hat{y} \in \{0, 1\}$
or
$\in \{-1, 1\}$
w_0
1
bias

Perceptron 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**

Perceptron 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** $\boxed{w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i}$

Step 4:

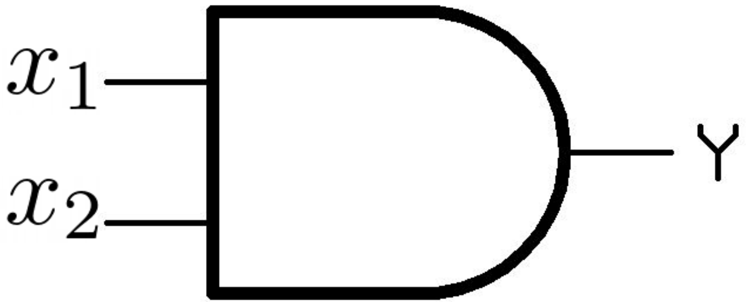
- Go to **step 2**

Perceptron Learning
Computation

Weights adjustment when emulate the behavior of logic gate.

AND Gate via Perceptron

x_1	x_2	y
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1



AND Gate via Perceptron

x_1	x_2	y
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

Step 1:

- Initially, set the initial weights

  $$w_0 = 0, w_1 = 0, w_2 = 0$$

- Define the value of learning rate

  $$\eta = 1$$

- Define the threshold value = 0

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

- Define the stopping criteria
  - number of round = 1

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	Step 3.1 $u = w_0 + \sum_{i=1}^{n} w_i x_i$	Step 3.2 $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	Step 3.3 $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	$\overset{w_0}{0} + (\overset{w_1}{0} * \quad) + (\overset{w_2}{0} * \quad)$				

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	**Step 3.3** $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	$\overset{w_0}{\mathbf{0}} + (\overset{w_1}{0} * 1) + (\overset{w_2}{0} * 1) = \mathbf{0}$				

AND Gate via Perceptron

Round 1 (Train model)

			Step 3.1	Step 3.2	Step 3.3 $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$		
x_1	x_2	y	$u = w_0 + \sum_{i=1}^{n} w_i x_i$	$\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = **0**	-1			

AND Gate via Perceptron

Round 1 (Train model)

			Step 3.1	**Step 3.2**	**Step 3.3** $\boxed{w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i}$		
x_1	x_2	y	$u = w_0 + \sum_{i=1}^{n} w_i x_i$	$\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	w_0	w_1	w_2
1	1	1	**0 + (0*1) + (0*1) = 0**	-1	$w_i(t) + \eta(y - \hat{y})x_i$	$w_i(t) + \eta(y - \hat{y})x_i$	$w_i(t) + \eta(y - \hat{y})x_i$

AND Gate via Perceptron

Round 1 (Train model)

			Step 3.1	**Step 3.2**	**Step 3.3**		
					$w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$		
x_1	x_2	y	$u = w_0 + \sum_{i=1}^{n} w_i x_i$	$\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	w_0	w_1	w_2
1	1	1	0 + (0*1) + (0*1) = 0	-1	$0 + \overset{\eta}{1} * (1 - (-1)) * 1$	$0 + \overset{\eta}{1} * (1 - (-1)) * 1$	$0 + \overset{\eta}{1} * (1 - (-1)) * 1$

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \le 0 \end{cases}$	**Step 3.3** $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	0 + (0*1) + (0*1) = 0	-1	Bias 0+ 1* (1 - (-1)) *1	0+ 1* (1 - (-1)) *1	0+ 1* (1 - (-1)) *1

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \le 0 \end{cases}$	**Step 3.3** $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	Step 3.1 $u = w_0 + \sum_{i=1}^{n} w_i x_i$	Step 3.2 $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	Step 3.3 $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	$\overset{w_0}{\mathbf{2}}$ + ($\overset{w_1}{\mathbf{2}}$*) + ($\overset{w_2}{\mathbf{2}}$*)				

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	**Step 3.3** $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	w_0 w_1 w_2 **2** + (**2***1) + (**2***-1) **= 2**				

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	**Step 3.3** $\boxed{w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i}$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = **2**	**1**			

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \le 0 \end{cases}$	**Step 3.3** $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = **2**	**1**	$w_i(t) + \eta(y - \hat{y})x_i$	$w_i(t) + \eta(y - \hat{y})x_i$	$w_i(t) + \eta(y - \hat{y})x_i$

AND Gate via Perceptron

Round 1 (Train model)

