Artificial Neural Networks - #1

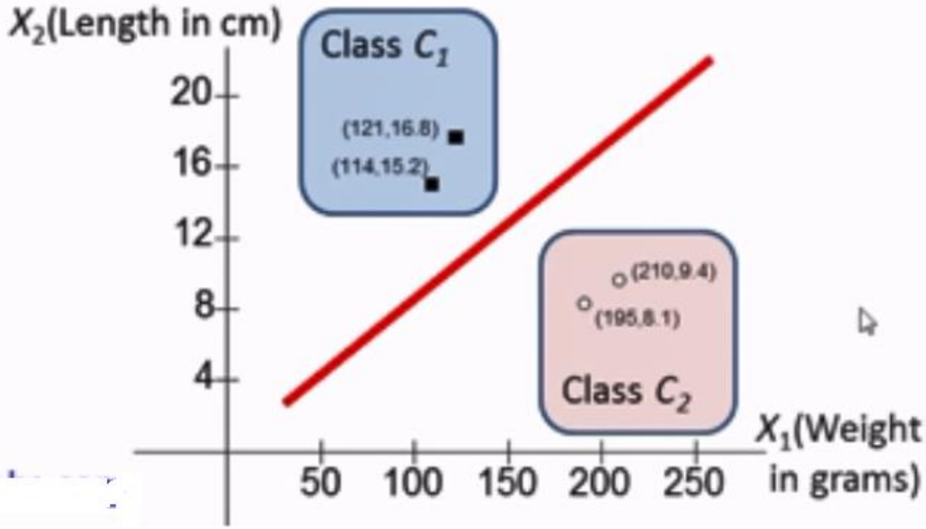
Classification using Single Layer Perceptron Model

- The following table shows sample data obtained from two different fruits.

	Weight (grams)	Length (cm)
Fruit 1 (Class C1)	121	16.8
	114	15.2
Fruit 2 (Class C2)	210	9.4
	195	8.1

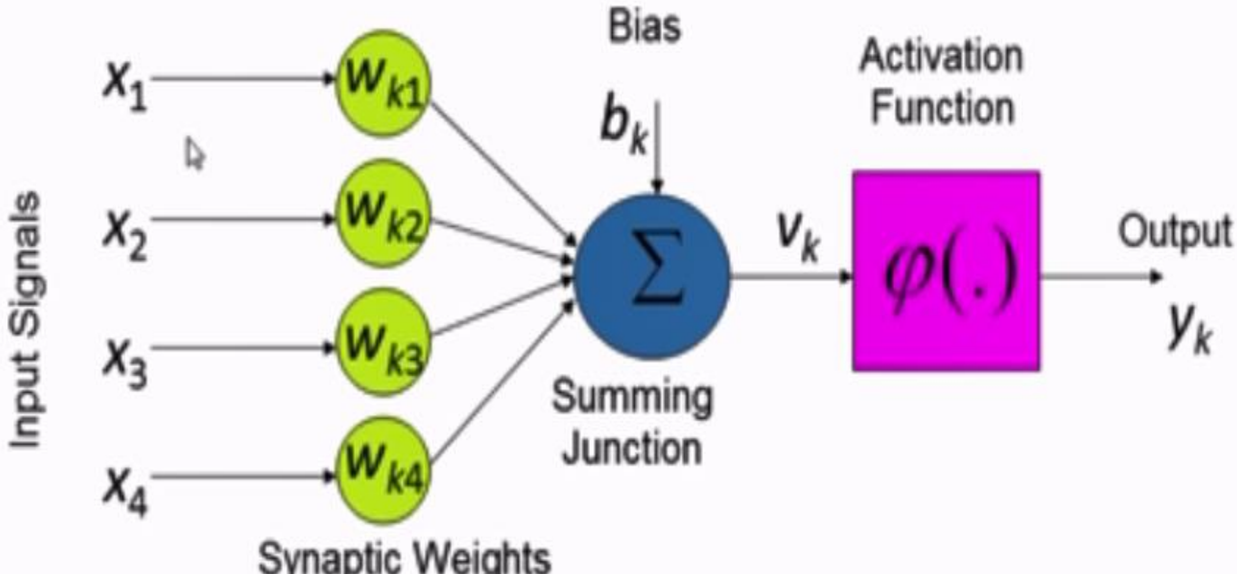
- Train a single layer perceptron model using the above parameters to classify the two fruits.
- Using the model parameters you have obtained classify the fruit with weight 140gm and length 17.9cm.

