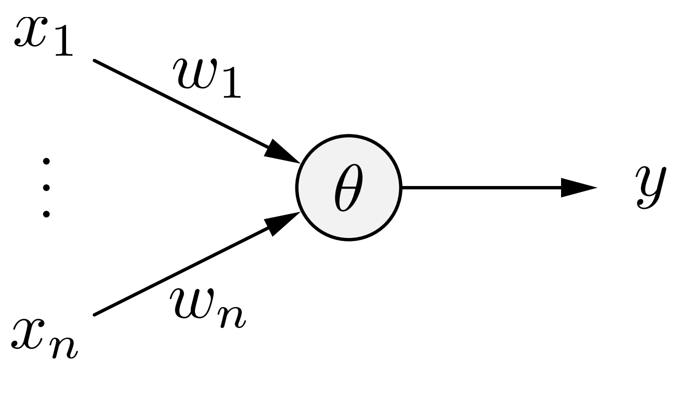
Threshold Logic Units

Threshold Logic Units

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. A threshold logic unit computes the function

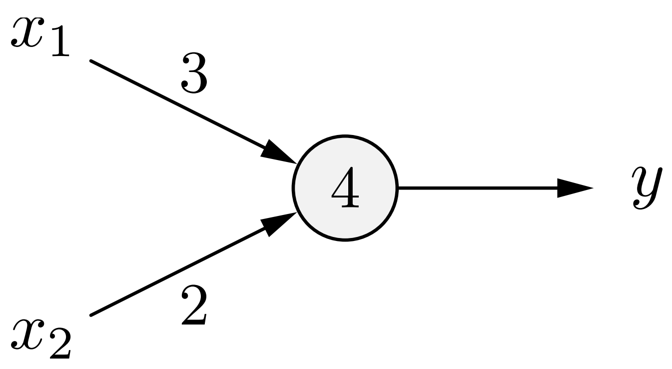
$$y = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i x_i \geq \theta, \\ 0, & \text{otherwise.} \end{cases}$$



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

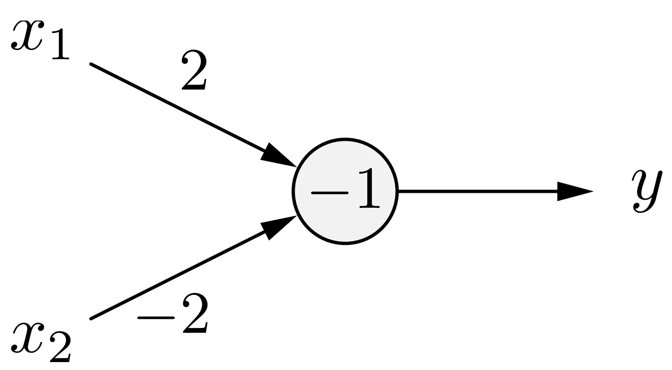
Threshold Logic Units: Examples

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

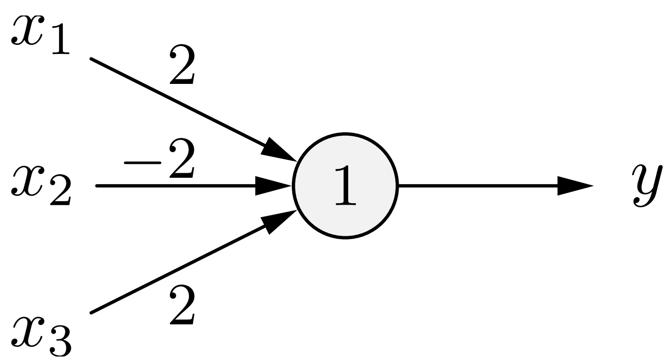
Threshold logic unit for the implication $x_2 \rightarrow x_1$.



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: Examples

Threshold logic unit for $(x_1 \wedge \overline{x_2}) \vee (x_1 \wedge x_3) \vee (\overline{x_2} \wedge x_3)$**.**



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

Rough Intuition:

- Positive weights are analogous to excitatory synapses.
- Negative weights are analogous to inhibitory synapses.

