Artificial Intelligence

Neural Networks

Lesson 2: Threshold Logic Units

Vincenzo Piuri

Università degli Studi di Milano

Contents

- Threshold logic units
- Geometric interpretation
- Linear separability
- Networks of threshold logic units
- Representing arbitrary boolean functions

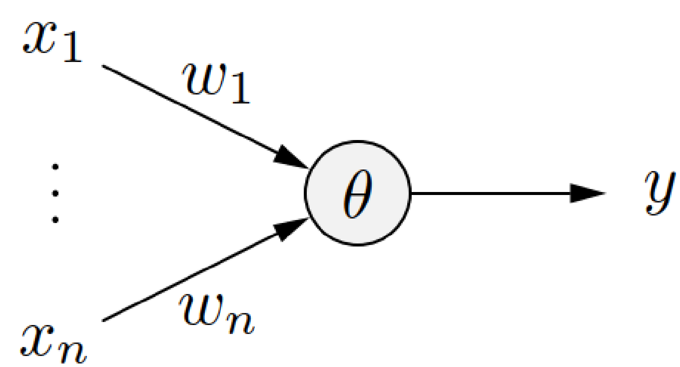
Threshold Logic Units (1)

A Threshold Logic Unit (TLU) is a processing unit for numbers with n inputs $x_1, \ldots, x_n$ and one output y. The unit has a threshold θ and each input x_i is associated with a weight w_i.

TLUs mimic the thresholding behavior of biological neurons in a (very) simple fashion.

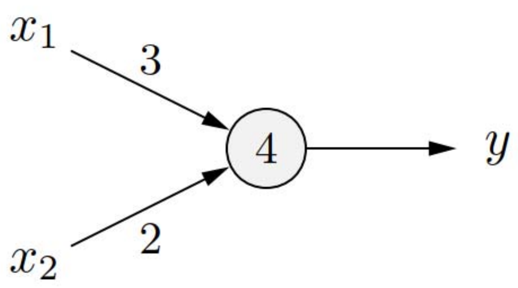
$$y = \begin{cases} 1, & \text{if} \quad \sum_{i=1}^{n} w_i x_i \geq \theta, \\ 0, & \text{otherwise.} \end{cases}$$

McCulloch-Pitts neuron



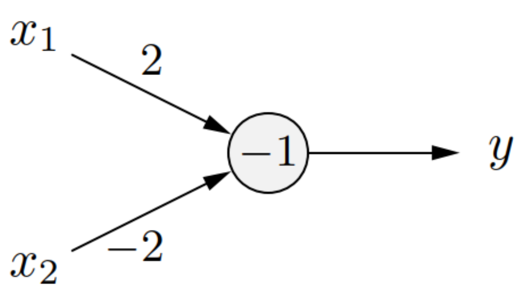
Threshold Logic Units (2)

Threshold logic unit for the conjunction $x_1 \wedge x_2$



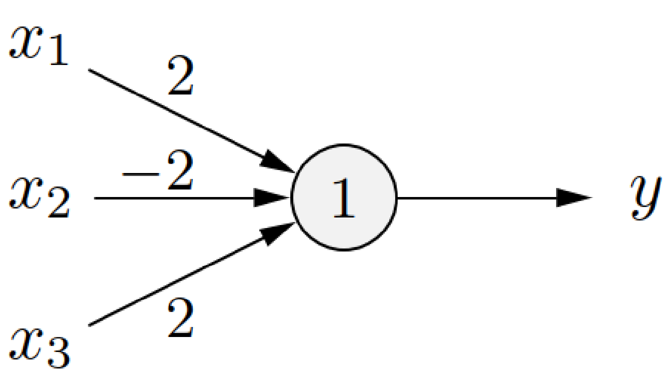
x_1	x_2	$3x_1 + 2x_2$	y
0	0	0	0
1	0	3	0
0	1	2	0
1	1	5	1

Threshold logic unit for the implication $x_2 \rightarrow x_1$



x_1	x_2	$2x_1 - 2x_2$	y
0	0	0	1
1	0	2	1
0	1	-2	0
1	1	0	1

Threshold Logic Units (3)



x_1	x_2	x_3	$\sum_i w_i x_i$	y
0	0	0	0	0
1	0	0	2	1
0	1	0	-2	0
1	1	0	0	0
0	0	1	2	1
1	0	1	4	1
0	1	1	0	0
1	1	1	2	1

