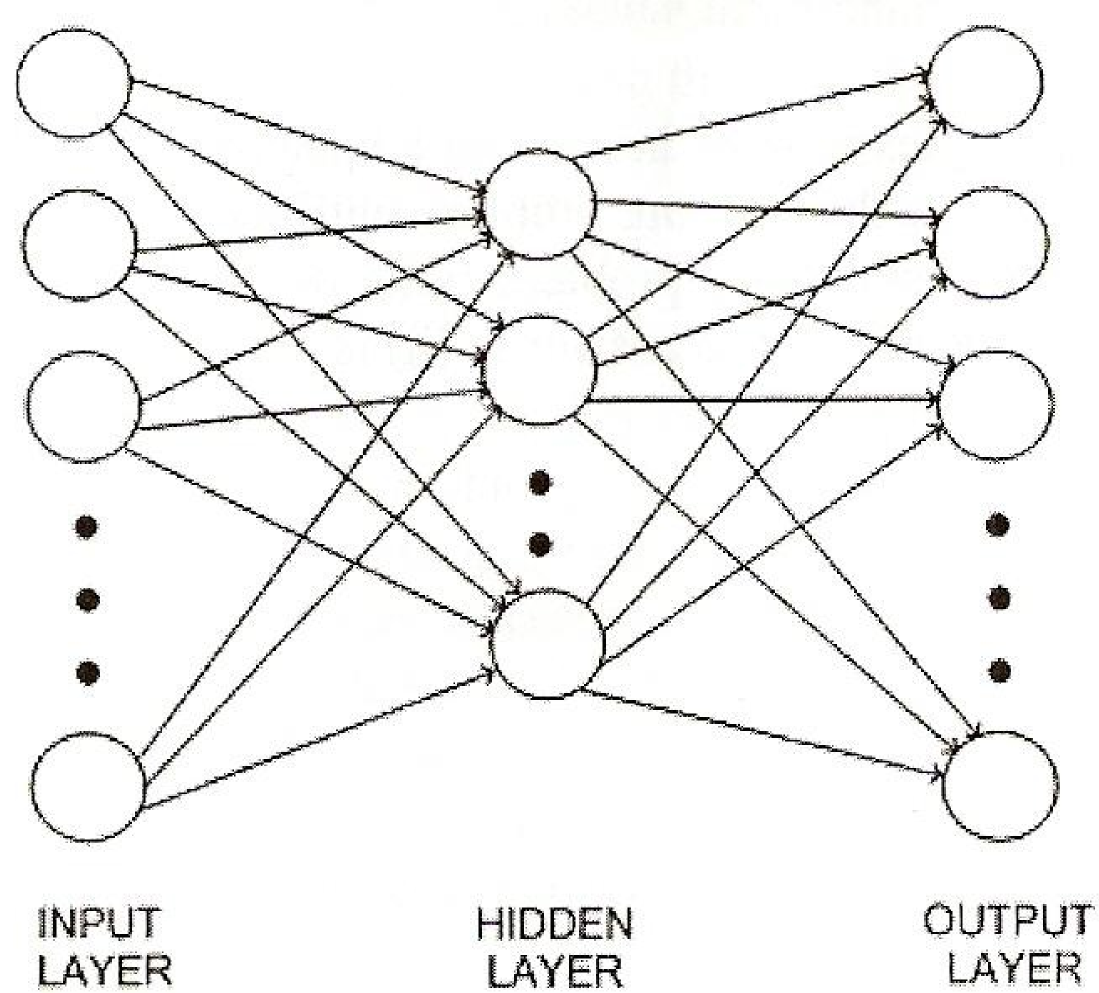
The Simple Perceptron

Artificial Neural Network

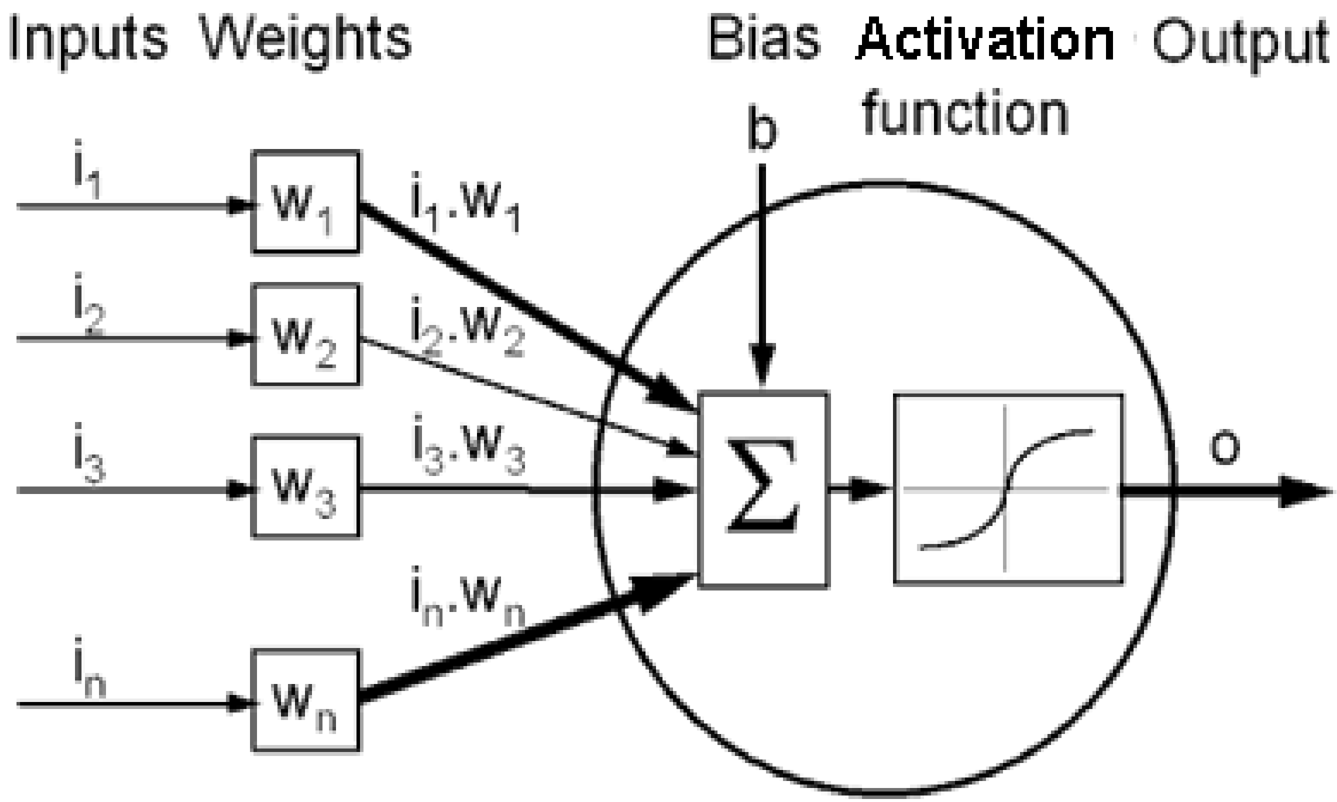
- Information processing architecture loosely modelled on the brain
- Consist of a large number of interconnected processing units (neurons)
- Work in parallel to accomplish a global task
- Generally used to model relationships between inputs and outputs or find patterns in data

Artificial Neural Network

- 3 Types of layers

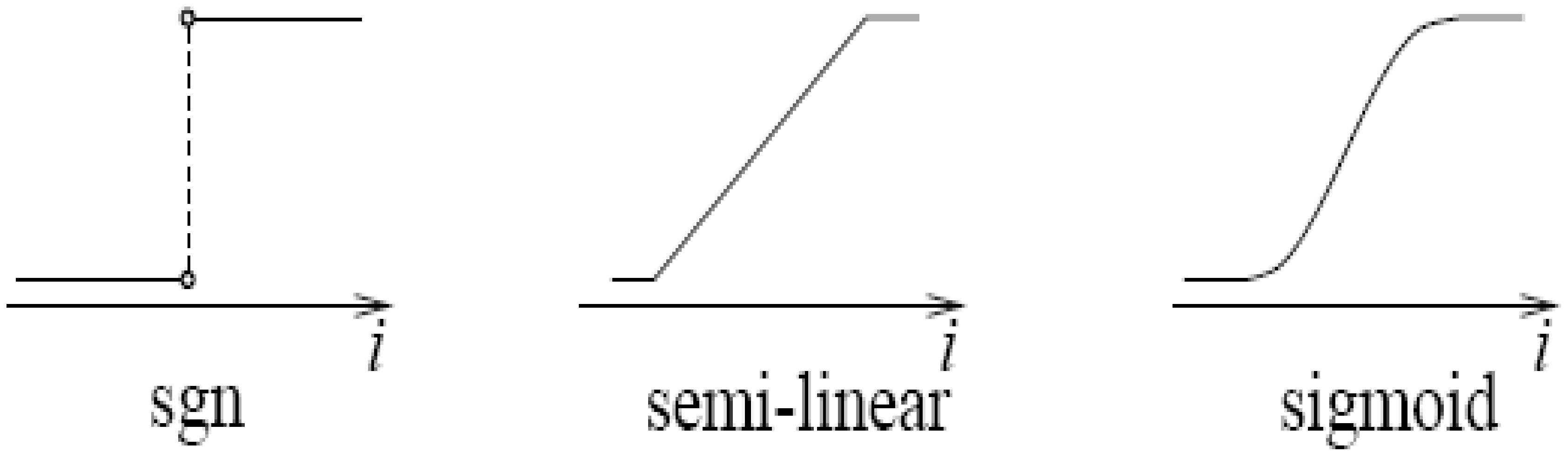


Single Processing Unit



Activation Functions

- Function which takes the total input and produces an output for the node given some threshold.