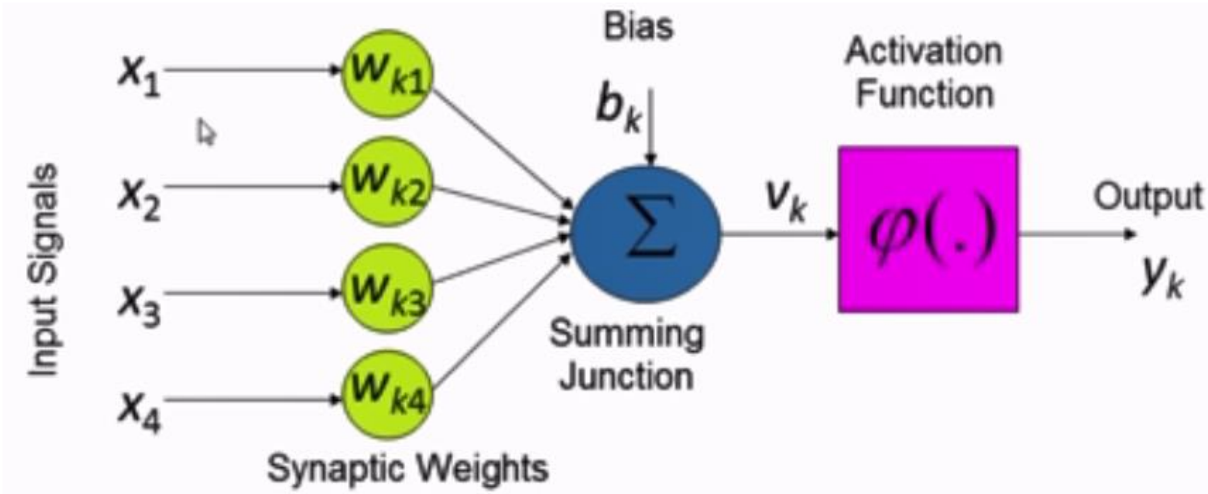
	Weight (grams)	Length (cm)
Fruit 1 (Class C1)	121	16.8
	114	15.2
Fruit 2 (Class C2)	210	9.4
	195	8.1



Model of Neuron

- The model consists of a set of synapses each of which is characterized by a weight or strength of its own.
- An adder, an activation function and a bias.





- In mathematical terms, a neuron k can be described by:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j$$

and

$$y_k = \varphi(u_k + b_k)$$

where u_k is the linear combiner output due to input signals.

- Also $v_k = u_k + b_k$

- The bias is an external parameter of artificial neuron and can be included into the equations as follows:

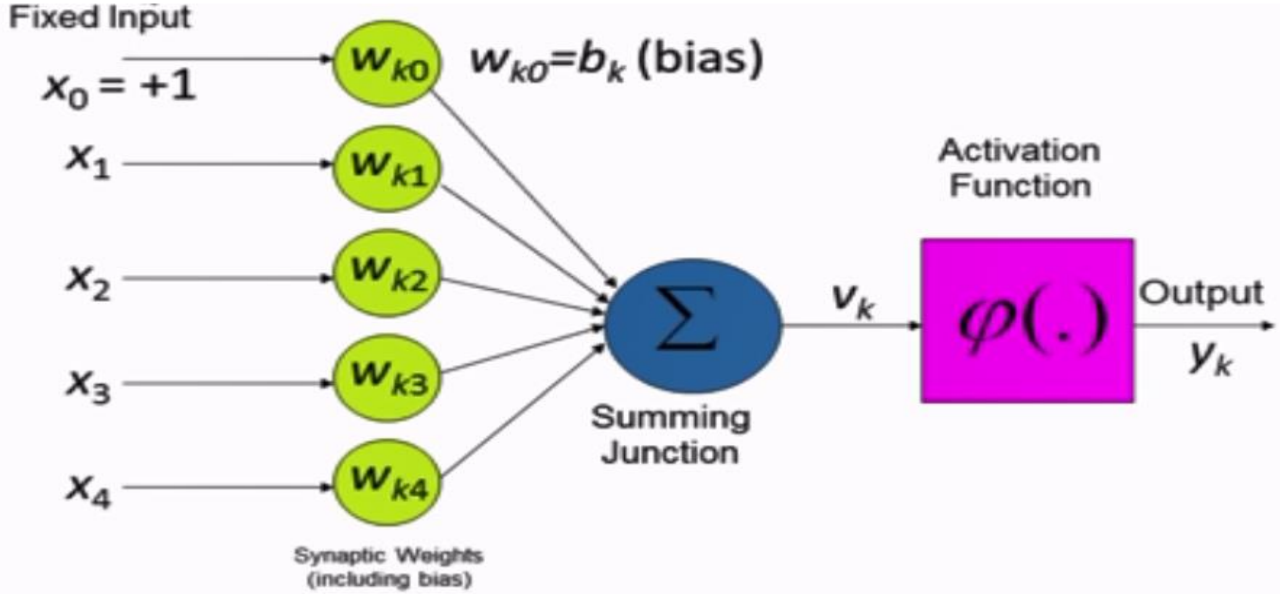
$$v_k = \sum_{j=0}^{m} w_{kj} x_j$$

- and

$$y_k = \varphi(v_k)$$

- Note the change of limits of *j* from 1 to 0.

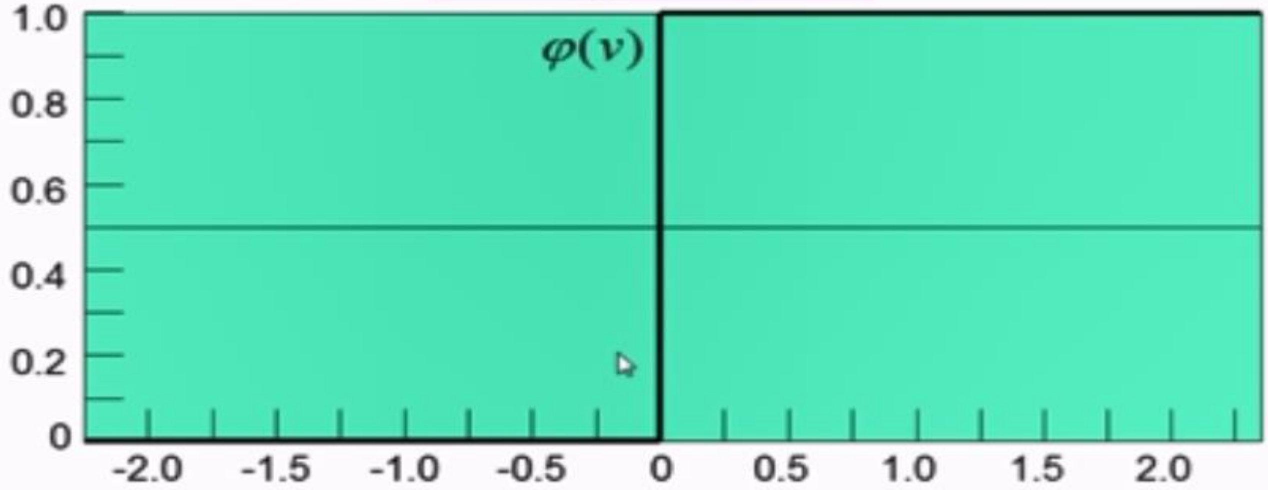
10



Types of Activation Functions

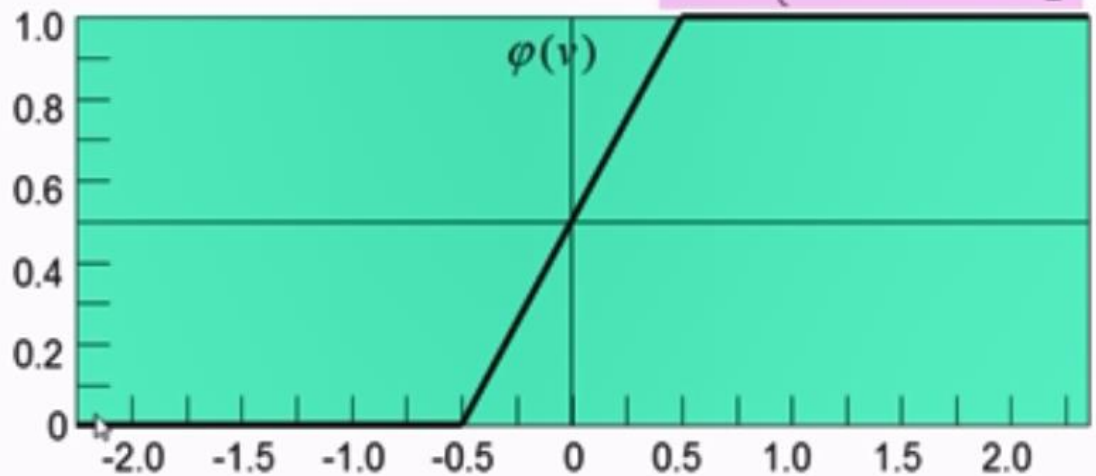
- Threshold Function or Heaviside Function:

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$



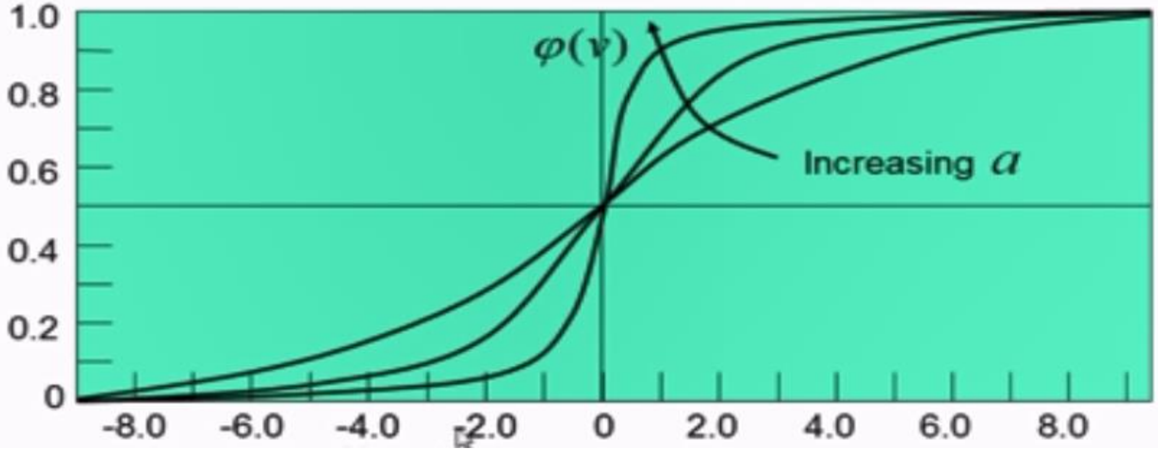
- Piecewise-Linear Function:

$$\varphi(v)=\begin{cases}1, & v \geq +\frac{1}{2} \\ v, & +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leq -\frac{1}{2}\end{cases}$$



- Sigmoid Function: $\varphi(v) = \dfrac{1}{1+\exp(-av)}$

where a is the slope parameter of the sigmoid function.



Single Layer Perceptron

- The neuronal model we have just discussed is also known as a perceptron.
- The perceptron is the simplest form of a neural network used for the classification of patterns said to be linearly separable.
- Basically, it consists of a single neuron with adjustable synaptic weights and bias.
- Now we will look at a method of achieving **learning** in our model we have formulated.

Perceptron Convergence (Learning) Algorithm

- Variables and Parameters

$$\mathbf{x}(n) = (m+1)\times 1 \quad \text{input vector}$$
$$= \left[+1, x_1(n), x_2(n), \ldots, x_m(n)\right]^T$$
$$\mathbf{w}(n) = (m+1)\times 1 \quad \text{weight vector}$$
$$= \left[b(n), w_1(n), w_2(n), \ldots, w_m(n)\right]^T$$
$$b(n) = \text{bias}$$
$$y(n) = \text{actual response}$$
$$d(n) = \text{desired response}$$
$$\eta = \text{learning-rate parameter, a postive constant less than unity}$$

1. ***Initialization***. Set $\mathbf{w}(0) = \mathbf{0}$. Then perform the following computations for time step $n = 1,2, \dots$
2. ***Activation***. At time step n, activate the perceptron by applying input vector $\mathbf{x}(n)$and desired response $d(n)$.
3. ***Computation of Actual Response***. Compute the actual response of the perceptron:

$$y(n) = \mathrm{sgn}[\mathbf{w}^T(n)\mathbf{x}(n)]$$

where $\mathrm{sgn}(.)$ is the signum function.

$$\mathrm{sgn}(x) = \begin{cases} +1, & \text{if} \quad x \geq 0 \\ -1, & \text{if} \quad x < 0 \end{cases}$$

4. ***Adaptation of Weight Vector***. Update the weight vector of the perceptron:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta\left[d(n) - y(n)\right]\mathbf{x}(n)$$

where

$$d(n) = \begin{cases} +1 & \text{if } \mathbf{x}(n) \text{ belongs to class } C_1 \\ -1 & \text{if } \mathbf{x}(n) \text{ belongs to class } C_2 \end{cases}$$

5. ***Continuation***. Increment time step n by one and go back to step 2.