Geometric Interpretation (1)

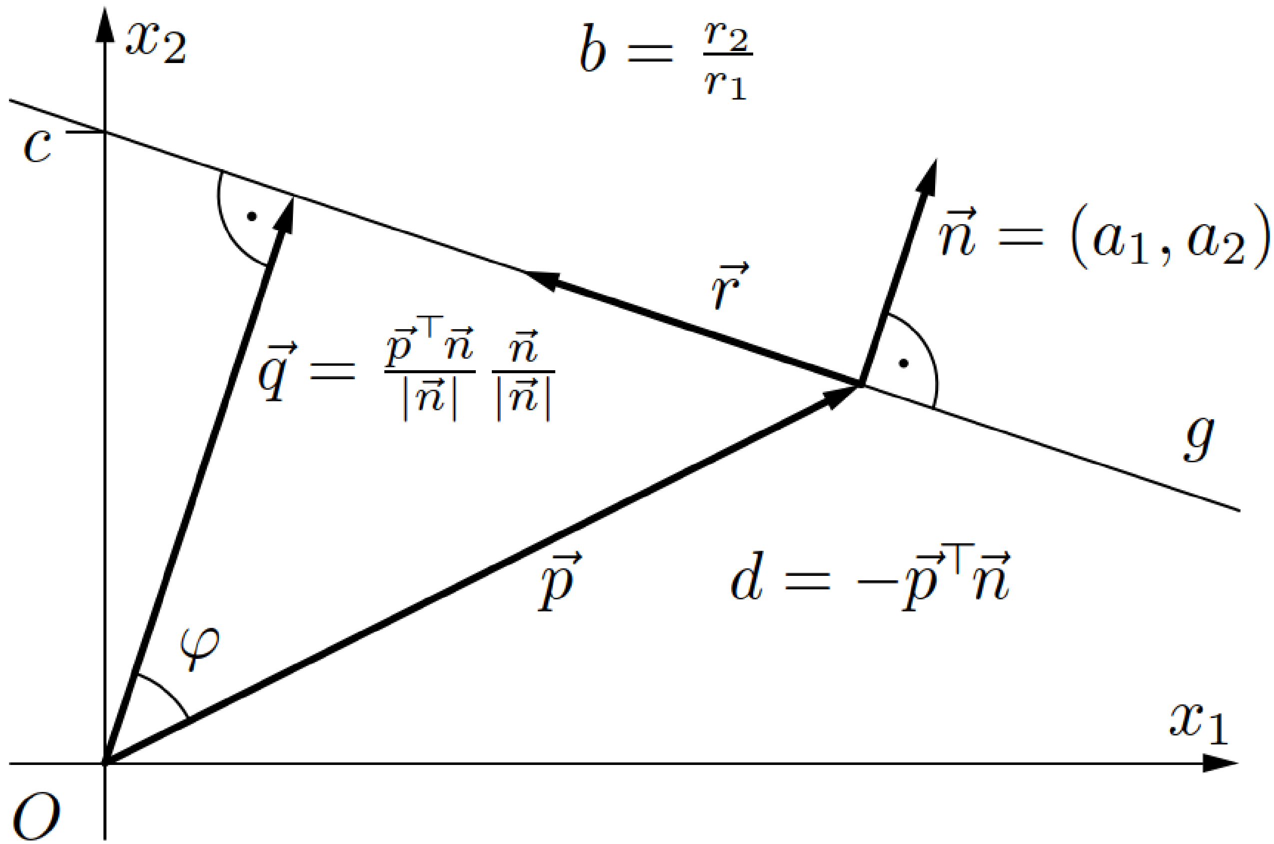
Straight lines are usually represented in one of the following forms:

$$
\begin{array}{lll}
\text{Explicit Form:} & g \equiv & x_2 = bx_1 + c \\
\text{Implicit Form:} & g \equiv & a_1x_1 + a_2x_2 + d = 0 \\
\text{Point-Direction Form:} & g \equiv & \vec{x} = \vec{p} + k\vec{r} \\
\text{Normal Form:} & g \equiv & (\vec{x} - \vec{p})^\top \vec{n} = 0
\end{array}
$$

with parameters:

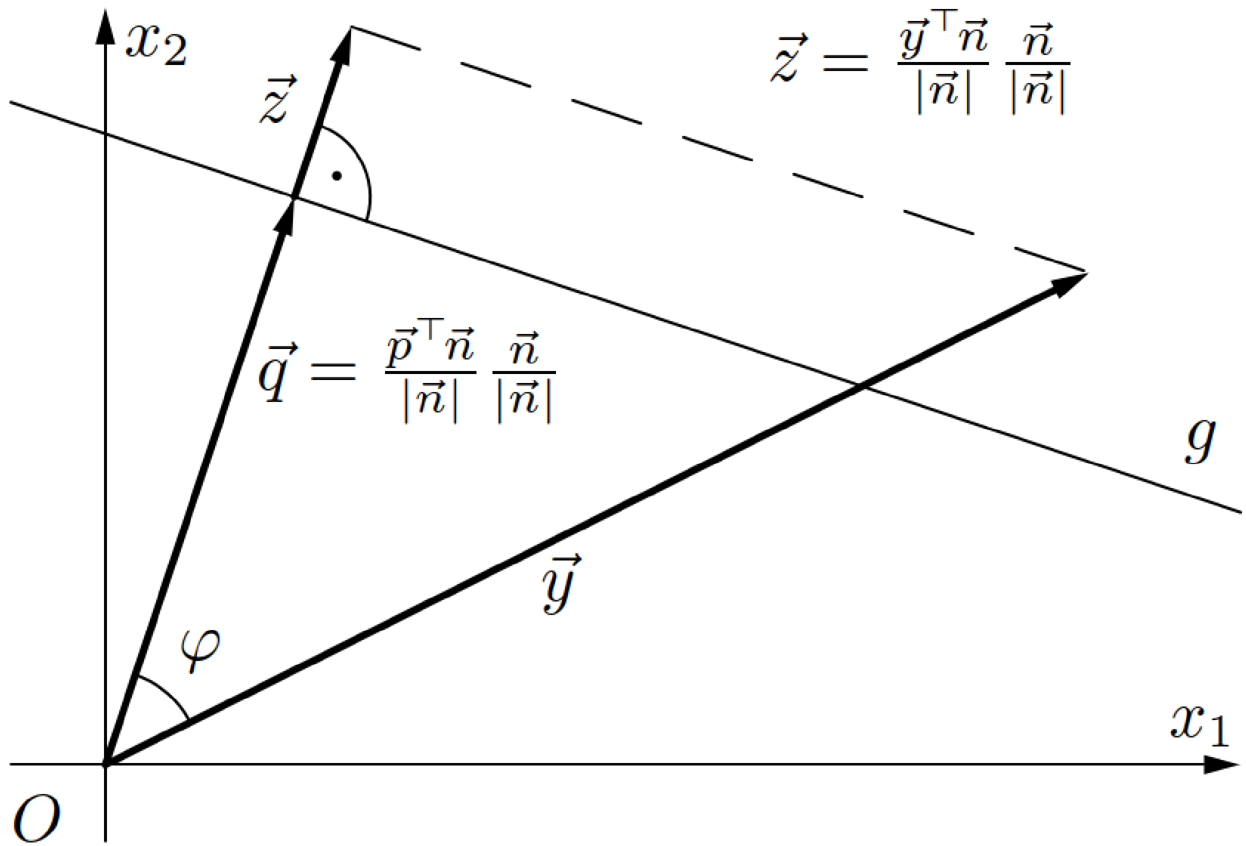
$$
\begin{array}{ll}
b: & \text{Gradient of the line} \\
c: & \text{Section of the } x_2 \text{ axis (intercept)} \\
\vec{p}: & \text{Vector of a point of the line (base vector)} \\
\vec{r}: & \text{Direction vector of the line} \\
\vec{n}: & \text{Normal vector of the line}
\end{array}
$$

Geometric Interpretation (2)



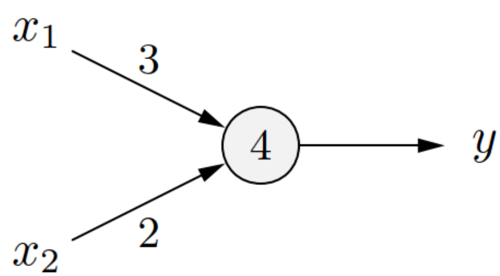
Geometric Interpretation (3)