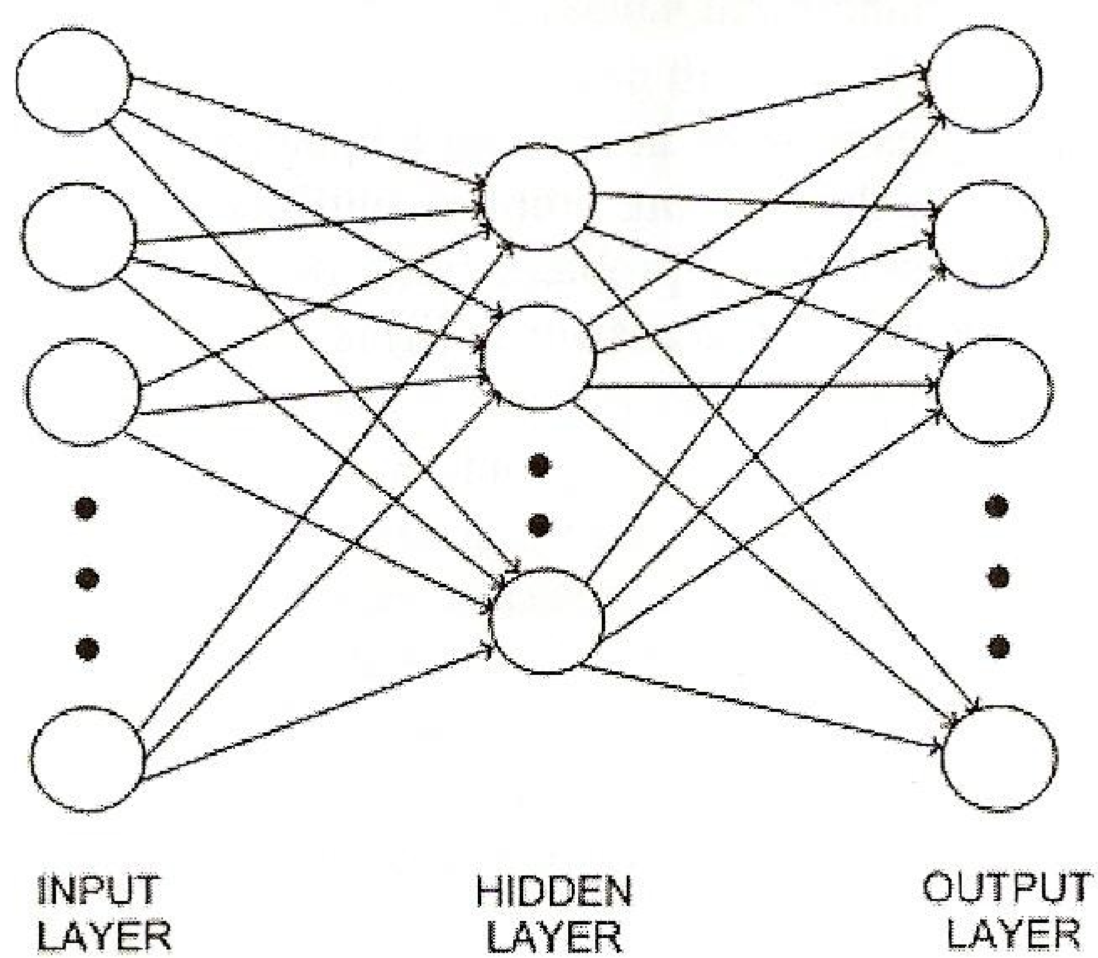


Network Structure

- Two main network structures

1. Feed-Forward Network

2. Recurrent Network

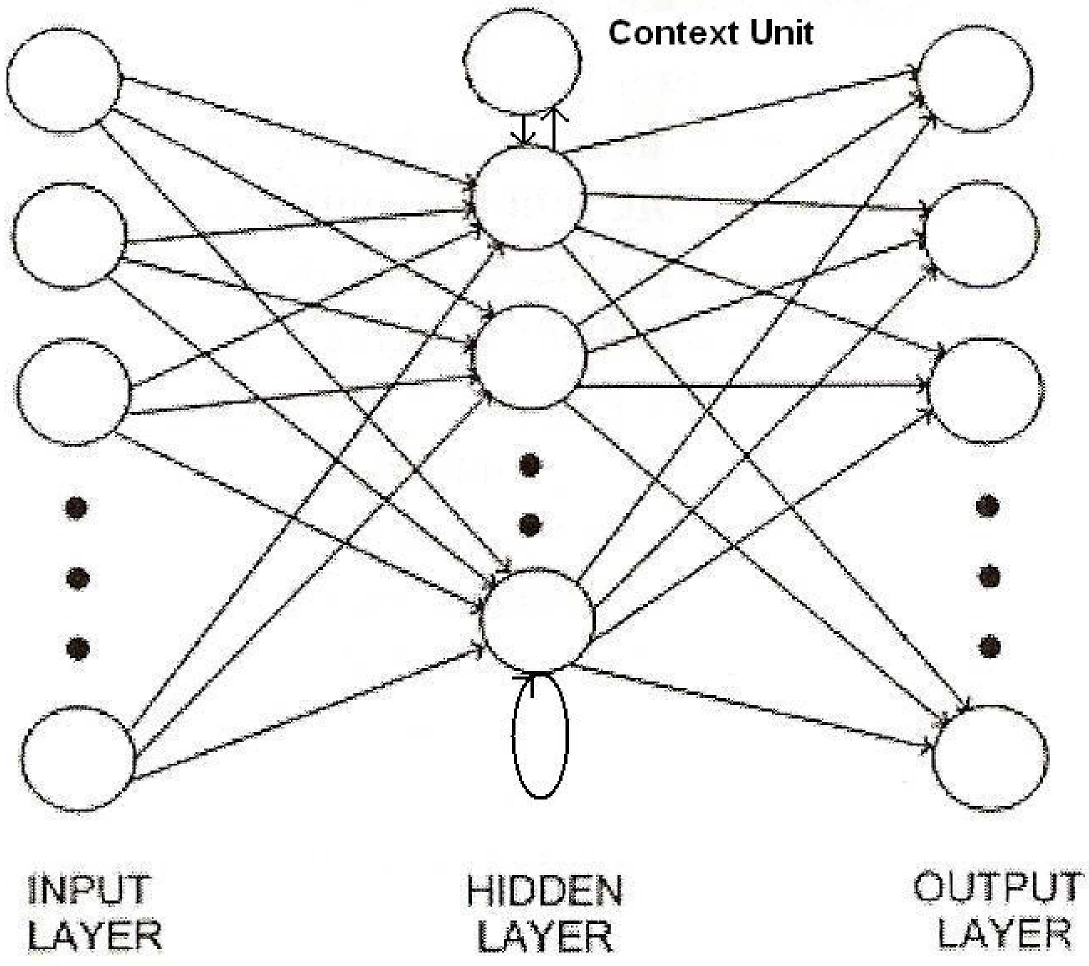


Network Structure

- Two main network structures

1. Feed-Forward Network

2. Recurrent Network



Learning Paradigms

- Supervised Learning:

  Given training data consisting of pairs of inputs/outputs, find a function which correctly matches them

- Unsupervised Learning:

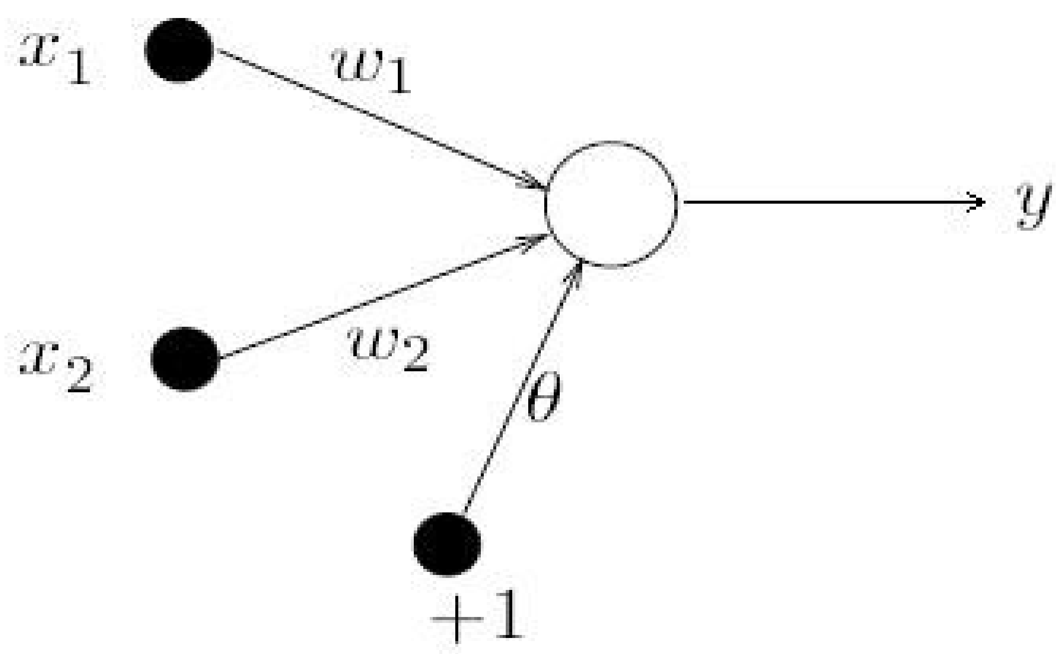
  Given a data set, the network finds patterns and categorizes into data groups.