Review of line representations

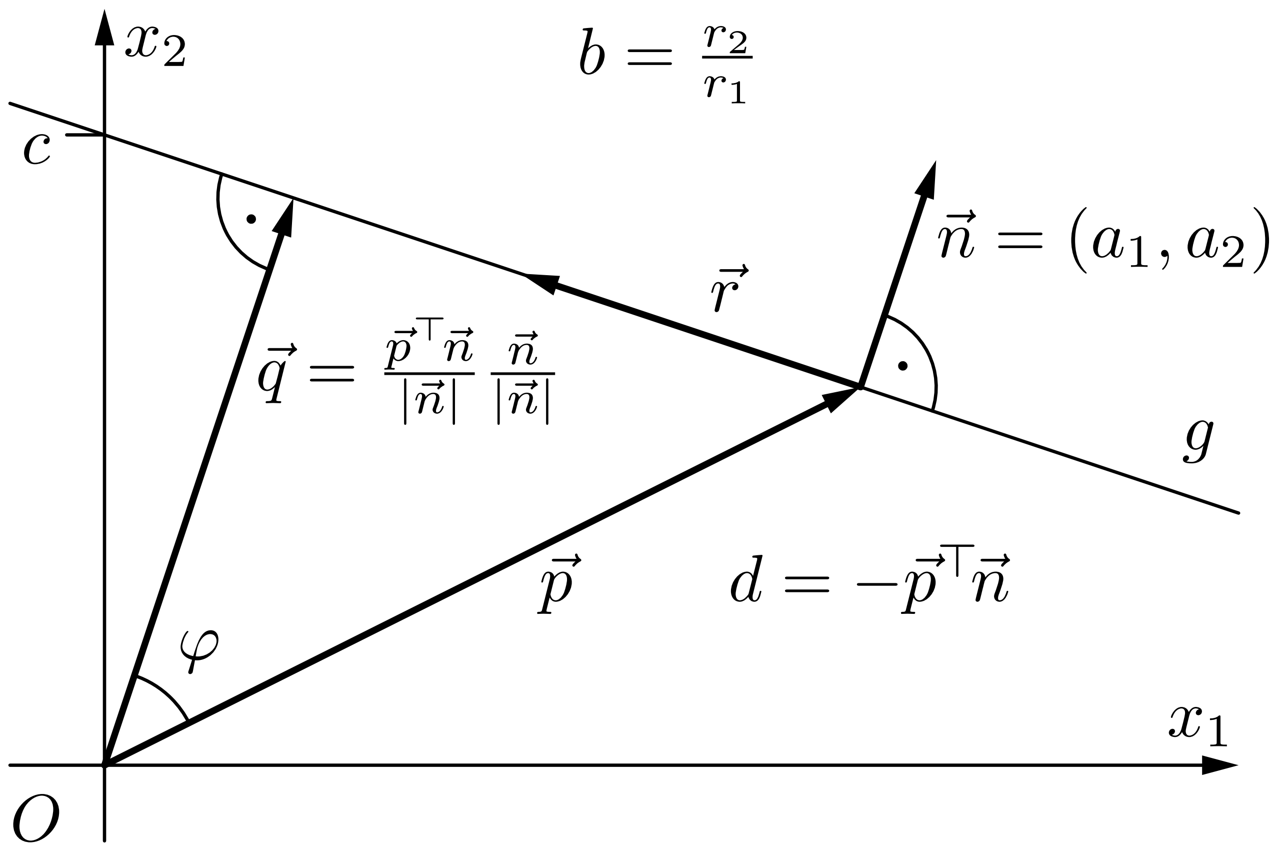
Straight lines are usually represented in one of the following forms:

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

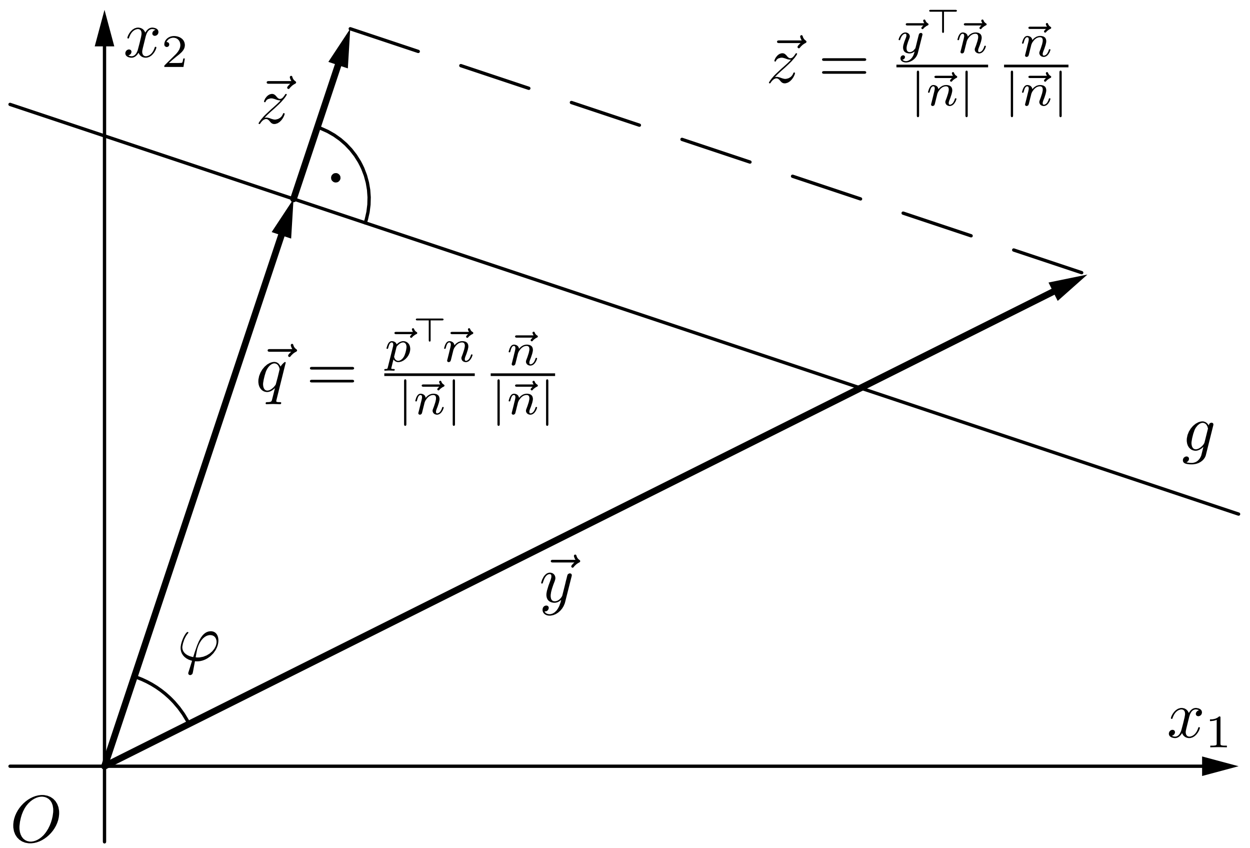
with the parameters:

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

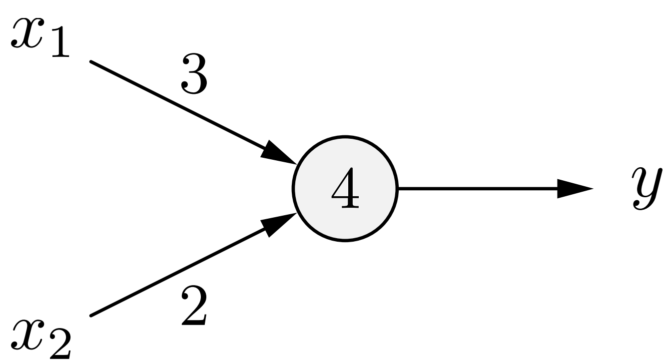
A straight line and its defining parameters:

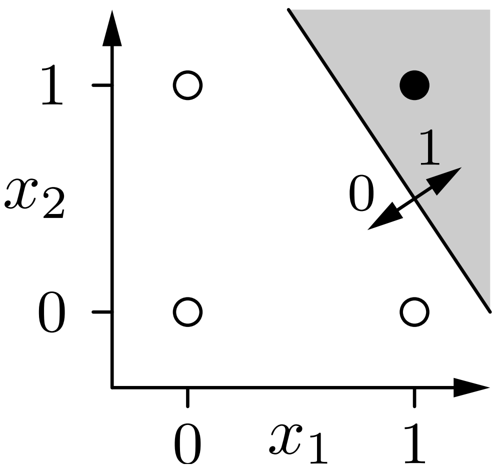


How to determine the side on which a point $\vec{y}$ lies:

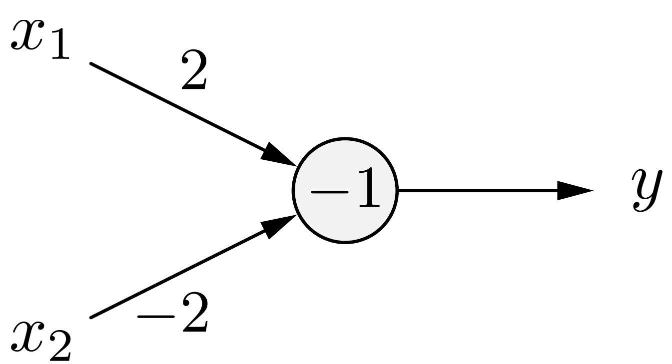


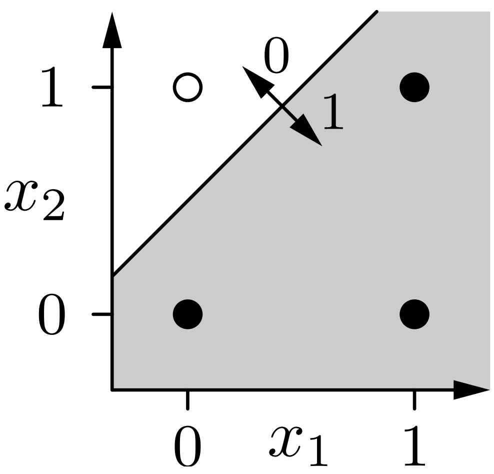
Threshold logic unit for $x_1 \wedge x_2$.





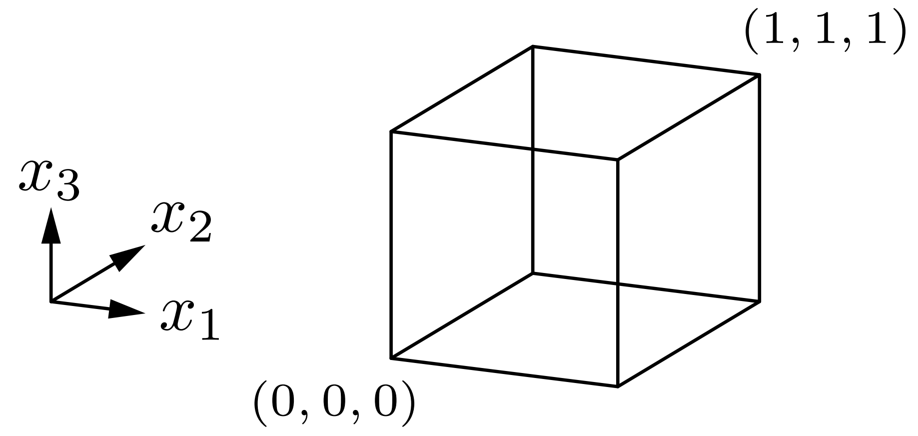
Threshold logic unit for $x_2 \rightarrow x_1$.



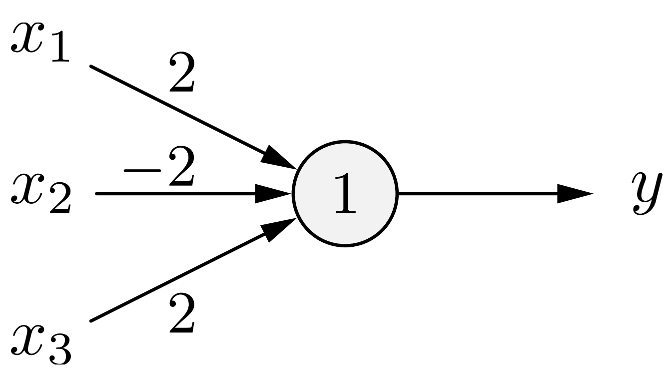


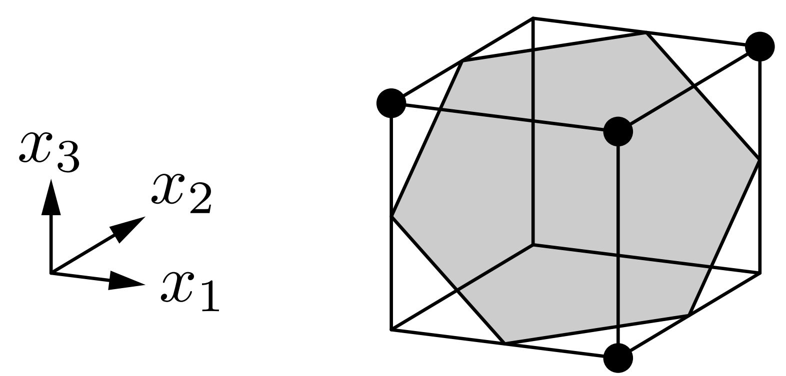
Threshold Logic Units: Geometric Interpretation