			Step 3.1	Step 3.2	Step 3.3		
					$w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$		
x_1	x_2	y	$u = w_0 + \sum_{i=1}^{n} w_i x_i$	$\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = **2**	**1**	η **2**+ 1* (-1 - 1) *1	η **2**+ 1* (-1 - 1) *1	η **2**+ 1* (-1 - 1) *-1

AND Gate via Perceptron

Round 1 (Train model)

			Step 3.1	Step 3.2	Step 3.3		
					$w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$		
x_1	x_2	y	$u = w_0 + \sum_{i=1}^{n} w_i x_i$	$\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	w_0	w_1	w_2
1	1	1	0 + (0*1) + (0*1) = 0	-1	0+ 1*(1-(-1))*1 = 2	0+ 1*(1-(-1))*1 = 2	0+ 1*(1-(-1))*1 = 2
1	-1	-1	2 + (2*1) + (2*-1) = 2	1	Bias 2+ 1* (-1 - 1) *1	2+ 1* (-1 - 1) *1	2+ 1* (-1 - 1) *-1

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	**Step 3.3** $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**

AND Gate via Perceptron

Round 1 (Train model)

			Step 3.1	**Step 3.2**	**Step 3.3**		
					$w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$		
x_1	x_2	y	$u = w_0 + \sum_{i=1}^{n} w_i x_i$	$\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	w_0 w_1 w_2 **0** + (**0***) + (**4***)				

AND Gate via Perceptron

Round 1 (Train model)

			Step 3.1	**Step 3.2**	**Step 3.3**		
					$w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$		
x_1	x_2	y	$u = w_0 + \sum_{i=1}^{n} w_i x_i$	$\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	w_0 w_1 w_2 **0** + (**0*** **1**) + (**4*****1**) **= 4**				

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	Step 3.1 $u = w_0 + \sum_{i=1}^{n} w_i x_i$	Step 3.2 $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	Step 3.3 $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0*** **1**) + (**4*****1**) = **4**	**1**			

AND Gate via Perceptron

Round 1 (Train model)

			Step 3.1	Step 3.2	Step 3.3		
					$w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$		
x_1	x_2	y	$u = w_0 + \sum_{i=1}^{n} w_i x_i$	$\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \le 0 \end{cases}$	w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0*** **1**) + (**4*****1**) **= 4**	**1**	$w_i(t) + \eta(y - \hat{y})x_i$	$w_i(t) + \eta(y - \hat{y})x_i$	$w_i(t) + \eta(y - \hat{y})x_i$

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	Step 3.1 $u = w_0 + \sum_{i=1}^{n} w_i x_i$	Step 3.2 $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \le 0 \end{cases}$	Step 3.3 $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0*** **1**) + (**4***1) **= 4**	**1**	η **0**+ 1* **(-1 - 1)** *1	η **0**+ 1* **(-1 - 1)** *-1	η **4**+ 1* **(-1 - 1)** *1

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	**Step 3.3** $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0*** **1**) + (**4*****1**) **= 4**	**1**	Bias **0**+ 1* (-1 - 1) ***1**	**0**+ 1* (-1 - 1) ***-1**	**4**+ 1* (-1 - 1) ***1**

AND Gate via Perceptron

Round 1 (Train model)

			Step 3.1	Step 3.2	Step 3.3		
					$w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$		
x_1	x_2	y	$u = w_0 + \sum_{i=1}^{n} w_i x_i$	$\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0***-1) + (**4***1) = 4	1	**0**+ 1*(-1-1)*1=**-2**	**0**+ 1*(-1-1)*-1 = **2**	**4**+ 1*(-1-1)*1 = **2**

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \le 0 \end{cases}$	**Step 3.3** $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0***-1) + (**4***1) = 4	1	**0**+ 1*(-1-1)*1=**-2**	**0**+ 1*(-1-1)*-1 = **2**	**4**+ 1*(-1-1)*1 = **2**
-1	-1	-1	w_0 w_1 w_2 **-2** + (**2***) + (**2***)				

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	**Step 3.3** $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0***-1) + (**4***1) = 4	1	**0**+ 1*(-1-1)*1=**-2**	**0**+ 1*(-1-1)*-1 = **2**	**4**+ 1*(-1-1)*1 = **2**
-1	-1	-1	$\underset{w_0}{\textbf{-2}} + (\underset{w_1}{\textbf{2}}*\textbf{-1}) + (\underset{w_2}{\textbf{2}}*\textbf{-1})$ **= -6**				