- Reinforcement Learning:

  No data given. Agent interacts with the environment calculating cost of actions.

Simple Perceptron

- The perceptron is a single layer feed-forward neural network.



Simple Perceptron

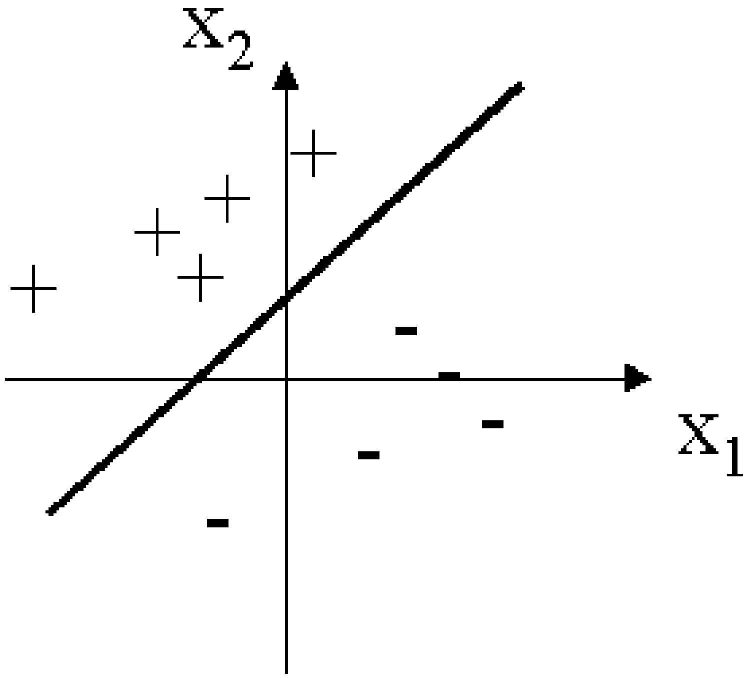
- Simplest output function

$$y = \operatorname{sgn}\left(\sum_{i=1}^{2} w_i x_i + \theta\right)$$

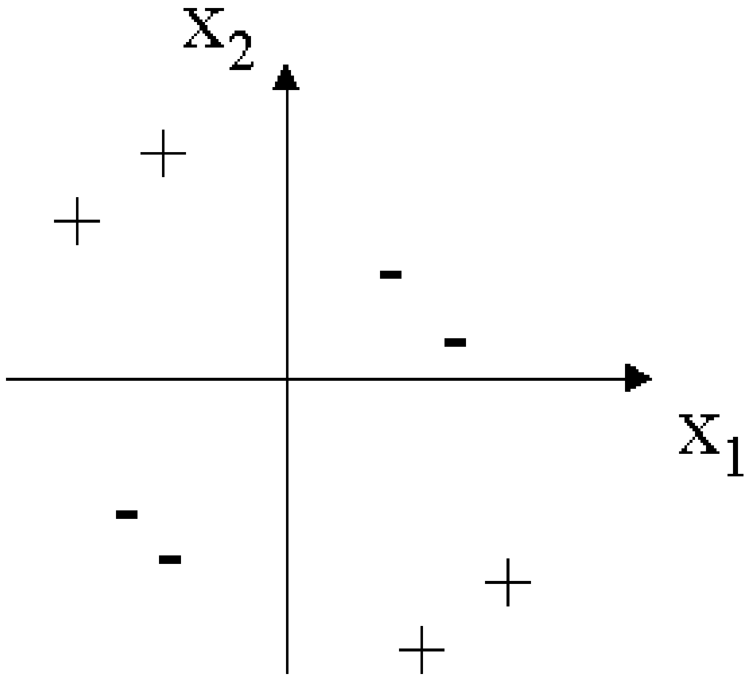
$$\operatorname{sgn}(s) = \begin{cases} 1 & \text{if } s > 0 \\ -1 & \text{otherwise.} \end{cases}$$

- Used to classify patterns said to be linearly separable

Linearly Separable



Linearly Separable



Not Linearly Separable

$$w_1 x_1 + w_2 x_2 + \theta = 0$$

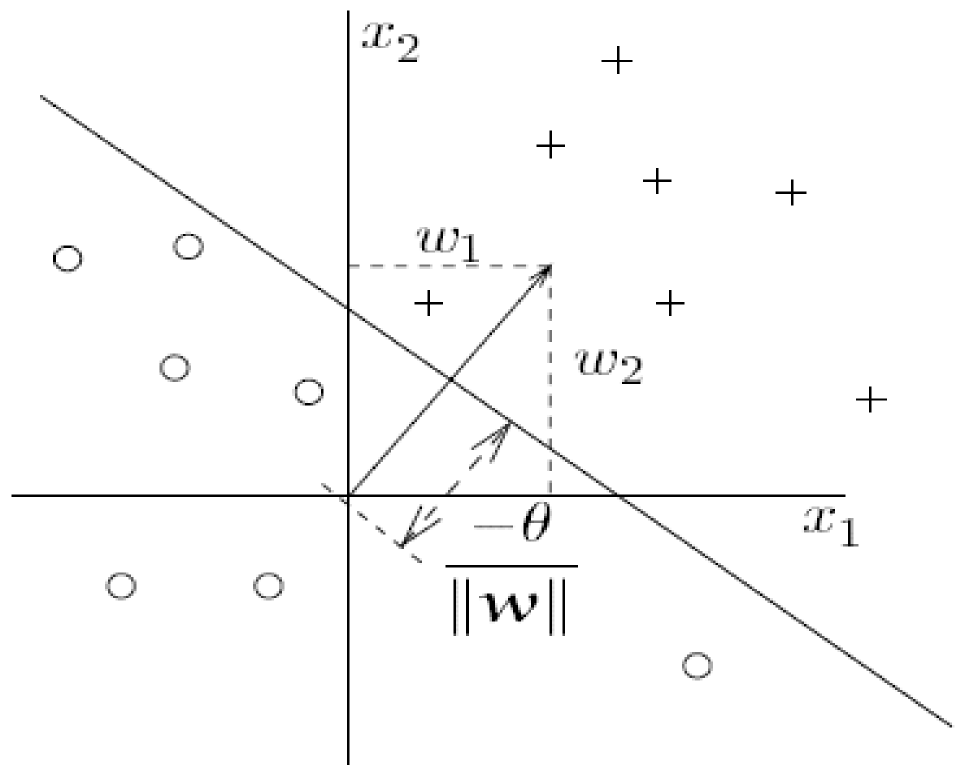
Linearly Separable

The bias is proportional to the offset of the plane from the origin

The weights determine the slope of the line

The weight vector is perpendicular to the plane

$$x_2 = -\frac{w_1}{w_2}x_1 - \frac{\theta}{w_2}$$



Perceptron Learning Algorithm

- We want to train the perceptron to classify inputs correctly
- Accomplished by adjusting the connecting weights and the bias
- Can only properly handle linearly separable sets

Perceptron Learning Algorithm

- We have a "training set" which is a set of input vectors used to train the perceptron.
- During training both w_i and θ *(bias)* are modified for convenience, let $w_0 = \theta$ and $x_0 = 1$
- *Let,* η*,* the *learning rate*, be a small positive number (small steps lessen the possibility of destroying correct classifications)
- Initialise w_i to some values

Perceptron Learning Algorithm

$$\textit{Desired output} \qquad d(n) = \begin{cases} +1 & \textit{if } x(n) \in \textit{set } A \\ -1 & \textit{if } x(n) \in \textit{set } B \end{cases}$$

1. Select random sample from training set as input
2. If classification is correct, do nothing
3. If classification is incorrect, modify the weight vector *w* using