Decision Boundary

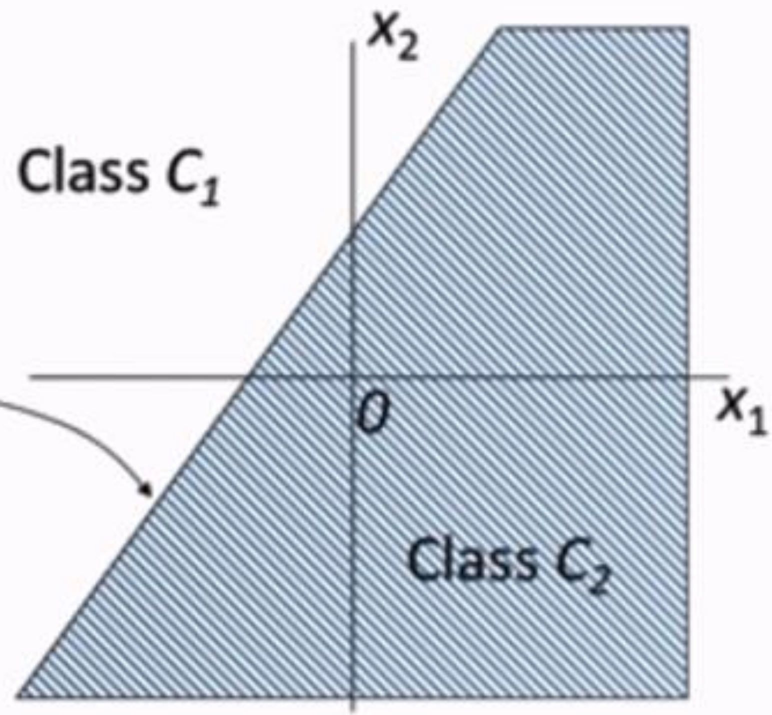
- The hyper-plane

$$\sum_{i=1}^{m} w_i x_i + b = 0$$

or

$$w_1 x_1 + w_2 x_2 + b = 0$$

is the decision boundary for a two class classification problem.



Artificial Neural Networks - #1

Classification using Single Layer Perceptron Model

- The following table shows sample data obtained from two different fruits.

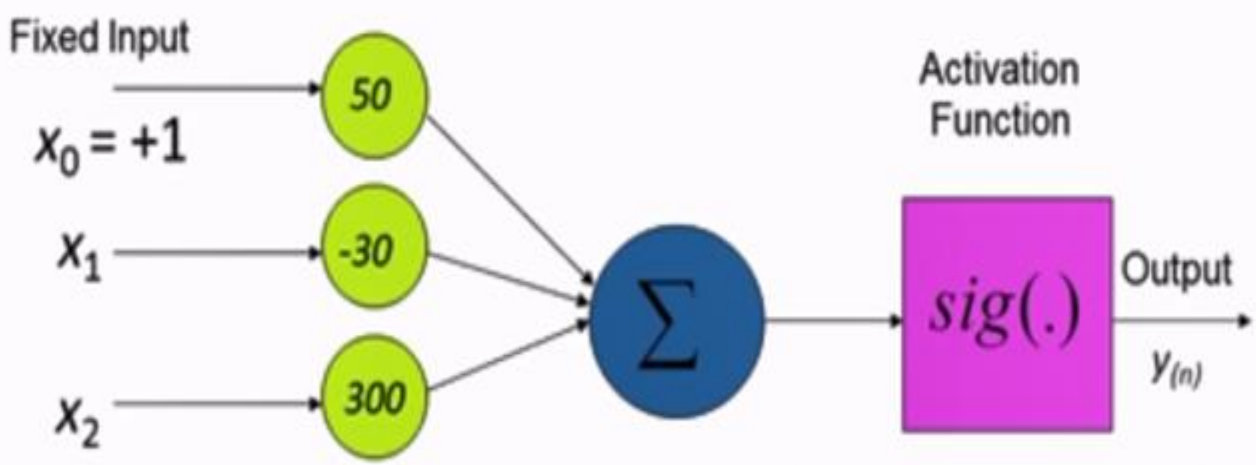
	Weight (grams)	Length (cm)
Fruit 1 (Class C1)	121	16.8
	114	15.2
Fruit 2 (Class C2)	210	9.4
	195	8.1

- Train a single layer perceptron model using the above parameters to classify the two fruits.
- Using the model parameters you have obtained classify the fruit with weight 140gm and length 17.9cm.

