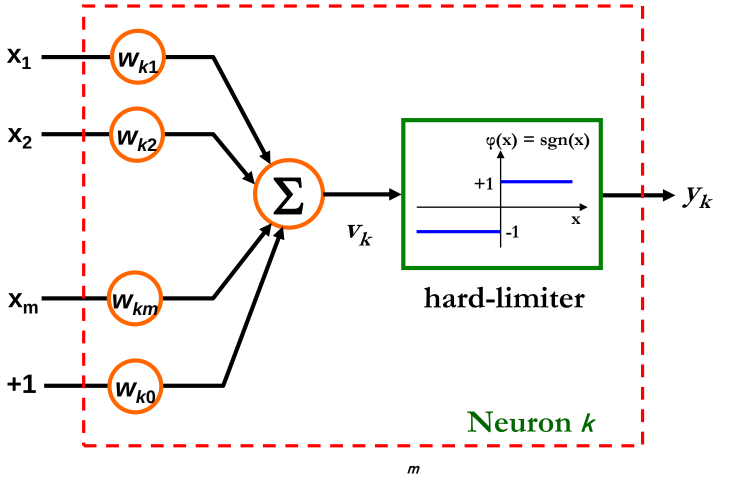
Perceptron

The McCulloch-Pitts neuron model



We set : $\mathbf{x_0 = 1,\ w_{k0} = -T_k}$. *Hence,* $\mathbf{v_k} = \sum_{i=0}^{m} \mathbf{w_{ki}\, x_i}$ *and* $\mathbf{y_k} = \mathrm{sgn}(\mathbf{v_k})$

Frank Rosenblatt

Fig. 1.4. • Frank Rosenblatt (the inventor of the perceptron and designer of the Mark I Perceptron neurocomputer) with the 400 pixel (20 × 20) Mark I Perceptron image sensor. Photo courtesy of Arvin Calspan Advanced Technology Center.

The 1st Neurocomputer / Mark I Perceptron

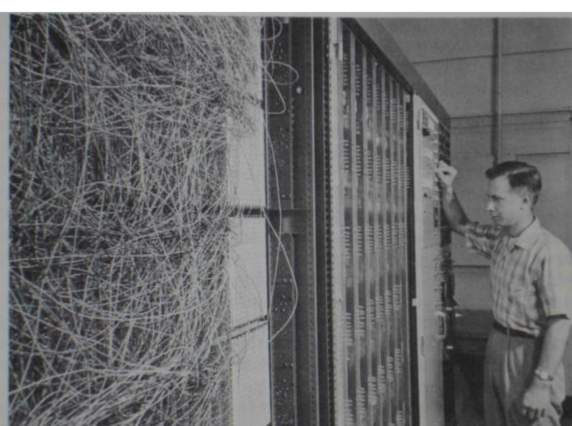
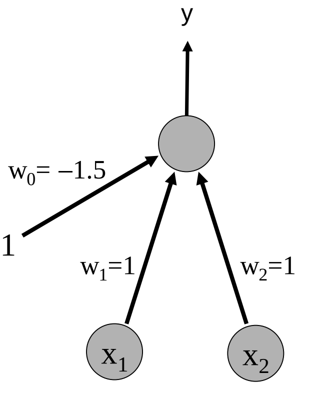


Fig. 1.6. • The Mark I Perceptron patchboard. The connection patterns were typically "random", so as to illustrate the ability of the perceptron to learn the desired pattern without need for precise wiring (in contrast to, unlike the precise wiring required in a programmed computer). Photo courtesy of Arvin Calspan Advanced Technology Center.

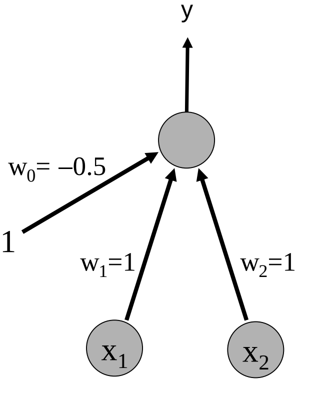
The Boolean AND problem

x1	x2	y
0	0	-1
0	1	-1
1	0	-1
1	1	1

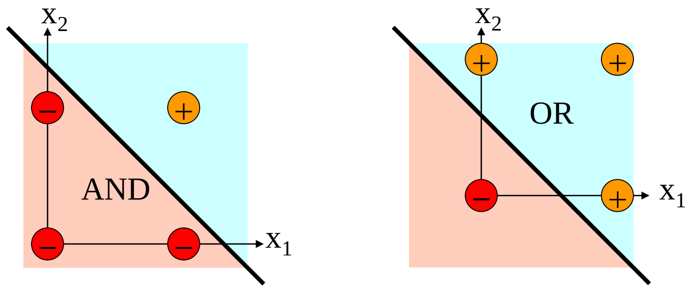


The Boolean OR problem

x1	x2	y
0	0	-1
0	1	1
1	0	1
1	1	1



Linear Separability



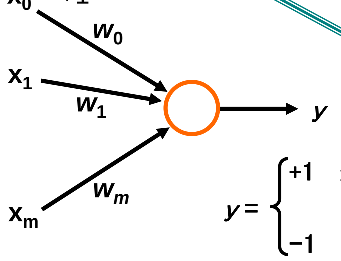
The Perceptron learning rule

Training mode

desired output *actual output*
learning rate *input*

$$w_i^{(new)} = w_i^{(old)} + \underbrace{\varepsilon\,(d - y)\,x_i}_{\Delta w_i}$$

correction term



$$y = \begin{cases} +1 & \text{if } \sum_{i=0}^{m} w_i x_i > 0 \\ -1 & \text{otherwise} \end{cases}$$