$$w_i = w_i + \eta d(n) x_i(n)$$

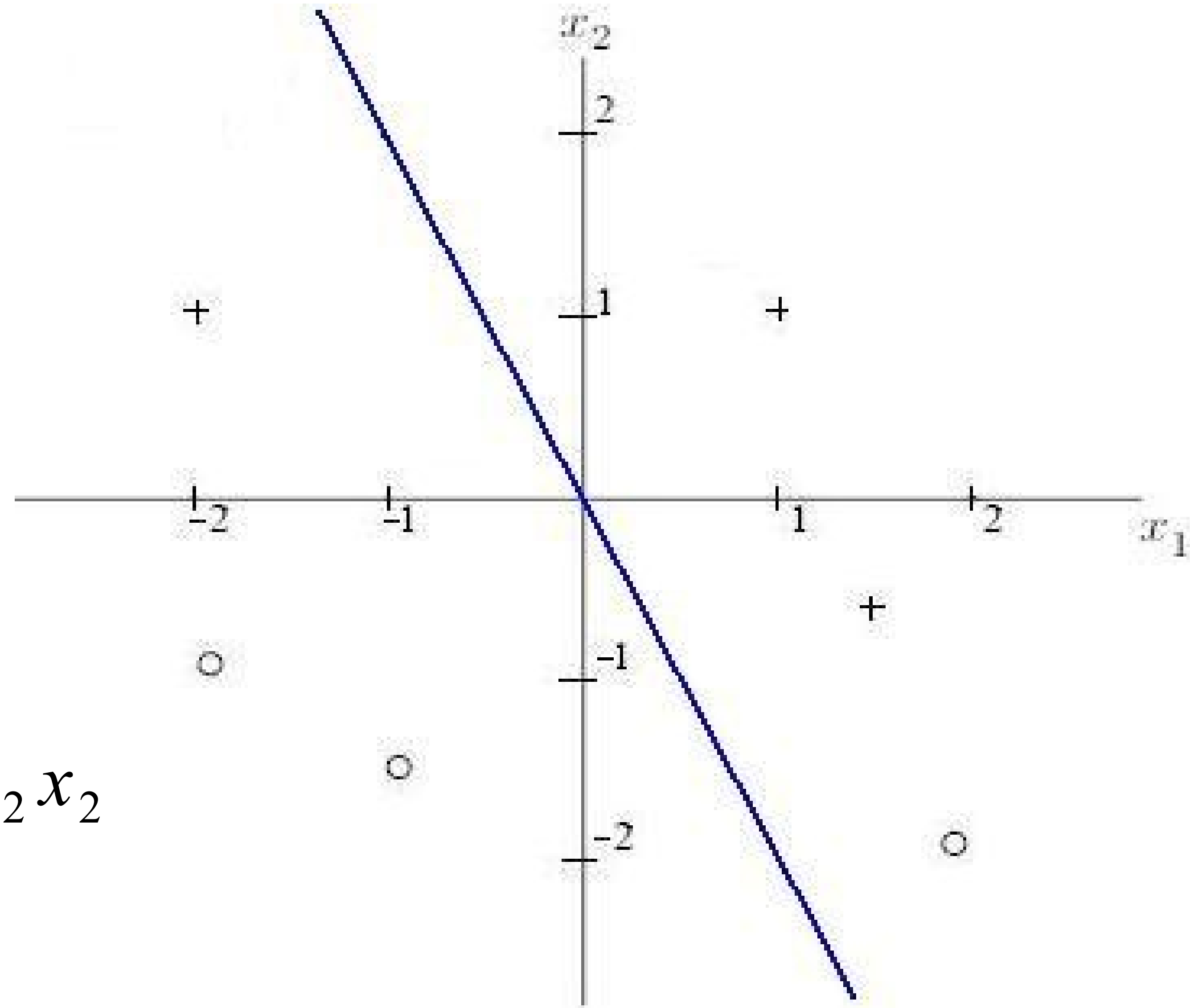
Repeat this procedure until the entire training set is classified correctly

Learning Example

Initial Values:

$\eta = 0.2$

$$w = \begin{pmatrix} 0 \\ 1 \\ 0.5 \end{pmatrix}$$

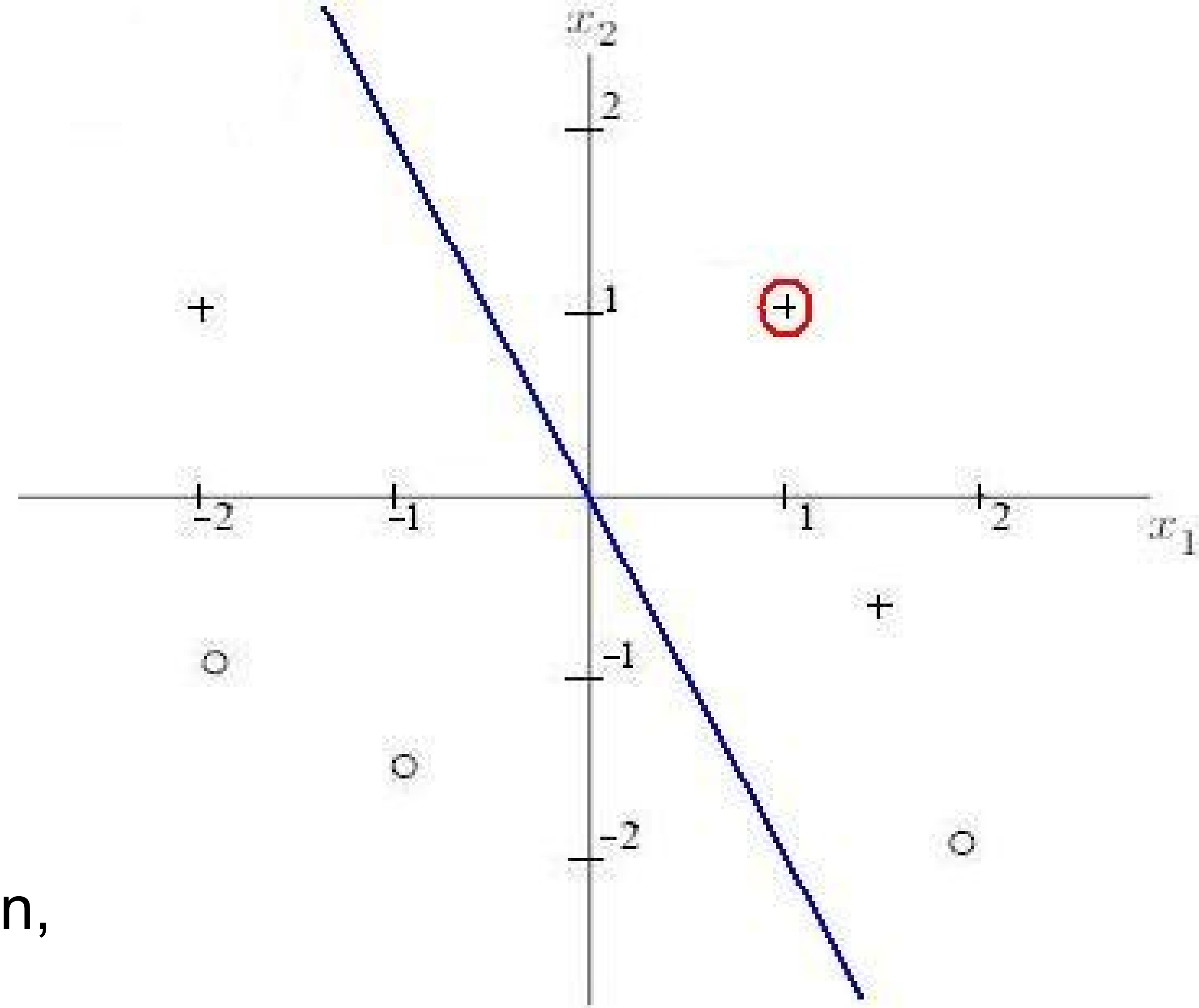


$$0 = w_0 + w_1 x_1 + w_2 x_2$$
$$= 0 + x_1 + 0.5x_2$$
$$\Rightarrow \quad x_2 = -2x_1$$

Learning Example

$\eta = 0.2$

$$w = \begin{pmatrix} 0 \\ 1 \\ 0.5 \end{pmatrix}$$



$x_1 = 1$, $x_2 = 1$
$w^T x > 0$

Correct classification, no action

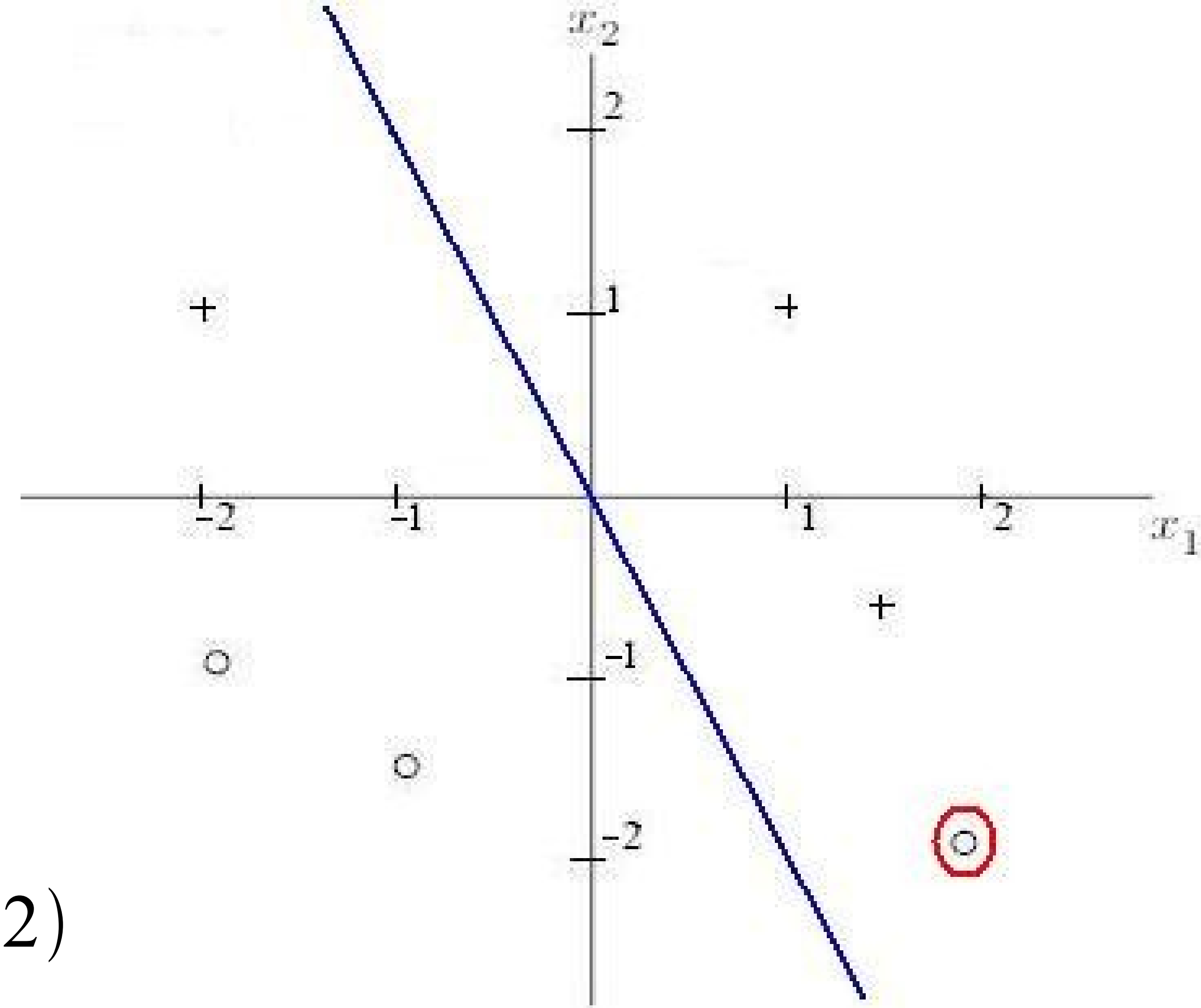
Learning Example

$\eta = 0.2$

$$w = \begin{pmatrix} 0 \\ 1 \\ 0.5 \end{pmatrix}$$

$x_1 = 2,\ x_2 = -2$

$$w_0 = w_0 - 0.2 * 1$$
$$w_1 = w_1 - 0.2 * 2$$
$$w_2 = w_2 - 0.2 * (-2)$$



Learning Example

$\eta = 0.2$

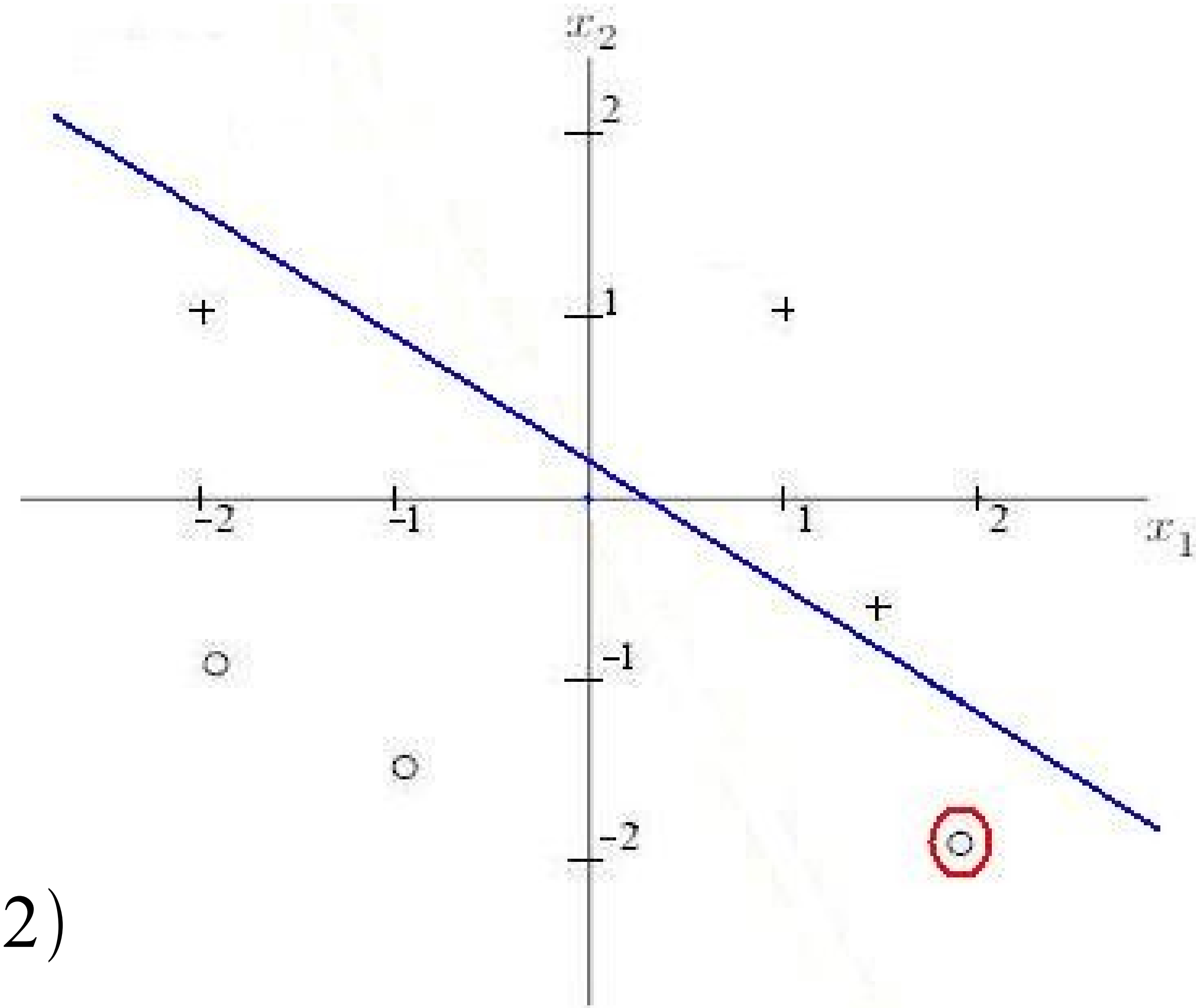
$$w = \begin{pmatrix} -0.2 \\ 0.6 \\ 0.9 \end{pmatrix}$$

$x_1 = 2,\ x_2 = -2$

$$w_0 = w_0 - 0.2 * 1$$

$$w_1 = w_1 - 0.2 * 2$$

$$w_2 = w_2 - 0.2 * (-2)$$



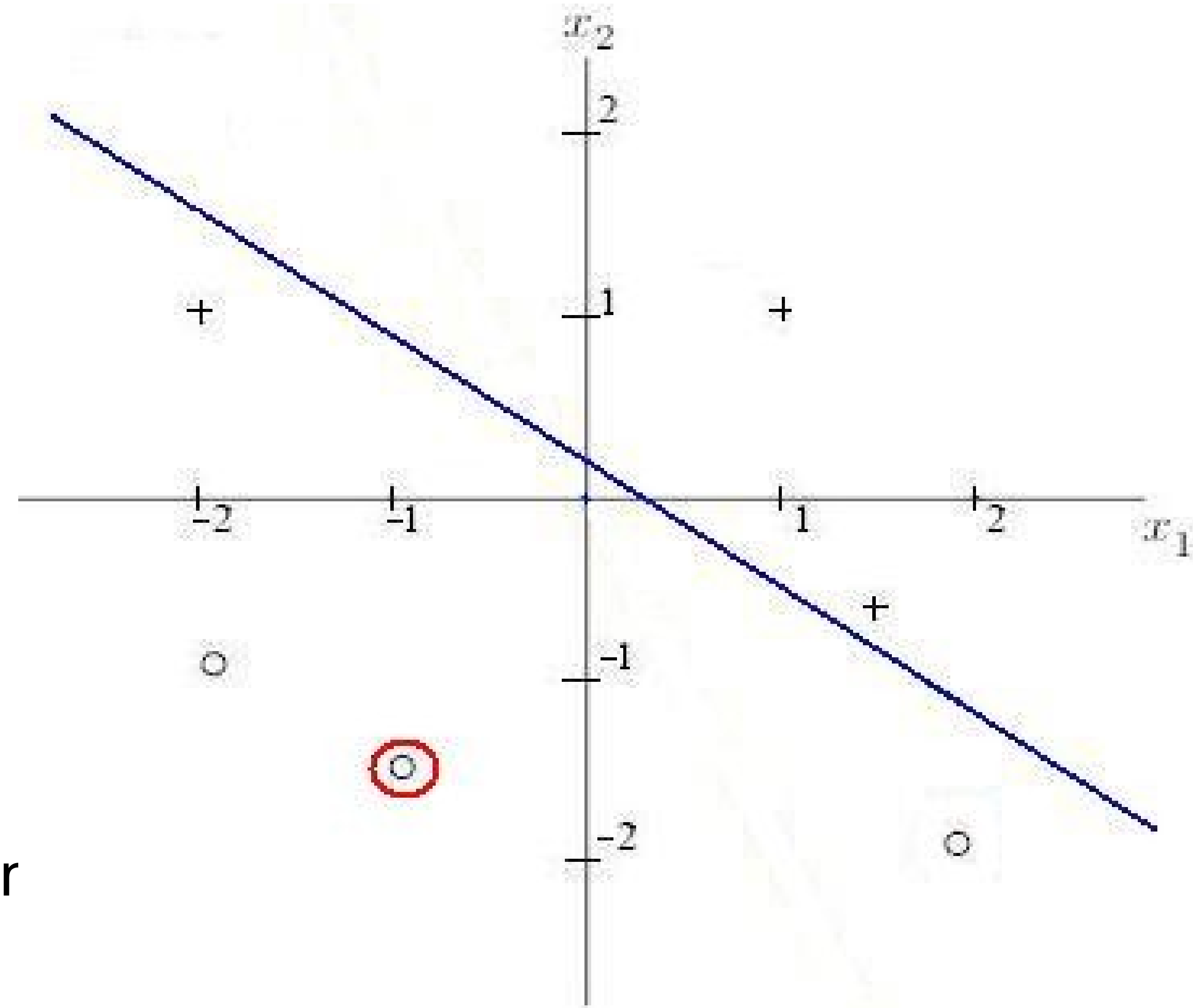
Learning Example

$\eta = 0.2$

$$w = \begin{pmatrix} -0.2 \\ 0.6 \\ 0.9 \end{pmatrix}$$

$x_1 = -1,\ x_2 = -1.5$
$w^T x < 0$

Correct classificatior
no action



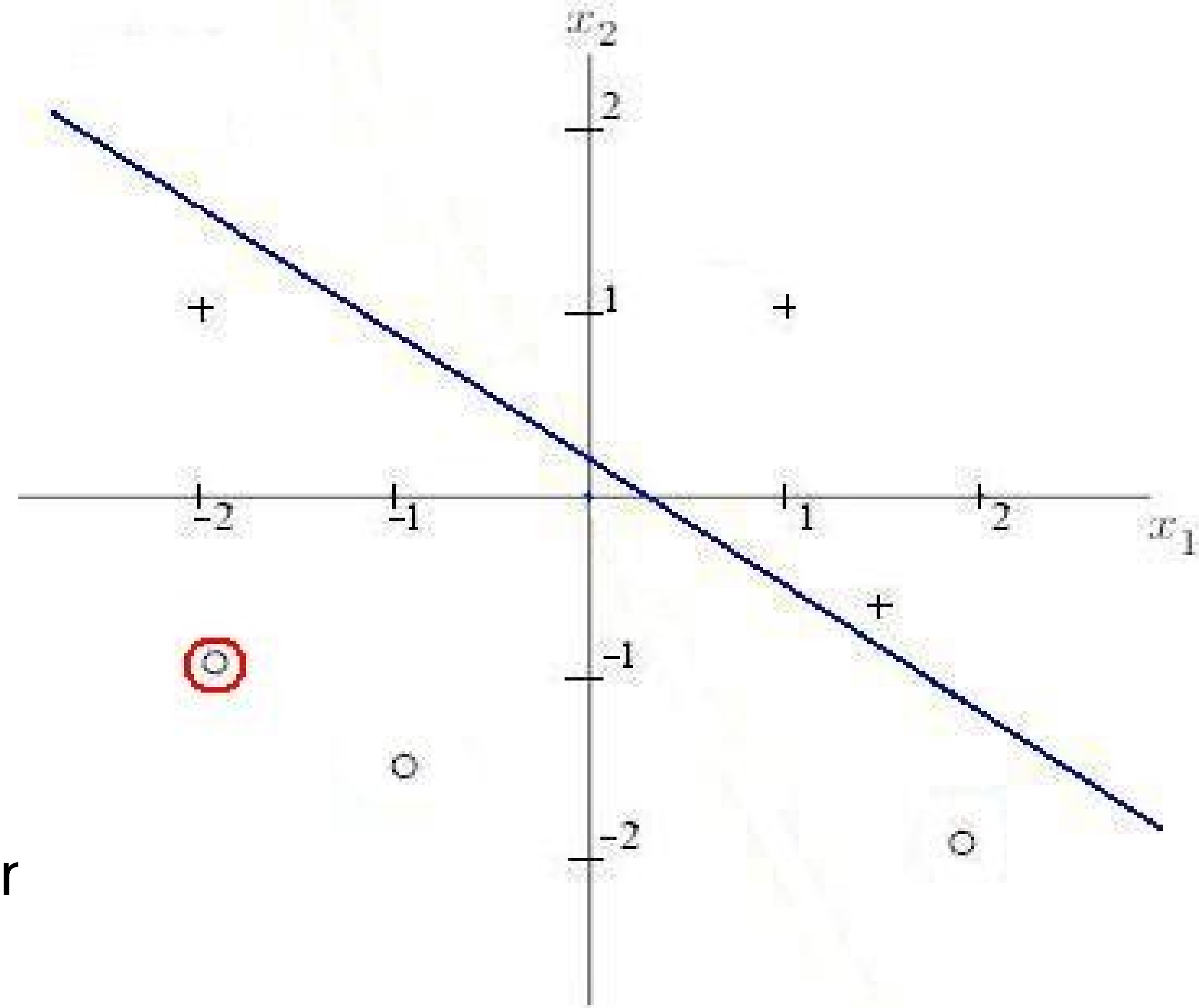
Learning Example

$\eta = 0.2$

$$w = \begin{pmatrix} -0.2 \\ 0.6 \\ 0.9 \end{pmatrix}$$

$x_1 = -2,\ x_2 = -1$
$w^T x < 0$

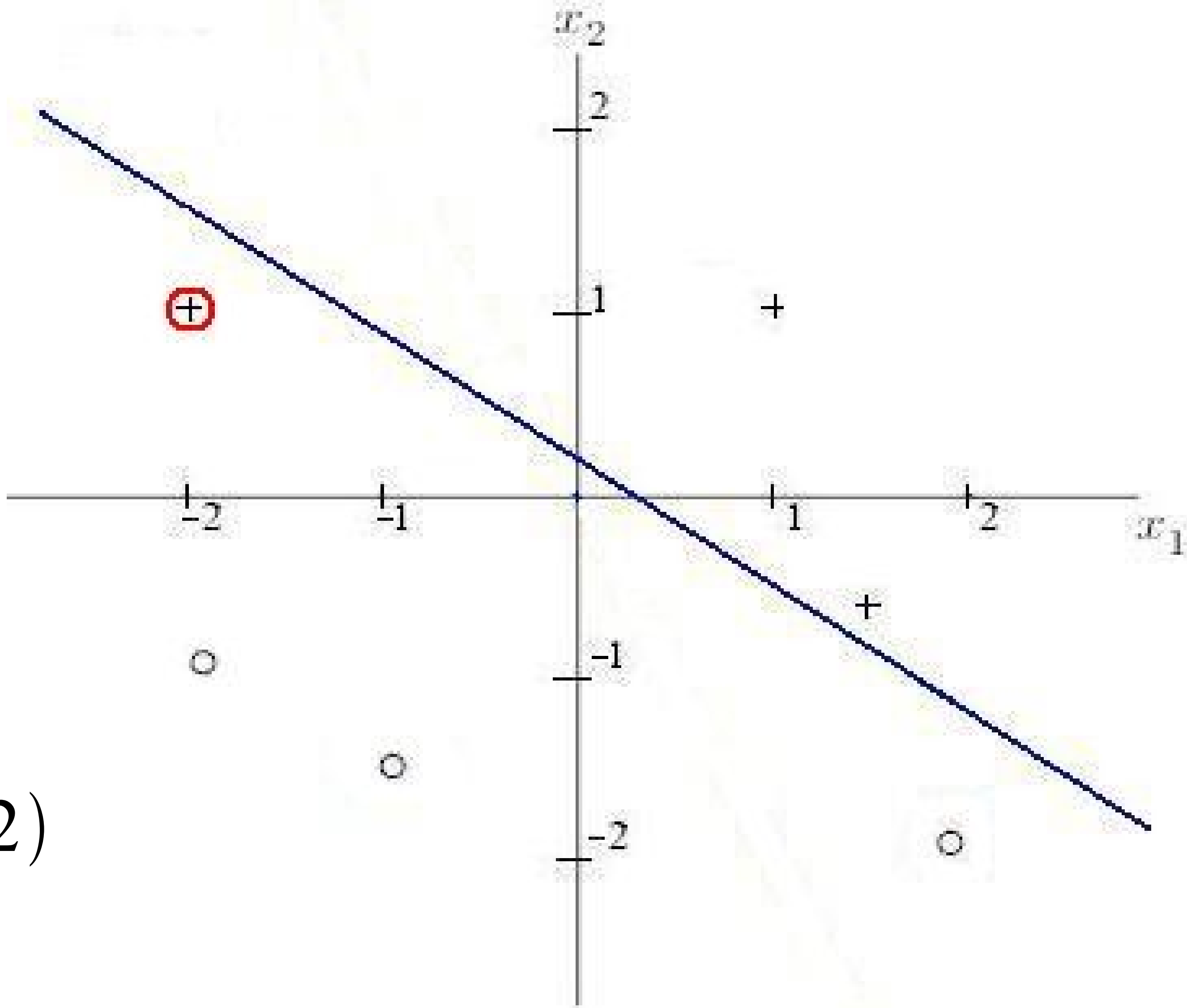
Correct classificatior
no action