Solution to the Example
(with correct initial weights and bias)

With correct initial weights and bias

$$\left.\begin{array}{l} w_1(0) = -30,\ w_2(0) = 300, \\ b(0) = 50,\ \eta = 0.01 \end{array}\right\} given$$



$$\mathrm{sgn}(x) = \begin{cases} +1, & \text{if} \quad x \geq 0 \\ -1, & \text{if} \quad x < 0 \end{cases}$$

$$\left.\begin{array}{l} w_1(0) = -30,\ w_2(0) = 300, \\ b(0) = 50,\ \eta = 0.01 \end{array}\right\} given$$

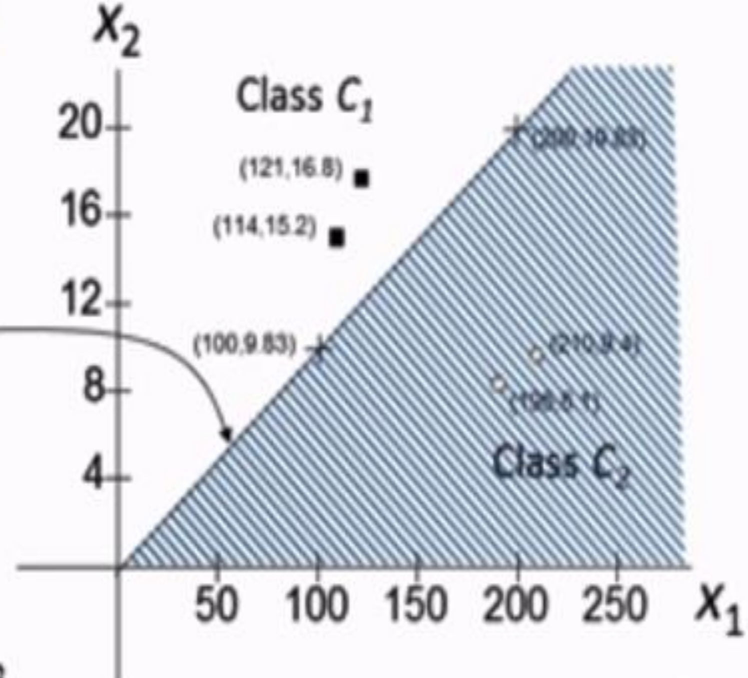
Therefore the Initial Decision Boundary for this example is:

$$\boxed{w_1 x_1 + w_2 x_2 + b = 0}$$

$$-30x_1 + 300x_2 + 50 = 0$$

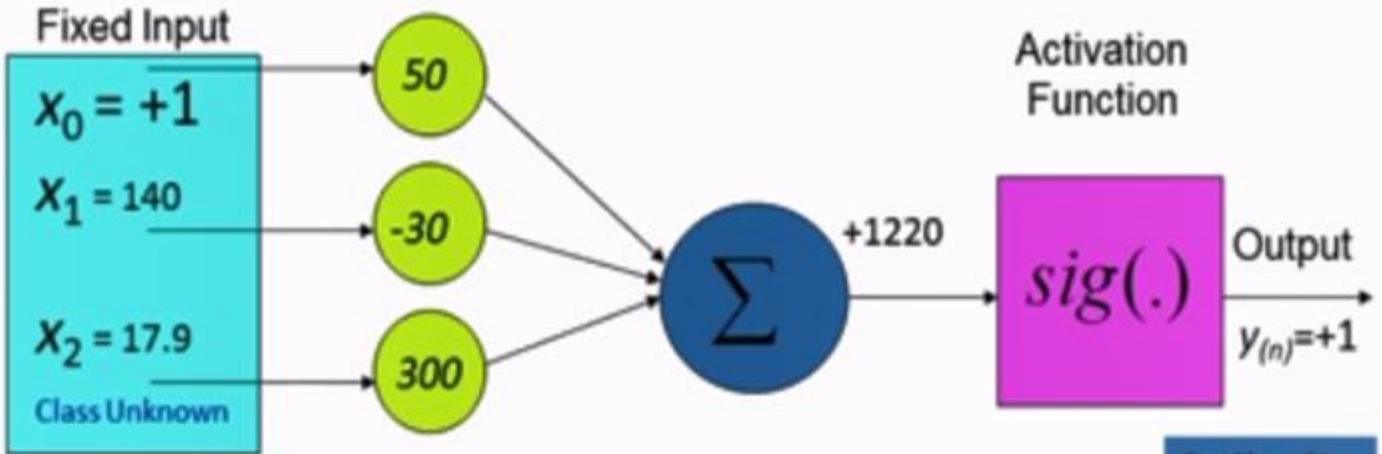
$$x_1 = 100,\ x_2 = \frac{30 \times 100 - 50}{300} = 9.83$$