Classification mode

The Perceptron Training Algorithm

Step 1. Initialize weights and bias to small random values.

Step 2. Present new input $(x_1, \ldots, x_m)$ and desired output $d(t)$.

Assume that $x_0 = +1$ and $w_0 = -T$ (the bias term).

Step 3. Calculate actual output $y(t) = \text{sgn}\left(\sum_{i=0}^{m} w_i(t)x_i(t)\right)$

Step 4. Adapt weights

$$w_i(t+1) = w_i(t) + \varepsilon\,[d(t) - y(t)]\,x_i(t) \qquad \forall\, 0 \le i \le m$$

where $d(t) = 1$ or -1 if $x_i(t)$ from class A or B respectively

and the learning rate is chosen as $0 < \varepsilon < 1$.

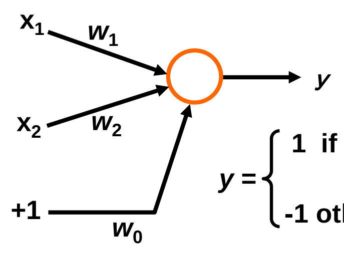
Step 5. Goto Step 2

The Perceptron Convergence Theorem

- If the problem is linearly separable then the perceptron learning algorithm converges in a finite (but unknown) number of iterations.
- On the other hand, if the problem is not linearly separable, Perceptron will never converge.

Solving the AND problem

x1	0	1	0	1
x2	0	0	1	1
d	-1	-1	-1	1



$$y = \begin{cases} 1 & \text{if } w_1x_1+w_2x_2+w_0 > 0 \\ -1 & \text{otherwise} \end{cases}$$

The decision boundary between categories is linear:

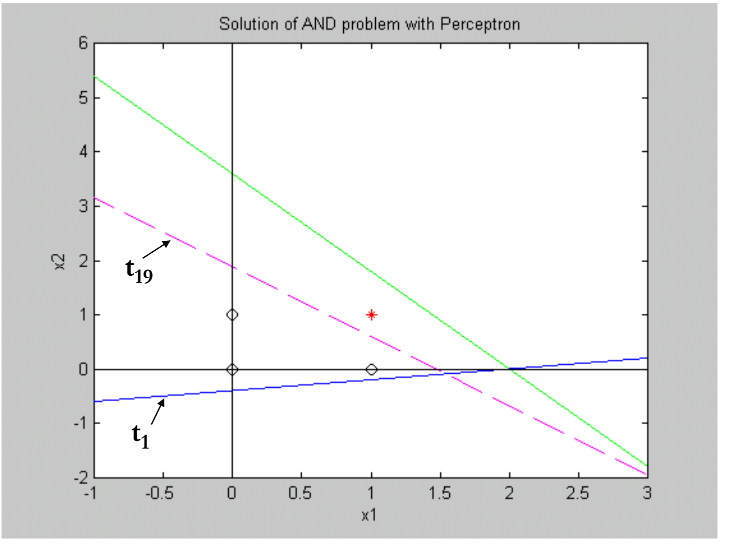
$$w_1x_1 + w_2x_2 + w_0 = 0 \quad \text{or} \quad \boxed{x_2 = \frac{-w_1}{w_2}x_1 + \frac{-w_0}{w_2}}$$

Let the initial weights be: $w_1 = -0.1$, $w_2 = 0.5$, $w_0 = 0.2$ and let $\varepsilon = 0.5$ and $v = w_1x_1 + w_2x_2 + w_0$

	x_1	x_2	v	$\frac{-w_1}{w_2}$	$\frac{-w_0}{w_2}$	y	d	Δw_1	Δw_2	Δw_0	w_1	w_2	w_0
t1:	0	0	0.20	0.20	1.60	1	-1	0	0	-1	-0.10	0.50	-0.80
t2:	1	1	-0.40	-0.60	-0.13	-1	1	1	1	1	0.90	1.50	0.20
t3:	0	1	1.70	-1.80	1.60	1	-1	0	-1	-1	0.90	0.50	-0.80
t4:	1	0	0.10	0.20	3.60	1	-1	-1	0	-1	-0.10	0.50	-1.80
t5:	0	0	-1.80	0.20	3.60	-1	-1	0	0	0	-0.10	0.50	-1.80
	. . .												
t17:	0	0	-2.80	-0.60	1.87	-1	-1	0	0	0	0.90	1.50	-2.80
t18:	1	1	-0.40	-0.76	0.72	-1	1	1	1	1	1.90	2.50	-1.80
t19:	0	1	0.70	-1.27	1.87	1	-1	0	-1	-1	1.90	1.50	-2.80
t20:	1	0	-0.90	-1.27	1.87	-1	-1	0	0	0	1.90	1.50	-2.80