Learning Example

$\eta = 0.2$

$$w = \begin{pmatrix} -0.2 \\ 0.6 \\ 0.9 \end{pmatrix}$$



x_1 = -2, x_2 = 1

$$w_0 = w_0 + 0.2 * 1$$

$$w_1 = w_1 + 0.2 * (-2)$$

$$w_2 = w_2 + 0.2 * 1$$

Learning Example

$\eta = 0.2$

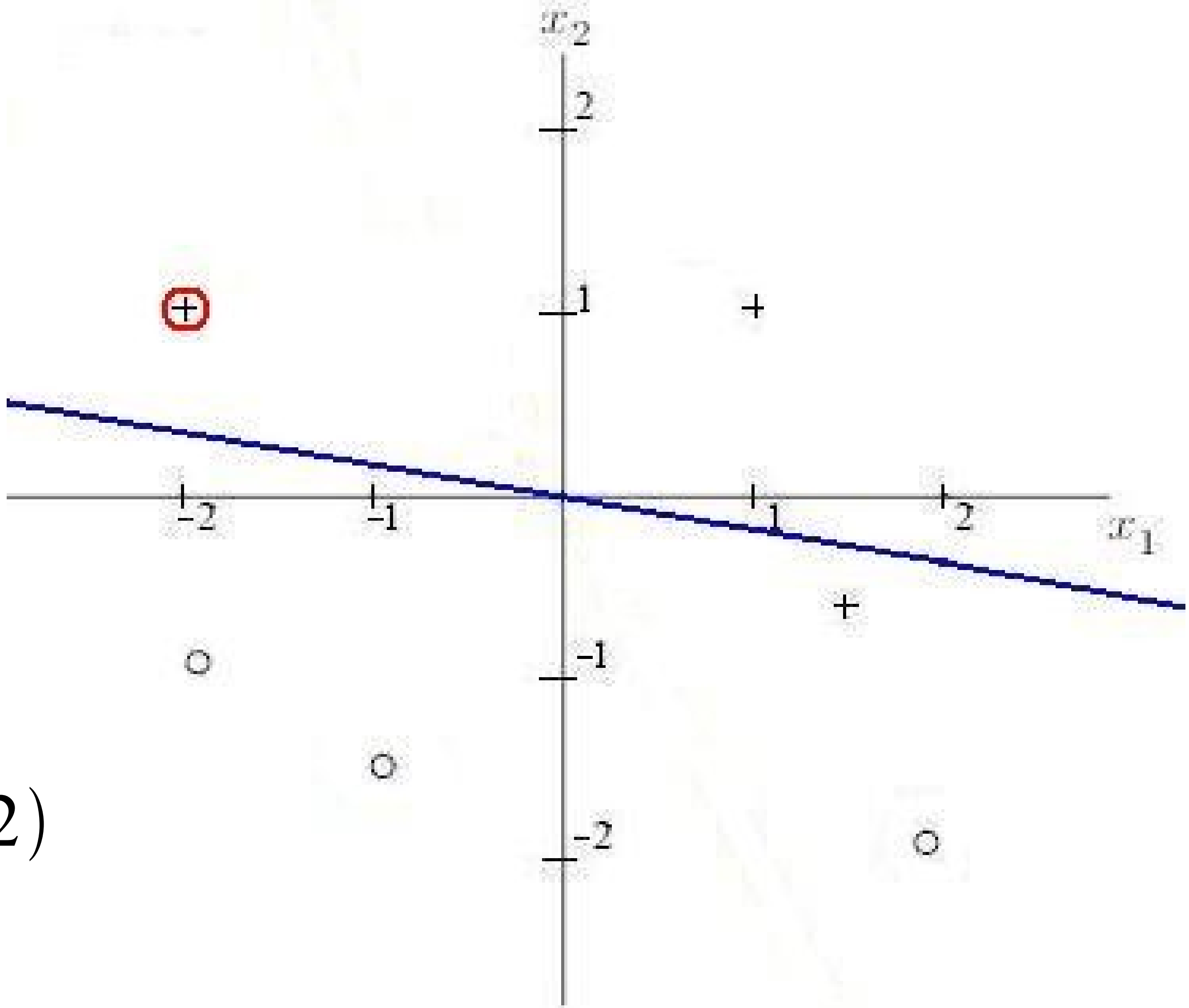
$$w = \begin{pmatrix} 0 \\ 0.2 \\ 1.1 \end{pmatrix}$$

x_1 = -2, x_2 = 1

$$w_0 = w_0 + 0.2 * 1$$

$$w_1 = w_1 + 0.2 * (-2)$$

$$w_2 = w_2 + 0.2 * 1$$



Learning Example

$\eta = 0.2$

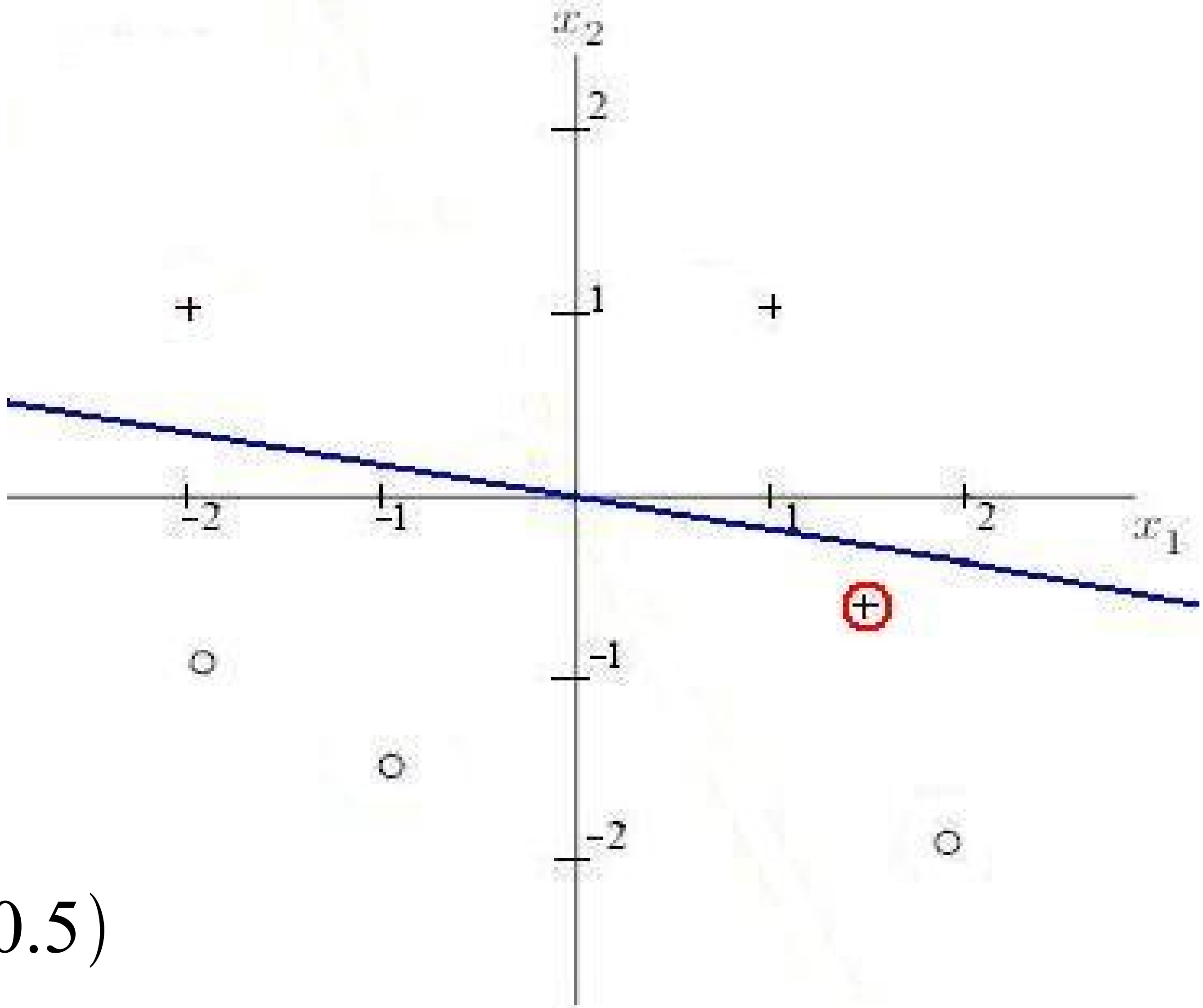
$$w = \begin{pmatrix} 0 \\ 0.2 \\ 1.1 \end{pmatrix}$$