$$x_1 = 200,\ x_2 = \frac{30 \times 200 - 50}{300} = 19.83$$



Initial hyper-plane does separate the two classes.

Classification of the Unknown Fruit



For Class C1, Output = +1

Now use the above model to classify the unknown fruit.

$$\mathbf{x}(\text{unknown}) = [+1, 140, 17.9]^T$$

$$\mathbf{w}(3) = [50, -30, 300]^T$$

$$y(\text{unknown}) = \text{sgn}\left(\mathbf{w}^T(3)\mathbf{x}(\text{unknown})\right) = \text{sgn}(50\times 1 - 30\times 140 + 300\times 17.9)$$

$$= \text{sgn}(1220) = +1$$

$\therefore$ this unknown fruit belongs to the class C_1.

Solution to the Example

(with unknown initial weights and bias)

With unknown initial weights and bias

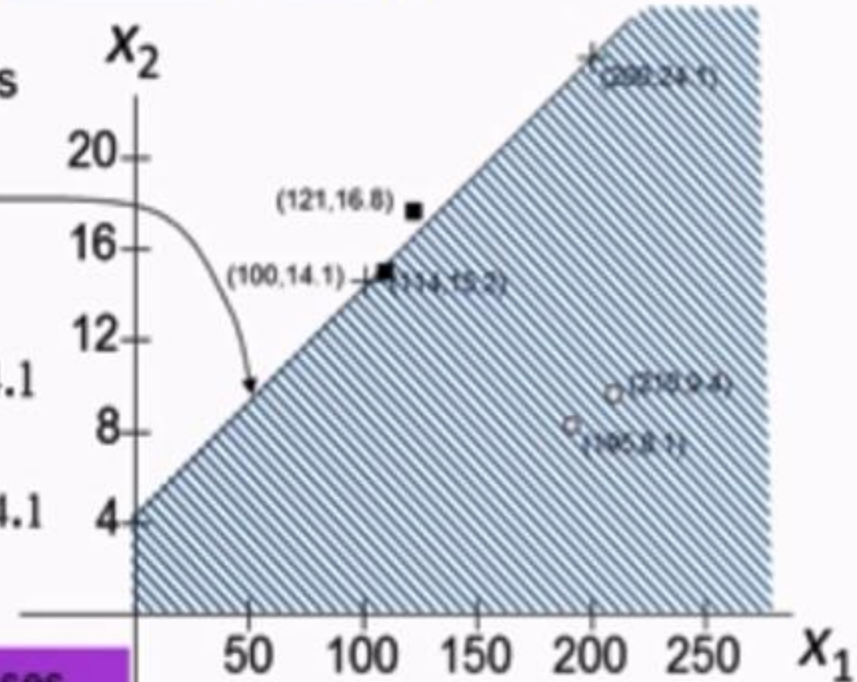
$$\left.\begin{array}{l} w_1(0) = -30,\ w_2(0) = 300, \\ b(0) = -1230,\ \eta = 0.01 \end{array}\right\} given$$

Therefore the Decision Boundary for this case:

$$-30x_1 + 300x_2 - 1230 = 0$$

$$x_1 = 100,\ x_2 = \frac{30 \times 100 + 1230}{300} = 14.1$$

$$x_1 = 200,\ x_2 = \frac{30 \times 200 + 1230}{300} = 24.1$$



Initial hyperplane does not separate the two classes.
Therefore we need to **Train** the Neural Network