Visualization of 3-dimensional Boolean functions:



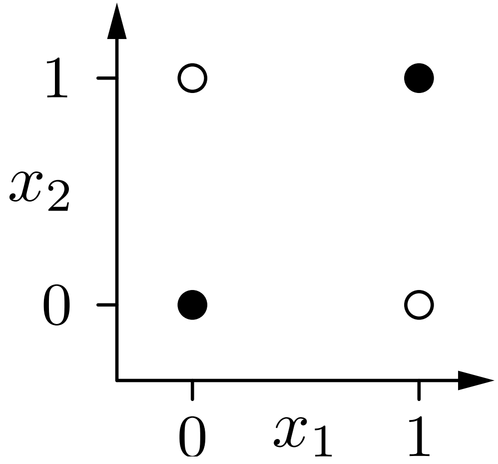
Threshold logic unit for $(x_1 \wedge \overline{x_2}) \vee (x_1 \wedge x_3) \vee (\overline{x_2} \wedge x_3)$**.**





The 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



Formal proof by *reductio ad absurdum*:

$$\begin{array}{llll} \text{since } (0,0) \mapsto 1: & 0 & \geq \theta, & (1) \\ \text{since } (1,0) \mapsto 0: & w_1 & < \theta, & (2) \\ \text{since } (0,1) \mapsto 0: & w_2 & < \theta, & (3) \\ \text{since } (1,1) \mapsto 1: & w_1 + w_2 & \geq \theta. & (4) \end{array}$$

(2) and (3): $w_1 + w_2 < 2\theta$. With (4): $2\theta > \theta$, or $\theta > 0$. Contradiction to (1).

Linear Separability

Definition: 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**. Formally: Two sets $X, Y \subset \mathbb{R}^m$ are linearly separable iff $\vec{w} \in \mathbb{R}^m$ and $\theta \in \mathbb{R}$ exist such that

$$\forall \vec{x} \in X : \quad \vec{w}^\top \vec{x} < \theta \qquad \text{and} \qquad \forall \vec{y} \in Y : \quad \vec{w}^\top \vec{y} \geq \theta.$$

- **Boolean functions** define two points sets, namely the set of points that are mapped to the function value 0 and the set of points that are mapped to 1.
  $\Rightarrow$ The term "linearly separable" can be transferred to Boolean functions.

- As we have seen, **conjunction** and **implication** are **linearly separable** (as are **disjunction**, NAND, NOR etc.).

- The **biimplication** is **not linearly separable** (and neither is the **exclusive or** (XOR)).

Linear Separability

Definition: 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.

Definition: 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.

Theorem: 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.

Threshold Logic Units: Limitations

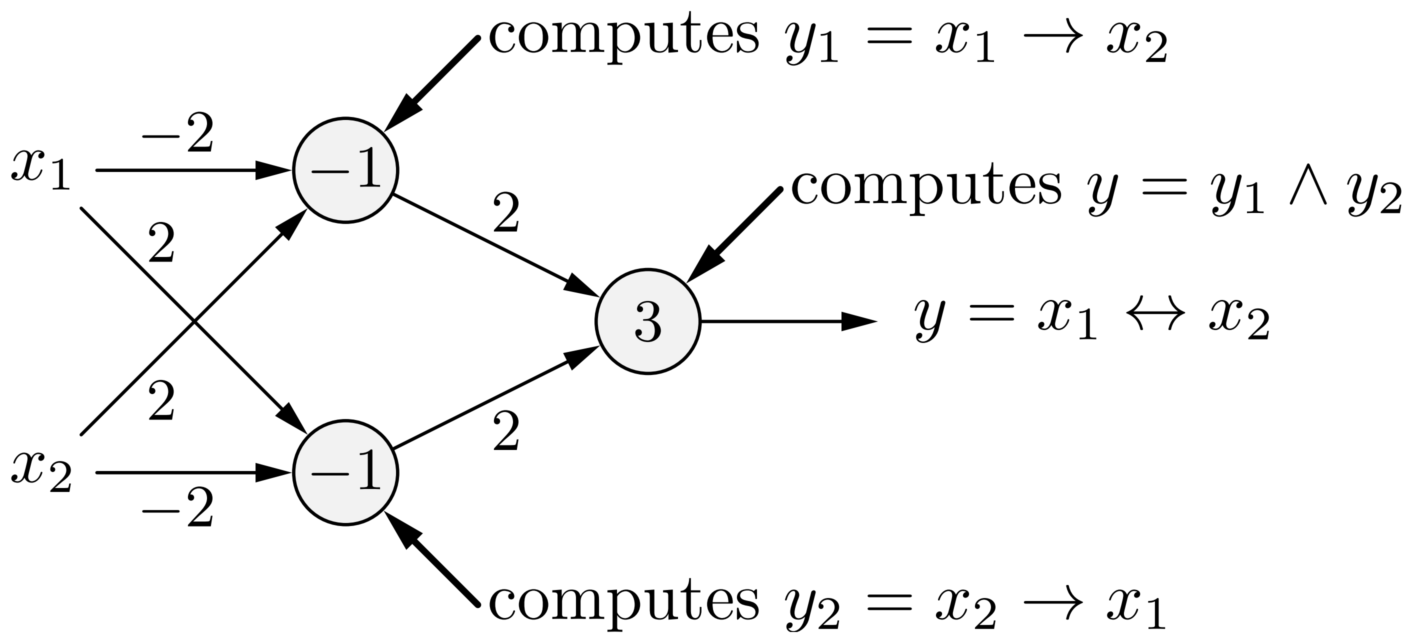
Total number and number of linearly separable Boolean functions
(On-Line Encyclopedia of Integer Sequences, `oeis.org`, A001146 and A000609):