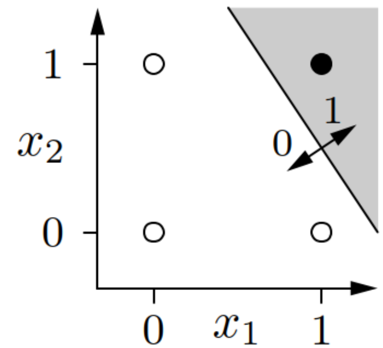
How to determine the side on which a point y lies:



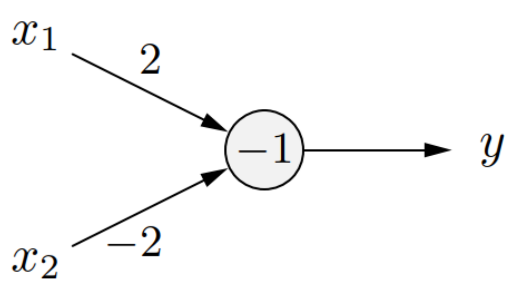
Geometric Interpretation (4)

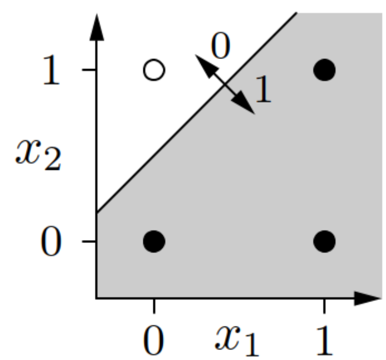
TLU for $x_1 \wedge x_2$



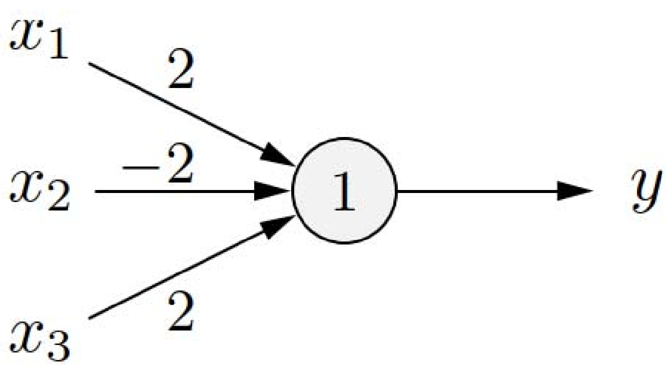


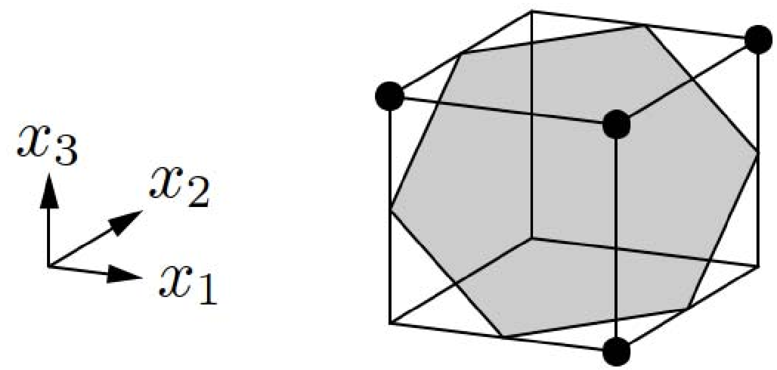
TLU for $x_2 \rightarrow x_1$





Geometric Interpretation (5)



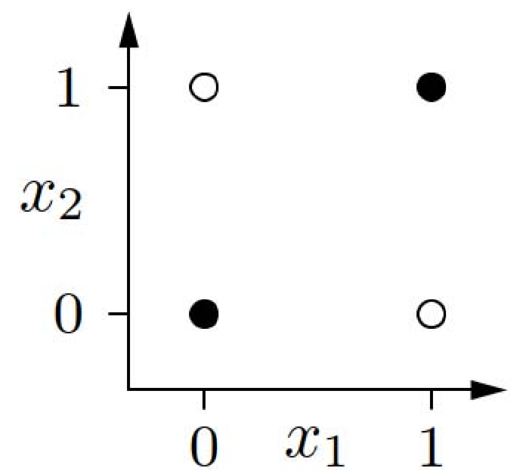


Geometric Interpretation (6)