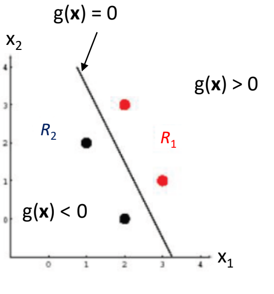
final weights

Graphical solution to AND problem



Geometric interpretation

- Input vectors **x** for which $\mathbf{w}^T\mathbf{x} + w_0 > 0$ belong to region R_1 and are classified as ω_1 while those for which $\mathbf{w}^T\mathbf{x} + w_0 < 0$ belong to region R_2 and are classified as ω_2. Input vectors for which $\mathbf{w}^T\mathbf{x} + w_0 = 0$ belong to the ***decision boundary***.
- ***Perceptron's goal***: find a linear decision boundary (i.e. estimate **w** and w_0) that classifies all training samples correctly.
- The function $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$ is called ***discriminant function*** because it can be used to discriminate the two classes.



Geometric interpretation

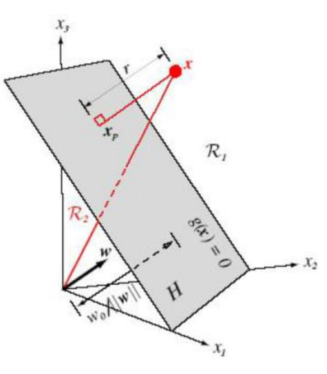
- The perceptron's decision boundary is the hyperplane H

  H: $g(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0$

- The orientation of the hyperplane is determined by the vector **w** and its exact position by the bias w_0.
- Vector **w** is perpendicular to the hyperplane.
- Equivalently, the unit vector $\hat{\mathbf{n}} = \mathbf{w} / \|\mathbf{w}\|$ determines the orientation of H and w_0 its translation from the origin of the axes.
- If $w_0 = 0$ then the hyperplane passes through the origin of the axes.
- The perceptron learning rule is a method of finding **w** and w_0.

Geometric interpretation

- g(**x**) provides an algebraic measure of the distance of **x** from H.



First, write **x** as:

$$\mathbf{x} = \mathbf{x}_p + r\,\hat{\mathbf{n}} = \mathbf{x}_p + r\,\frac{\mathbf{w}}{\|\mathbf{w}\|}$$

where $\mathbf{x}_p$ is the orthogonal projection of **x** on H.

Geometric interpretation

- Then, by substitution:

$$g(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = \mathbf{w}^{\mathrm{T}}(\mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|}) + w_0 =$$

$$= \mathbf{w}^{\mathrm{T}}\mathbf{x}_p + r\frac{\mathbf{w}^{\mathrm{T}}\mathbf{w}}{\|\mathbf{w}\|} + w_0 = \cancelto{0}{g(\mathbf{x}_p)} + r\,\|\mathbf{w}\| =$$

$$= r\,\|\mathbf{w}\|$$

where

$$\mathbf{w}^{\mathrm{T}}\mathbf{w} = \|\mathbf{w}\|^2$$

and

$$g(\mathbf{x}_p) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_p + w_0 = 0$$

since $\mathbf{x}_p$ belongs to the hyperplane.

Geometric interpretation

- Therefore, the distance of $\mathbf{x}$ from the hyperplane will be:

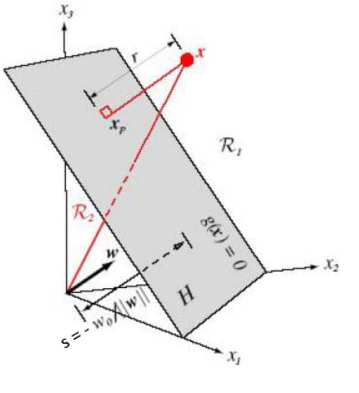
$$r = \frac{g(\mathbf{x})}{\|\mathbf{w}\|} \quad \text{if } x \in R_1 \text{ i.e. } g(x) > 0$$

$$r = \frac{-g(\mathbf{x})}{\|\mathbf{w}\|} \quad \text{if } x \in R_2 \text{ i.e. } g(x) < 0$$

while the distance s of H from the origin of the axes is found by setting $\mathbf{x} = 0$:

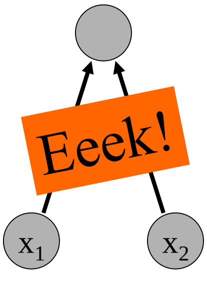
$$s = \frac{-w_0}{\|\mathbf{w}\|}$$

.



The Boolean XOR problem

x1	x2	y
0	0	-1
0	1	1
1	0	1
1	1	-1



XOR: Linear Separability?