inputs	Boolean functions	linearly separable functions
1	4	4
2	16	14
3	256	104
4	65,536	1,882
5	4,294,967,296	94,572
6	18,446,744,073,709,551,616	15,028,134
n	$2^{(2^n)}$	no general formula known

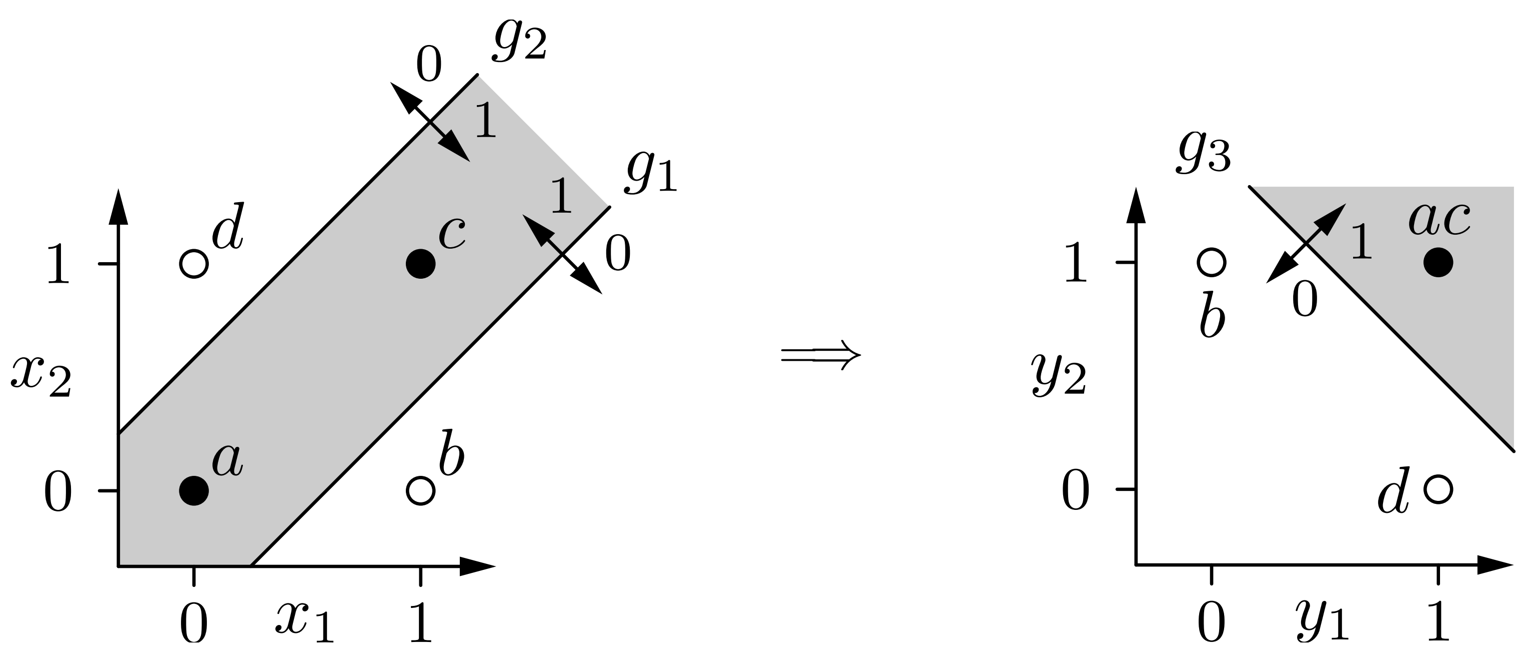
- For many inputs a threshold logic unit can compute almost no functions.
- Networks of threshold logic units are needed to overcome the limitations.

Solving the biimplication problem with a network.

Idea: logical decomposition $\quad x_1 \leftrightarrow x_2 \;\equiv\; (x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_1)$



Solving the biimplication problem: Geometric interpretation



- The first layer computes new Boolean coordinates for the points.
- After the coordinate transformation the problem is linearly separable.

Representing Arbitrary Boolean Functions

Algorithm: Let $y = f(x_1, \ldots, x_n)$ be a Boolean function of n variables.

(i) Represent the given function $f(x_1, \ldots, x_n)$ in disjunctive normal form. That is, determine $D_f = C_1 \lor \ldots \lor C_m$, where all C_j are conjunctions of n literals, that is, $C_j = l_{j1} \land \ldots \land l_{jn}$ with $l_{ji} = x_i$ (positive literal) or $l_{ji} = \neg x_i$ (negative literal).

(ii) Create a neuron for each conjunction C_j of the disjunctive normal form (having n inputs — one input for each variable), where

$$w_{ji} = \begin{cases} 2, & \text{if } l_{ji} = x_i, \\ -2, & \text{if } l_{ji} = \neg x_i, \end{cases} \qquad \text{and} \qquad \theta_j = n - 1 + \frac{1}{2}\sum_{i=1}^{n} w_{ji}.$$

(iii) Create an output neuron (having m inputs — one input for each neuron that was created in step (ii)), where

$$w_{(n+1)k} = 2, \quad k = 1, \ldots, m, \qquad \text{and} \qquad \theta_{n+1} = 1.$$

Remark: weights are set to ± 2 instead of ± 1 in order to ensure integer thresholds.

Representing Arbitrary Boolean Functions

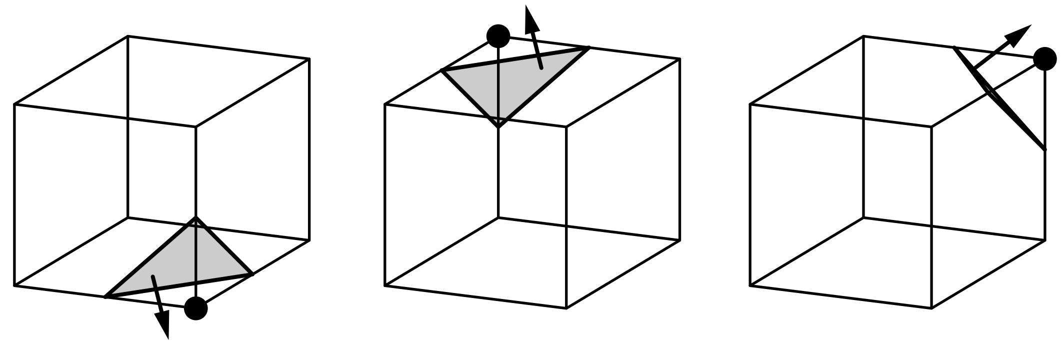
Example:
ternary Boolean function:

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	$x_1 \wedge x_2 \wedge x_3$

$D_f = C_1 \vee C_2 \vee C_3$

One conjunction for each row where the output y is 1 with literals according to input values.

First layer (conjunctions):

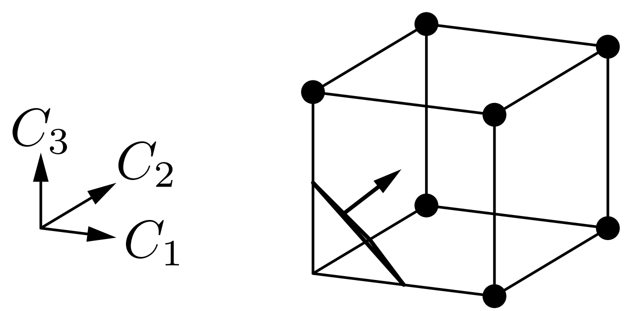


$C_1 = x_1 \wedge \overline{x_2} \wedge \overline{x_3}$

$C_2 = \overline{x_1} \wedge x_2 \wedge x_3$

$C_3 = x_1 \wedge x_2 \wedge x_3$

Second layer (disjunction):



$D_f = C_1 \vee C_2 \vee C_3$

Representing Arbitrary Boolean Functions

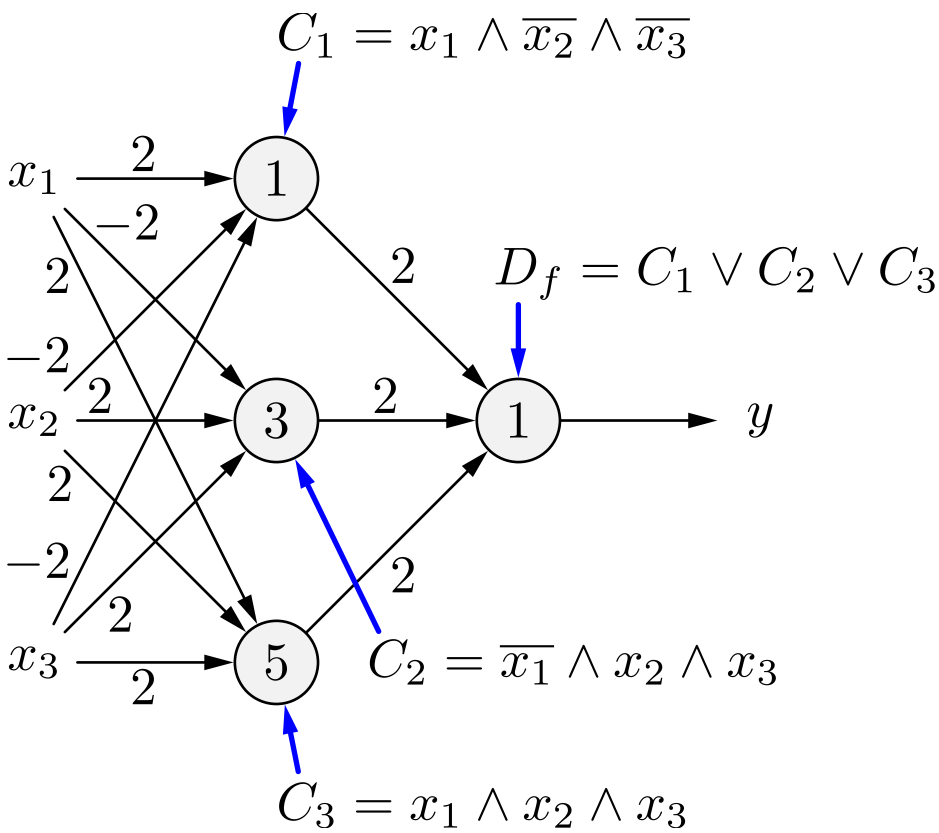
Example:
ternary Boolean function:

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	$x_1 \wedge x_2 \wedge x_3$

$D_f = C_1 \vee C_2 \vee C_3$

One conjunction for each row where the output y is 1 with literals according to input value.

Resulting network of threshold logic units:

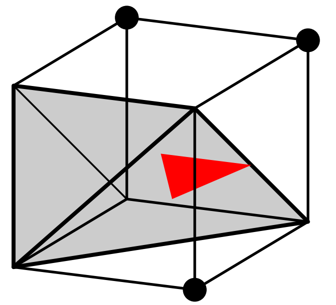


Reminder: Convex Hull Theorem

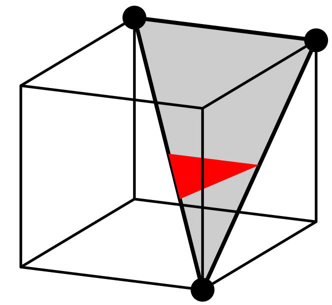
Theorem: Two sets of points in a Euclidean space are **linearly separable** if and only if their convex hulls are disjoint (that is, have no point in common).

Example function on the preceding slide:

$$y = f(x_1, x_2, x_3) = (x_1 \wedge \overline{x_2} \wedge \overline{x_3}) \vee (\overline{x_1} \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_3)$$



Convex hull of points with $y = 0$



Convex hull of points with $y = 1$

- The convex hulls of the two point sets are not disjoint (red: intersection).
- Therefore the function $y = f(x_1, x_2, x_3)$ is not linearly separable.