Biimplication problem $x_1 \leftrightarrow x_2$:

There is no separating line.

x_1	x_2	y
0	0	1
1	0	0
0	1	0
1	1	1



Linear Separability (1)

Two sets of points in a Euclidean space are called linearly separable, iff there exists at least one point, line, plane or hyperplane (depending on the dimension of the Euclidean space), such that all points of the one set lie on one side and all points of the other set lie on the other side of this point, line, plane or hyperplane (or on it). That is, the point sets can be separated by a linear decision function.

Linear Separability (2)

A set of points in a Euclidean space is called **convex** if it is non-empty and connected (that is, if it is a region) and for every pair of points in it every point on the straight line segment connecting the points of the pair is also in the set.

The **convex hull** of a set of points X in a Euclidean space is the smallest convex set of points that contains X. Alternatively, the convex hull of a set of points X is the intersection of all convex sets that contain X.

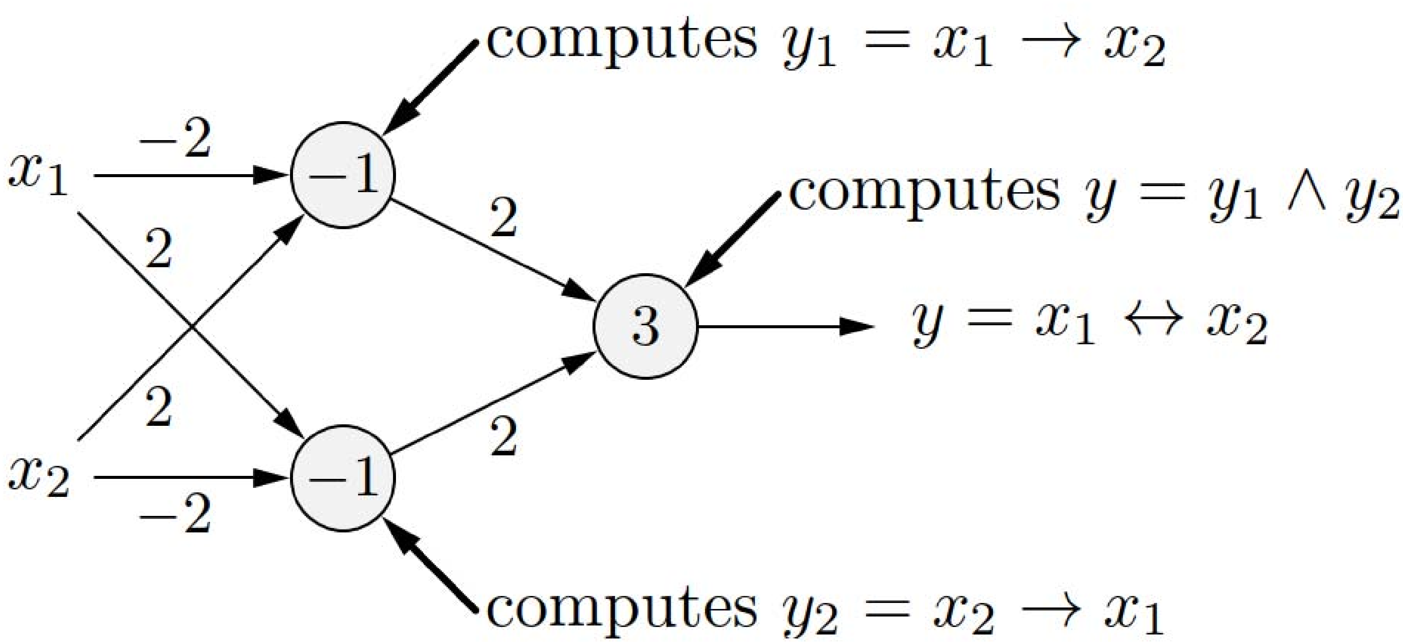
Linear Separability (3)

Two sets of points in a Euclidean space are **linearly separable** if and only if their convex hulls are disjoint (that is, have no point in common).