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \le 0 \end{cases}$	**Step 3.3** $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ — w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0***-1) + (**4***1) = 4	1	**0**+ 1*(-1-1)*1=**-2**	**0**+ 1*(-1-1)*-1 = **2**	**4**+ 1*(-1-1)*1 = **2**
-1	-1	-1	**-2** + (**2***-1) + (**2***-1) = **-6**	**-1**			

AND Gate via Perceptron

Round 1 (Train model)

			Step 3.1	Step 3.2	Step 3.3		
					$w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$		
x_1	x_2	y	$u = w_0 + \sum_{i=1}^{n} w_i x_i$	$\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0***-1) + (**4***1) = 4	1	**0**+ 1*(-1-1)*1=**-2**	**0**+ 1*(-1-1)*-1 = **2**	**4**+ 1*(-1-1)*1 = **2**
-1	-1	-1	**-2** + (**2***-1) + (**2***-1) = **-6**	**-1**	$w_i(t) + \eta(y - \hat{y})x_i$	$w_i(t) + \eta(y - \hat{y})x_i$	$w_i(t) + \eta(y - \hat{y})x_i$

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	Step 3.1 $u = w_0 + \sum_{i=1}^{n} w_i x_i$	Step 3.2 $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	Step 3.3 $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0***-1) + (**4***1) = 4	1	**0**+ 1*(-1-1)*1=**-2**	**0**+ 1*(-1-1)*-1 = **2**	**4**+ 1*(-1-1)*1 = **2**
-1	-1	-1	**-2** + (**2***-1) + (**2***-1) = -6	-1	**-2**+ $\overset{\eta}{1}$* (-1 - (-1)) *1	**2**+ $\overset{\eta}{1}$* (-1 - (-1)) *-1	**2**+ $\overset{\eta}{1}$* (-1 - (-1)) *-1

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	**Step 3.1** $u = w_0 + \sum_{i=1}^{n} w_i x_i$	**Step 3.2** $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \le 0 \end{cases}$	**Step 3.3** $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0***-1) + (**4***1) = 4	1	**0**+ 1*(-1-1)*1=**-2**	**0**+ 1*(-1-1)*-1 = **2**	**4**+ 1*(-1-1)*1 = **2**
-1	-1	-1	**-2** + (**2***-1) + (**2***-1) = -6	-1	Bias **-2**+ 1* (-1 - (-1)) ***1**	**2**+ 1* (-1 - (-1)) ***-1**	**2**+ 1* (-1 - (-1)) ***-1**

AND Gate via Perceptron

Round 1 (Train model)

x_1	x_2	y	Step 3.1 $u = w_0 + \sum_{i=1}^{n} w_i x_i$	Step 3.2 $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \le 0 \end{cases}$	Step 3.3 $w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i$ w_0	w_1	w_2
1	1	1	**0** + (**0***1) + (**0***1) = 0	-1	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**	**0**+ 1*(1-(-1))*1 = **2**
1	-1	-1	**2** + (**2***1) + (**2***-1) = 2	1	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*1 = **0**	**2**+ 1*(-1-1)*-1 = **4**
-1	1	-1	**0** + (**0***-1) + (**4***1) = 4	1	**0**+ 1*(-1-1)*1=**-2**	**0**+ 1*(-1-1)*-1 = **2**	**4**+ 1*(-1-1)*1 = **2**
-1	-1	-1	**-2** + (**2***-1) + (**2***-1) = -6	-1	**-2**+ 1*(-1-(-1))*1 = **-2**	**2**+ 1*(-1-(-1))*-1 = **2**	**2**+ 1*(-1-(-1))*-1 = **2**

AND Gate via Perceptron

- After finish round 1, we obtained
  - $w_0 = -2 \qquad w_1 = 2 \qquad w_2 = -2$

- According to linear equation, we can get a linear line from our weights
  - $$w_0 + w_1 x_1 + w_2 x_2 = 0$$
    $$-2 + 2x_1 + 2x_2 = 0$$

Discuss !!

If the round number is **2**, how to continue?

AND Gate via Perceptron

Round 2 (Train model)

x_1	x_2	y	Step 3.1 $u = w_0 + \sum_{i=1}^{n} w_i x_i$	Step 3.2 $\hat{y} = \begin{cases} 1; & u > 0 \\ -1; & u \leq 0 \end{cases}$	Step 3.3 $\boxed{w_i(t+1) = w_i(t) + \eta(y - \hat{y})x_i}$ w_0	w_1	w_2
1	1	1	$\overset{w_0}{-2} + (\overset{w_1}{2} * \;) + (\overset{w_2}{2} * \;)$				

Obtained from round 1

Hands On