$x_1 = 1.5,\ x_2 = -0.5$

$$w_0 = w_0 + 0.2 * 1$$

$$w_1 = w_1 + 0.2 * 1.5$$

$$w_2 = w_2 + 0.2 * (-0.5)$$



Learning Example

$\eta = 0.2$

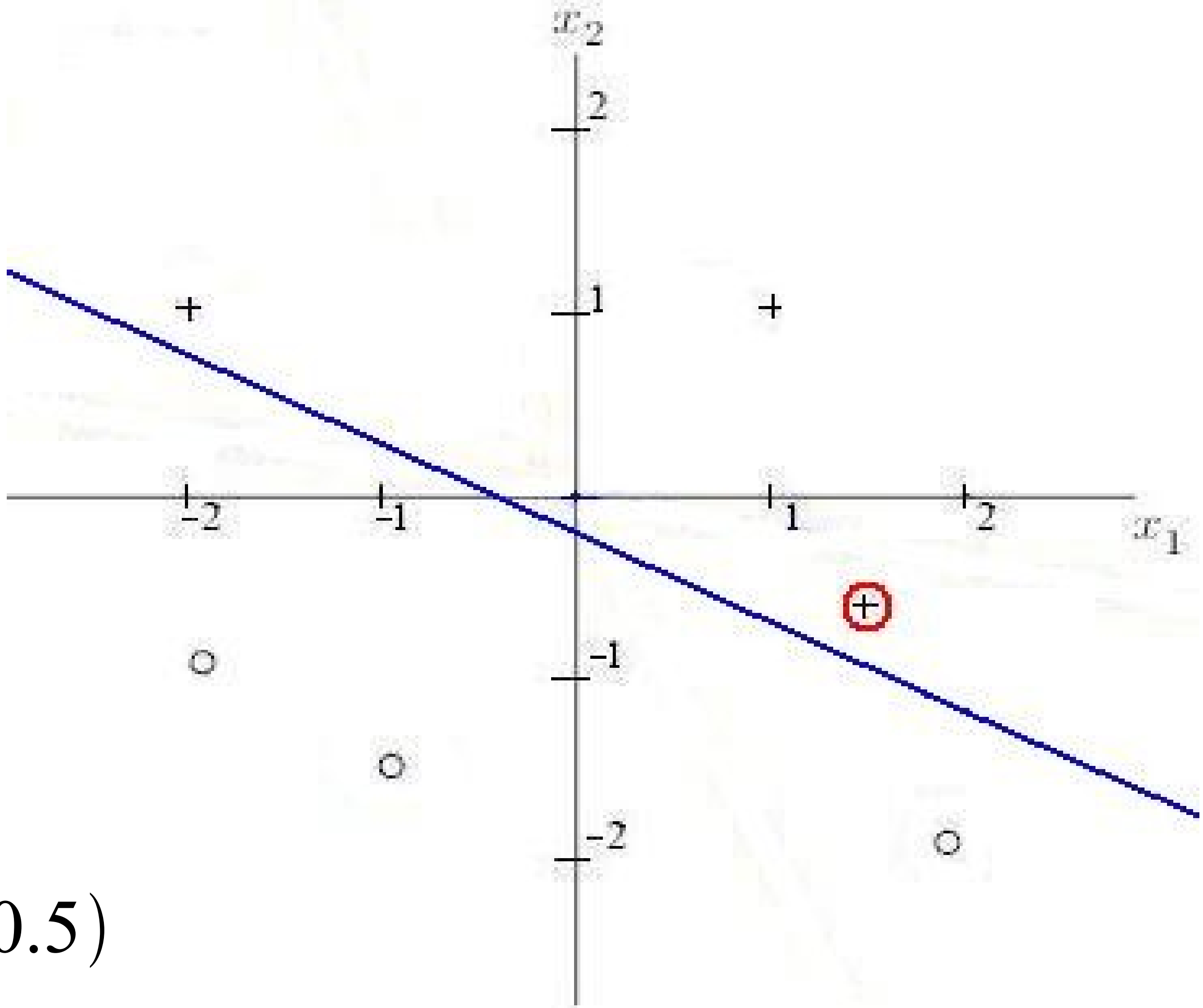
$$w = \begin{pmatrix} 0.2 \\ 0.5 \\ 1 \end{pmatrix}$$

$x_1 = 1.5,\ x_2 = -0.5$

$$w_0 = w_0 + 0.2 * 1$$

$$w_1 = w_1 + 0.2 * 1.5$$

$$w_2 = w_2 + 0.2 * (-0.5)$$



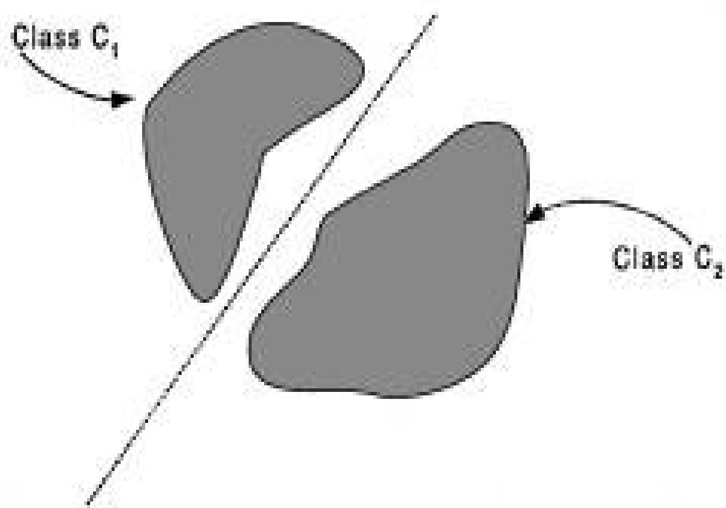
Perceptron Convergence Theorem

- The theorem states that for any data set which is linearly separable, the perceptron learning rule is guaranteed to find a solution in a finite number of iterations.

- Idea behind the proof: Find upper & lower bounds on the length of the weight vector to show finite number of iterations.

Perceptron Convergence Theorem

Let's assume that the input variables come from two linearly separable classes C_1 & C_2.



Let T_1 & T_2 be subsets of training vectors which belong to the classes C_1 & C_2 respectively. Then $T1 \cup T2$ is the complete training set.

Perceptron Convergence Theorem

As we have seen, the learning algorithms purpose is to find a weight vector *w* such that

$$w \cdot x > 0 \quad \forall x \in C_1$$
$$w \cdot x \leq 0 \quad \forall x \in C_2$$

*(**x** is an input vector)*

If the *k*th member of the training set, *x(k)*, is correctly classified by the weight vector *w(k)* computed at the *k*th iteration of the algorithm, then we do not adjust the weight vector.

However, if it is incorrectly classified, we use the modifier

$$w(k+1) = w(k) + \eta d(k) x(k)$$

Perceptron Convergence Theorem

So we get

$$w(k+1) = w(k) - \eta x(k) \quad if \quad w(k) \cdot x(k) > 0,\ x(k) \in C_2$$
$$w(k+1) = w(k) + \eta x(k) \quad if \quad w(k) \cdot x(k) \leq 0,\ x(k) \in C_1$$

We can set $\eta = 1$, as for $\eta \neq 1$ *(>0)* just scales the vectors.

We can also set the initial condition *w(0) = 0*, as any non-zero value will still converge, just decrease or increase the number of iterations.

Perceptron Convergence Theorem

Suppose that $w(k) \cdot x(k) < 0$ *for* $k = 1, 2, \ldots$ where $x(k) \in T_1$, so with an incorrect classification we get

$$w(k+1) = w(k) + x(k) \quad x(k) \in C_1$$

By expanding iteratively, we get

$$\begin{aligned} w(k+1) &= x(k) + w(k) \\ &= x(k) + x(k-1) + w(k-1) \\ &\;\vdots \\ &= x(k) + \ldots + x(1) + w(0) \end{aligned}$$

Perceptron Convergence Theorem

As we assume linear separability, $\exists$ a solution w^* where $w \cdot x(k) > 0$, $x(1) \ldots x(k) \in T_1$. Multiply both sides by the solution w^* to get

$$w^* \cdot w(k+1) = w^* \cdot x(1) + \ldots + w^* \cdot x(k)$$

These are all > 0, hence all >= α, where

$$\alpha = min \quad w^* \cdot x(k)$$

Thus we get

$$w^* \cdot w(k+1) \geq k\alpha$$

Perceptron Convergence Theorem

Now we make use of the Cauchy-Schwarz inequality which states that for any two vectors A, B

$$\|A\|^2\|B\|^2 = (A \cdot B)^2$$

Applying this we get

$$\|w^*\|^2\|w(k+1)\|^2 \geq (w^* \cdot w(k+1))^2$$

From the previous slide we know $w^* \cdot w(k+1) \geq k\alpha$

Thus, it follow that

$$\|w(k+1)\|^2 \geq \frac{k^2\alpha^2}{\|w^*\|^2}$$

Perceptron Convergence Theorem

We continue the proof by going down another route.

$$w(j+1) = w(j) + x(j) \quad for\ j = 1, \ldots, k \quad with\ x(j) \in T_1$$

We square the Euclidean norm on both sides

$$\begin{aligned} \|w(j+1)\|^2 &= \|w(j) + x(j)\|^2 \\ &= \|w(j)\|^2 + \|x(j)\|^2 + 2\mathrm{w}(j) \cdot x(j) \end{aligned}$$

incorrectly classified, so < 0

Thus we get

$$\|w(j+1)\|^2 - \|w(j)\|^2 \leq \|x(j)\|^2$$

Perceptron Convergence Theorem

Summing both sides for all *j*

$$\|w(j+1)\|^2 - \|w(j)\|^2 \le \|x(j)\|^2$$

$$\|w(j)\|^2 - \|w(j-1)\|^2 \le \|x(j-1)\|^2$$

$$\vdots$$

$$\|w(1)\|^2 - \|w(0)\|^2 \le \|x(1)\|^2$$

We get

$$\|w(k+1)\|^2 \le \sum_{j=1}^{k} \|x(j)\|^2$$

$$\le k\beta \qquad \beta = max \ \|x(j)\|^2$$

Perceptron Convergence Theorem

But now we have a conflict between the equations, for sufficiently large values of *k*

$$\|w(k+1)\|^2 \leq k\beta \qquad \|w(k+1)\|^2 \geq \frac{k^2\alpha^2}{\|w^*\|^2}$$

So, we can state that *k* cannot be larger than some value k_{max} for which the two equations are both satisfied.

$$k_{max}\beta = \frac{k_{max}^2\alpha^2}{\|w^*\|^2} \quad \Rightarrow \quad k_{max} = \frac{\beta\|w^*\|^2}{\alpha^2}$$

Perceptron Convergence Theorem

Thus it is proved that for $\eta_k = 1, \forall k, w(0) = 0$, given that a solution vector w^* exists, the perceptron learning rule will terminate after at most k_{max} iterations.

The End