- For the biimplication problem, the convex hulls are the diagonal line segments.
- They share their intersection point and are thus not disjoint.
- Therefore the biimplication is not linearly separable.

Networks of Threshold Logic Units (1)

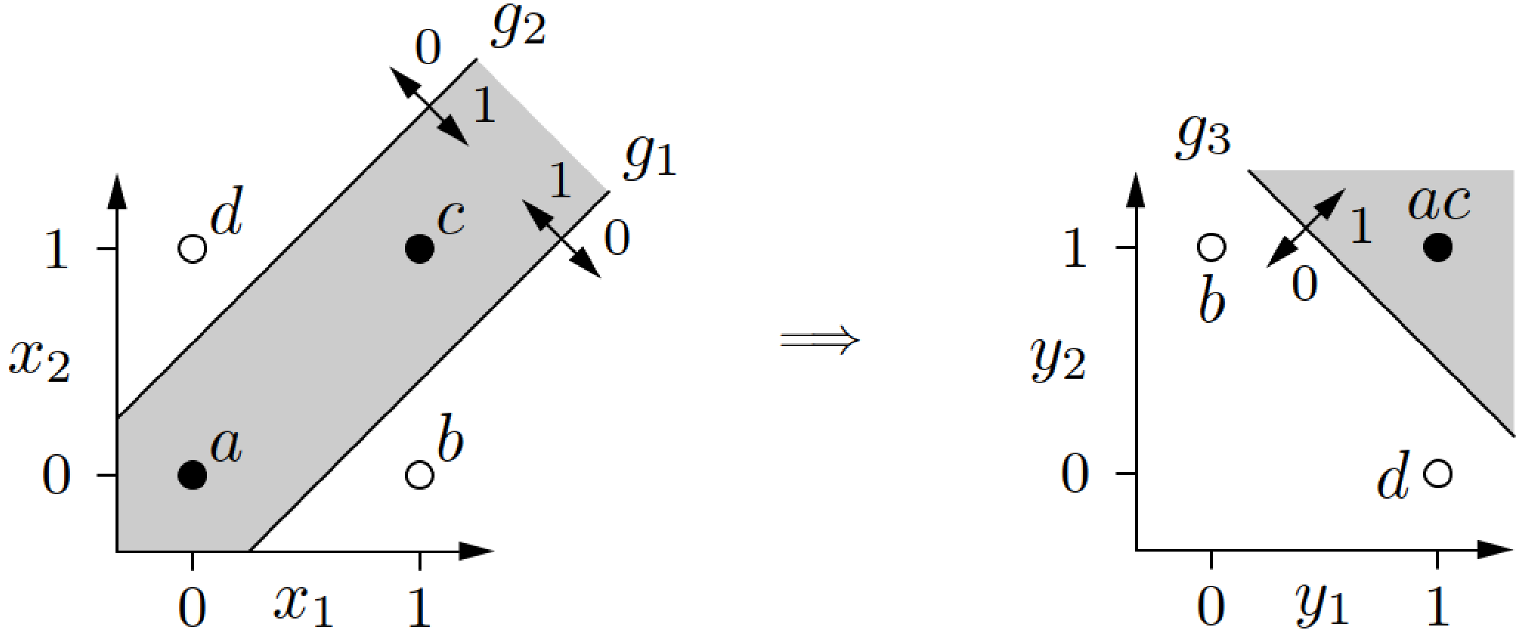
Solving the biimplication problem with a network.



- The first layer computes new Boolean coordinates for the points.

Networks of Threshold Logic Units (2)

- After the coordinate transformation the problem is linearly separable.



Arbitrary Boolean Functions (1)

Boolean function:

x_1	x_2	x_3	y	C_j
0	0	0	0	
1	0	0	1	$x_1 \wedge \overline{x_2} \wedge \overline{x_3}$
0	1	0	0	
1	1	0	0	
0	0	1	0	
1	0	1	0	
0	1	1	1	$\overline{x_1} \wedge x_2 \wedge x_3$
1	1	1	1	$x_1 \wedge x_2 \wedge x_3$

Arbitrary Boolean Functions (2)

Resulting network of threshold logic units: