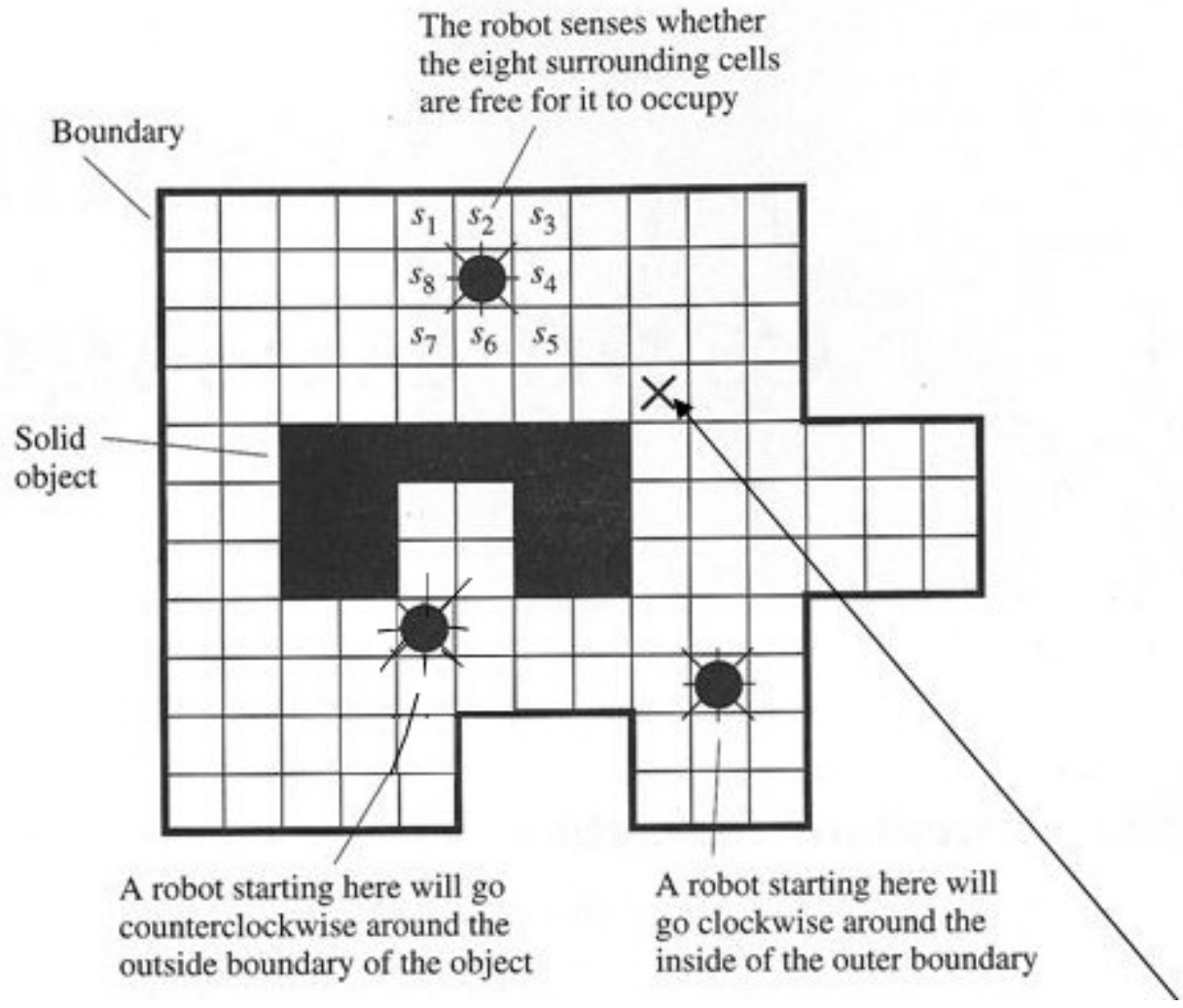
# Agenti stimolo - risposta

**Edmondo Trentin (DIISM)**

# Il robot nel mondo a griglia di Nils Nilsson




$s1 = 0, s_2 = 0, s_3 = 0, s_4 = 0,$
$s_5 = 0, s_6 = 0, s_7 = 1, s_8 = 0$

Task: raggiungere un muro (o un oggetto)
e costeggiarlo all'infinito. *[Nota: nella griglia non ci sono spazi stretti]*

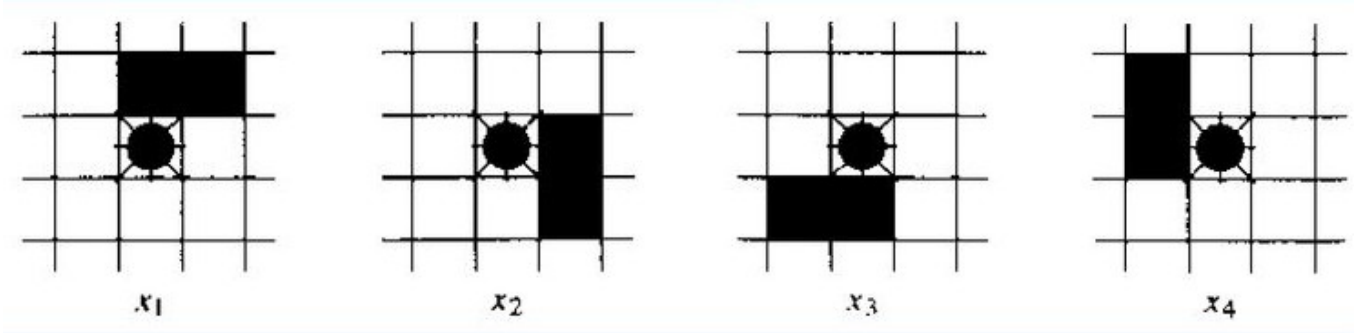
# Percezione sensoriale del mondo

Costruiamo il robot come una macchina dotata di **sensori** $s_1, \ldots, s_8$ che consentono una certa **percezione** del mondo.
*Esempi*: radar, sonar, telecamere, sensori tattili, ...

- Assumiamo che il segnale acquisito dai sensori sia rappresentabile in forma adeguata per l'elaborazione da parte della macchina, ad es. attraverso la **digitalizzazione** (es. $s_1 = 0, s_2 = 1, \ldots, s_8 = 0$)
- Da questa rappresentazione sensoriale del mondo si estrae un **vettore $\boldsymbol{x}$ di feature** (o "attributi"), es. $\boldsymbol{x} = (0, 1, 1, 0)$.
- $\boldsymbol{x}$ è **definito a monte** dal progettista del sistema e dipende dal task. Se ben definite, le feature possiedono due caratteristiche:
  1. contengono l'informazione necessaria perché la macchina possa espletare il proprio task
  2. sono le più semplici e in minor numero possibile.
- Le feature sono di due tipi principali:
  1. *Numeriche* (es. 0, 1; oppure 0.325, -2.144, ...)
  2. *Categoriche*, i.e., simboliche (es. "A", "B", ...; oppure "0","1"; oppure "pieno", "vuoto", ..., "rosso", "verde", ...)

# Percezione: esempio di feature nel mondo a griglia

Un vettore 4-dimensionale di feature $\boldsymbol{x} = (x_1, x_2, x_3, x_4)$ è estratto dagli 8 stimoli sensoriali $s_1, s_2, \ldots, s_8$




Tenendo presente il connettivo logico $\vee$ (cioé *OR*), le feature possono essere formalmente definite come:

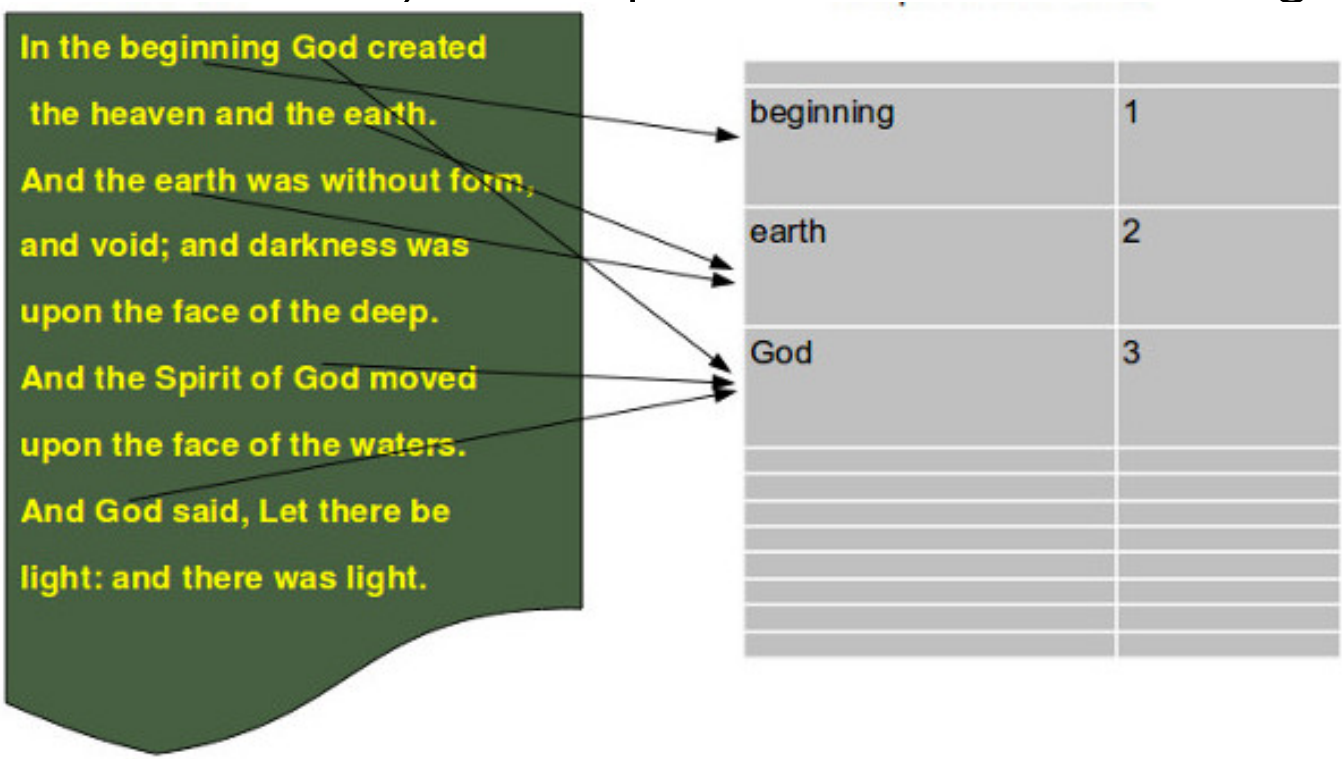
- $x_1 = s_2 \vee s_3$
- $x_2 = s_4 \vee s_5$
- $x_3 = s_6 \vee s_7$
- $x_4 = s_8 \vee s_1$

In pratica, ogni feature vale 1 se attorno al robot è presente un muro o un oggetto in almeno una delle due cellette nere disegnate nella corrispondente figura, altrimenti vale 0.

Queste feature saranno sufficienti al robot per compiere il task di navigazione.

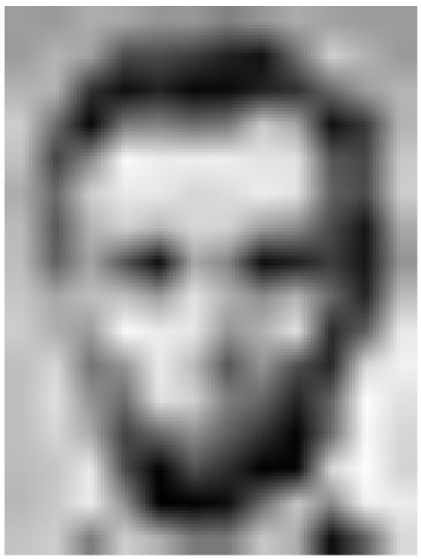
# Percezione: feature per testi letterari e documenti

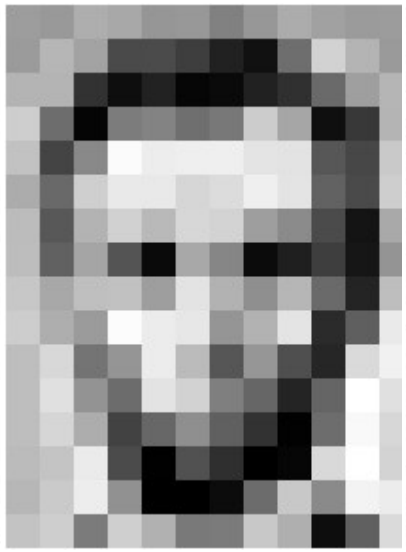
- Un testo è già rappresentato in termini di feature categoriche, nella fattispecie i simboli dell'alfabeto che lo compongono:
  *"N" "e" "l" " " " " "m" "e" "z" "z" "o" " " " " "d" "e" "l" ...*
  Questa rappresentazione è però molto pesante e ridondante. La sua dimensione (numero di feature) varia in base al documento.
- Alternativa: la *bag of words* è una collezione ordinata e "snella" di feature numeriche avente dimensione prefissata (pari al numero di parole nel dizionario). Perde parte dell'informazione originaria:



# Percezione: feature per Visione Artificiale

Da un'immagine digitalizzata ...





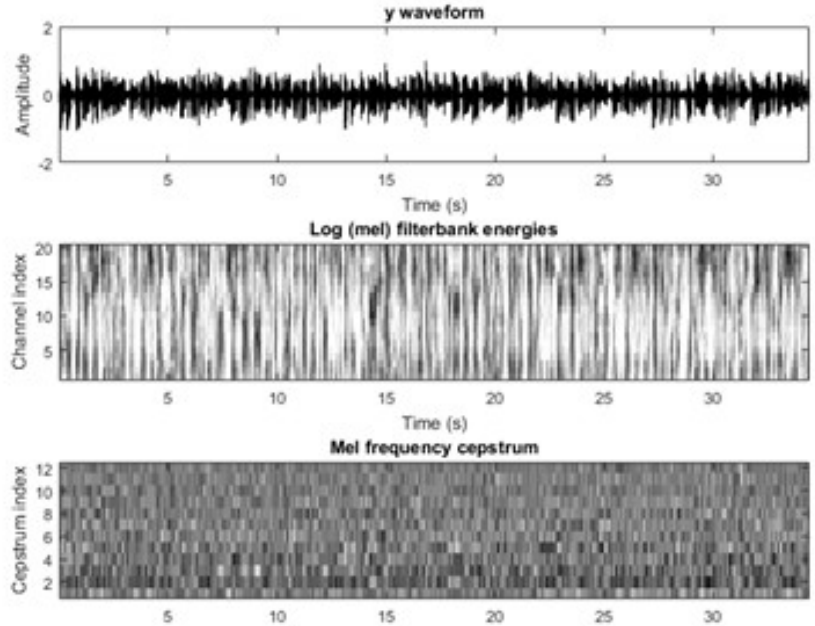
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 157 | 153 | 174 | 168 | 150 | 152 | 129 | 151 | 172 | 161 | 155 | 156 |
| 155 | 182 | 163 | 74 | 75 | 62 | 33 | 17 | 110 | 210 | 180 | 154 |
| 180 | 180 | 50 | 14 | 84 | 6 | 10 | 33 | 48 | 106 | 159 | 181 |
| 206 | 109 | 5 | 124 | 131 | 111 | 120 | 204 | 166 | 15 | 56 | 180 |
| 194 | 68 | 137 | 251 | 237 | 239 | 239 | 228 | 227 | 87 | 71 | 201 |
| 172 | 105 | 207 | 233 | 233 | 214 | 220 | 239 | 228 | 98 | 74 | 206 |
| 188 | 88 | 179 | 209 | 185 | 215 | 211 | 158 | 139 | 75 | 20 | 169 |
| 189 | 97 | 165 | 84 | 10 | 168 | 134 | 11 | 31 | 62 | 22 | 148 |
| 199 | 168 | 191 | 193 | 158 | 227 | 178 | 143 | 182 | 106 | 36 | 190 |
| 205 | 174 | 155 | 252 | 236 | 231 | 149 | 178 | 228 | 43 | 95 | 234 |
| 190 | 216 | 116 | 149 | 236 | 187 | 85 | 150 | 79 | 38 | 218 | 241 |
| 190 | 224 | 147 | 108 | 227 | 210 | 127 | 102 | 36 | 101 | 255 | 224 |
| 190 | 214 | 173 | 66 | 103 | 143 | 96 | 50 | 2 | 109 | 249 | 215 |
| 187 | 196 | 235 | 75 | 1 | 81 | 47 | 0 | 6 | 217 | 255 | 211 |
| 183 | 202 | 237 | 145 | 0 | 0 | 12 | 108 | 200 | 138 | 243 | 236 |
| 195 | 206 | 123 | 207 | 177 | 121 | 123 | 200 | 175 | 13 | 96 | 218 |


| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 157 | 153 | 174 | 168 | 150 | 152 | 129 | 151 | 172 | 161 | 155 | 156 |
| 155 | 182 | 163 | 74 | 75 | 62 | 33 | 17 | 110 | 210 | 180 | 154 |
| 180 | 180 | 50 | 14 | 34 | 6 | 10 | 33 | 48 | 106 | 159 | 181 |
| 206 | 109 | 5 | 124 | 131 | 111 | 120 | 204 | 166 | 15 | 56 | 180 |
| 194 | 68 | 137 | 251 | 237 | 239 | 239 | 228 | 227 | 87 | 71 | 201 |
| 172 | 105 | 207 | 233 | 233 | 214 | 220 | 239 | 228 | 98 | 74 | 206 |
| 188 | 88 | 179 | 209 | 185 | 215 | 211 | 158 | 139 | 75 | 20 | 169 |
| 189 | 97 | 165 | 84 | 10 | 168 | 134 | 11 | 31 | 62 | 22 | 148 |
| 199 | 168 | 191 | 193 | 158 | 227 | 178 | 143 | 182 | 106 | 36 | 190 |
| 205 | 174 | 155 | 252 | 236 | 231 | 149 | 178 | 228 | 43 | 95 | 234 |
| 190 | 216 | 116 | 149 | 236 | 187 | 86 | 150 | 79 | 38 | 218 | 241 |
| 190 | 224 | 147 | 108 | 227 | 210 | 127 | 102 | 36 | 101 | 255 | 224 |
| 190 | 214 | 173 | 66 | 103 | 143 | 96 | 50 | 2 | 109 | 249 | 215 |
| 187 | 196 | 235 | 75 | 1 | 81 | 47 | 0 | 6 | 217 | 255 | 211 |
| 183 | 202 | 237 | 145 | 0 | 0 | 12 | 108 | 200 | 138 | 243 | 236 |
| 195 | 206 | 123 | 207 | 177 | 121 | 123 | 200 | 175 | 13 | 96 | 218 |

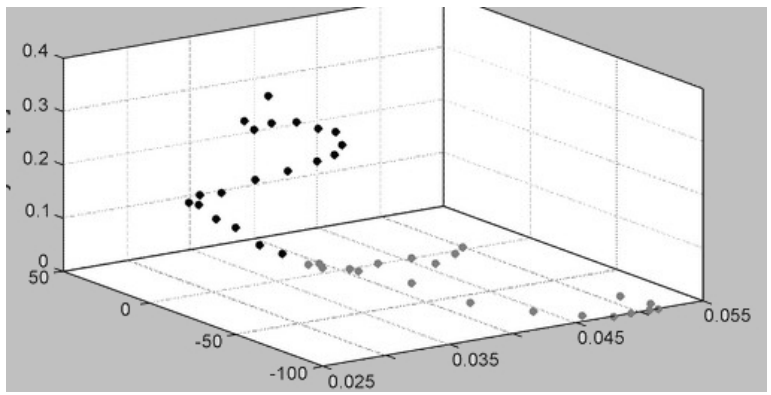

... estraggo una matrice di feature numeriche.

# Percezione: feature per riconoscimento del parlato

Dalla forma d'onda del segnale acustico ...




... estraggo una "traiettoria" di vettori di feature numeriche:



# Dalla percezione all'azione

Basandosi sui dati sensoriali (cioé sulle feature) il robot deve decidere quale azione intraprendere per aggredire il proprio task.

Assumiamo che al robot siano possibili queste 4 azioni:

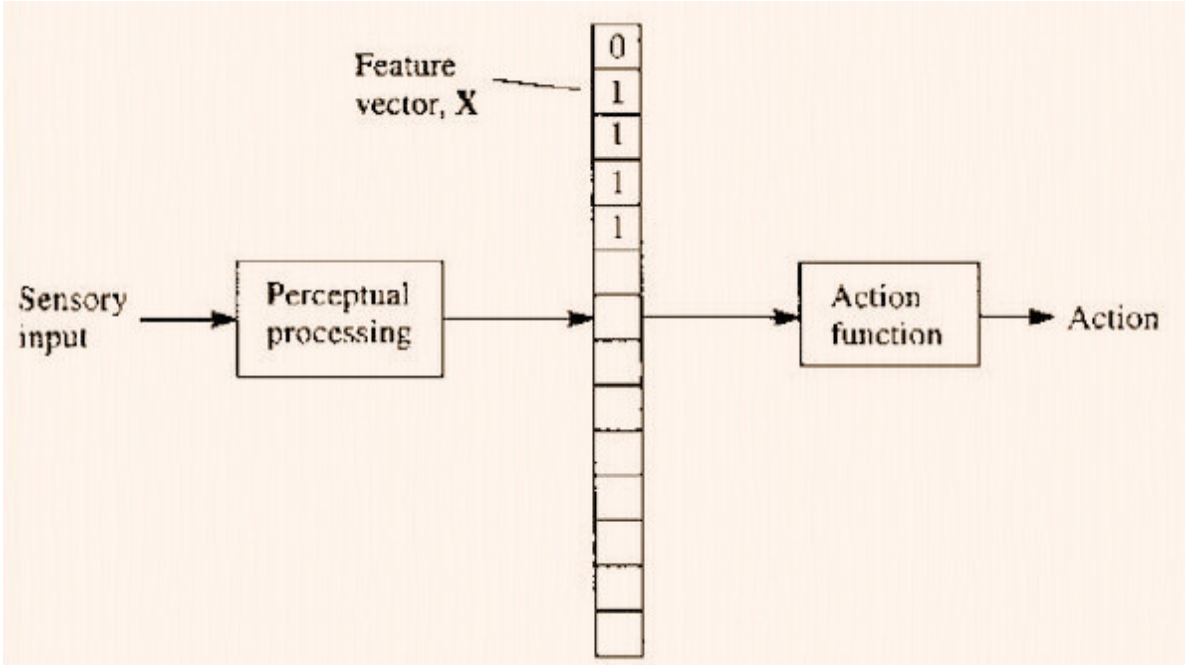
1. **nord**: muove il robot in su di una cella nella griglia
2. **est**: muove il robot a destra di una cella
3. **sud**: muove il robot in giù di una cella
4. **ovest**: muove il robot a sinistra di una cella

Ogni azione può essere tentata in ogni momento, ma se nella corrispondente direzione c'è una parete o un ostacolo essa non produce esiti (il robot resta fermo).

Formalmente, il robot deve quindi applicare una **funzione** (i.e., regola di azione o di **decisione**) che ha per dominio l'insieme $\{(0,0,0,1),(0,0,1,0),(0,0,1,1),\ldots\}$ dei *possibili* vettori delle feature e per codominio l'insieme delle possibili azioni: {**nord**, **est**, **sud**, **ovest**}.

# Agente stimolo - risposta

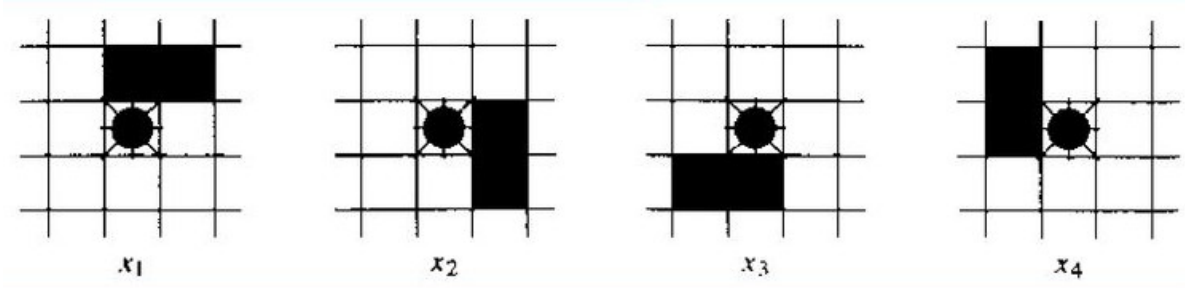
Reagisce immediatamente ad ogni stimolo sensoriale in input, generando un'univoca risposta (l'azione corrispondente)




• A differenza della MdT, **è una macchina senza stati interni** (cioé senza memoria): risponderà sempre allo stesso modo a fronte di un dato stimolo.

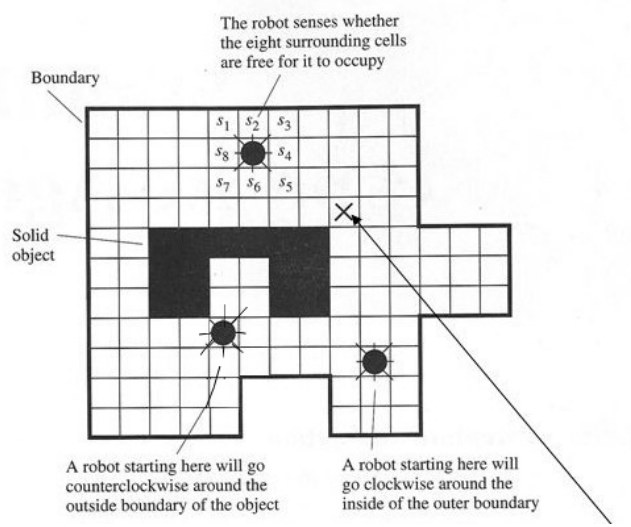
• Si può anche pensare che sia alla stregua di una **MdT con un solo stato interno**, che produce l'output in funzione solo dell'input.

# Azioni del robot nel mondo a griglia




Non essendoci *spazi stretti*, le seguenti regole di azione consentono al robot di realizzare il task di navigazione (*Esercizio*: verificalo):

(1) se $x_1 = 1$ e $x_2 = 0$ allora **est**
(2) se $x_2 = 1$ e $x_3 = 0$ allora **sud**
(3) se $x_3 = 1$ e $x_4 = 0$ allora **ovest**
(4) se $x_4 = 1$ e $x_1 = 0$ allora **nord**



# Esplicitazione delle azioni: sistemi di produzione

Un **sistema di produzione** ha la seguente forma:

$$c_1 \to a_1$$
$$c_2 \to a_2$$
$$\ldots$$
$$c_m \to a_m$$

dove $c_i$ esprime una **condizione** logica (può valere 0 o 1) **definita sul valore delle feature**, e $a_i$ è la corrispondente azione. La scrittura $c_i \to a_i$ si dice **regola di produzione** (o *produzione*).

- La macchina opera seguendo il tic-tac di un *orologio interno*. Ad ogni tic-tac essa scorre la lista delle produzioni dall'alto in basso.
- Appena trova una produzione $c_i \to a_i$ la cui $c_i$ vale 1 (cioè, è *vera*) la macchina attua $a_i$.
- Se nessuna delle $c_1, \ldots, c_m$ vale 1, la macchina non fa nulla e attende che agli istanti successivi il mondo attorno a essa muti quanto basti per alterare il valore di qualche condizione $c_j$. Se il mondo è immutabile, la macchina resta in stallo perpetuo.

# Esplicitazione delle azioni: sistemi di produzione

- **Se si vuole che la macchina si fermi definitivamente** al verificarsi di una certa condizione "finale"[1] $c_f$ perché essa a quel punto ha centrato il proprio **goal**, si usa la speciale produzione

  $c_f \rightarrow nil$

- Un sistema di produzione $\mathcal{S}$ può essere usato come **meta-azione** al verificarsi di una data condizione $c_k$ dentro un altro sistema di produzione, includendo in questo una produzione avente forma

  $c_k \rightarrow \mathcal{S}$

- È possibile specificare una **azione di default** $a_d$ da eseguire quando nessuna delle condizioni $c_1, \ldots, c_m$ sia vera, aggiungendo in coda al sistema la regola di produzione $1 \rightarrow a_d$:

  $c_1 \rightarrow a_1$

  $\ldots$

  $c_m \rightarrow a_m$

  $1 \rightarrow a_d$

[1] Nota l'analogia con lo "stato finale" della MdT.

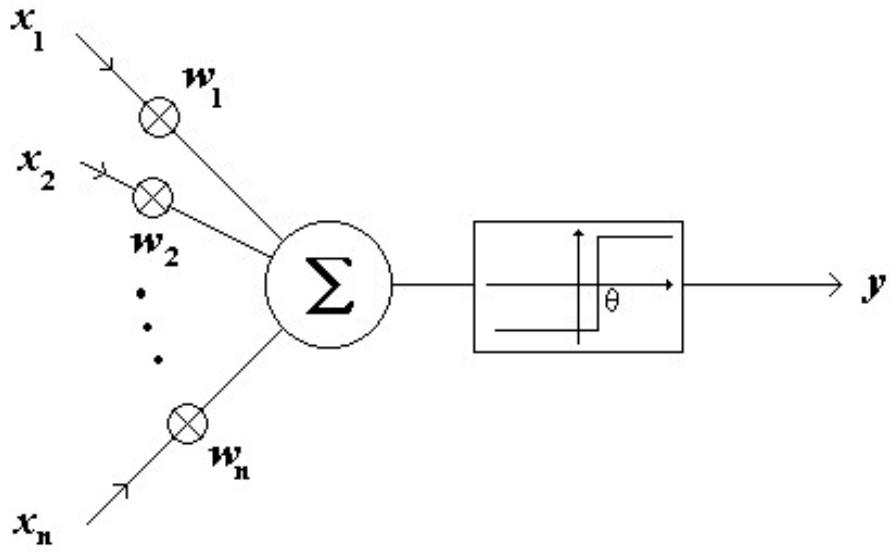
# Esplicitazione delle azioni: sistemi di produzione

Nell'esempio della navigazione robotica nel mondo a griglia un sistema di produzione che espliciti la strategia di realizzazione del task è dunque il seguente:

$$x_4 \wedge \neg x_1 \rightarrow \textbf{nord}$$
$$x_1 \wedge \neg x_2 \rightarrow \textbf{est}$$
$$x_2 \wedge \neg x_3 \rightarrow \textbf{sud}$$
$$x_3 \wedge \neg x_4 \rightarrow \textbf{ovest}$$
$$1 \rightarrow \textbf{nord}$$

dove l'azione di default, in assenza di adeguati contatti con il perimetro della griglia o di un oggetto, è quella di procedere verso l'alto di una cella.

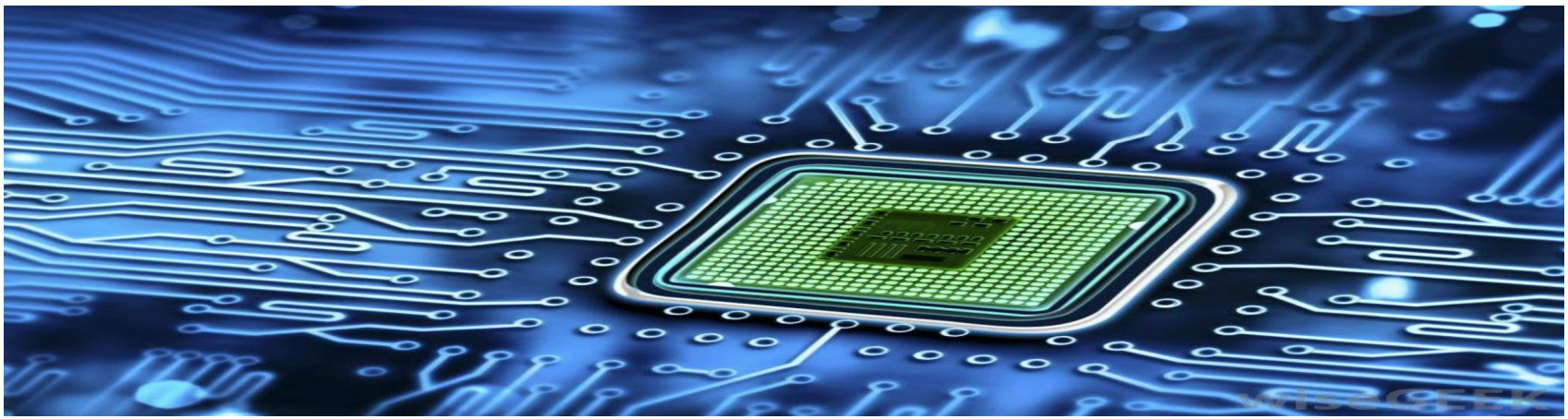
# Esplicitazione delle azioni: unità logiche a soglia




- $x_1, \ldots, x_n$ sono le feature di input
- $w_1, \ldots, w_n$ sono numeri reali ($w_i \in \mathbb{R}$) che determinano il contributo che ogni feature dà al comportamento della macchina
- il simbolo $\bigotimes$ rappresenta il prodotto (ad es. tra $x_i$ e $w_i$)
- il circuito etichettato "$\sum$" restituisce la somma $\sum_{j=1}^{n} w_j x_j$ delle feature pesate, cioé $w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$
- il circuito successivo (box rettangolare) applica una funzione a gradino con soglia $\theta$, cioé restituisce $y = 1$ se $\sum_{j=1}^{n} w_j x_j \geq \theta$ e $y = 0$ se $\sum_{j=1}^{n} w_j x_j < \theta$.

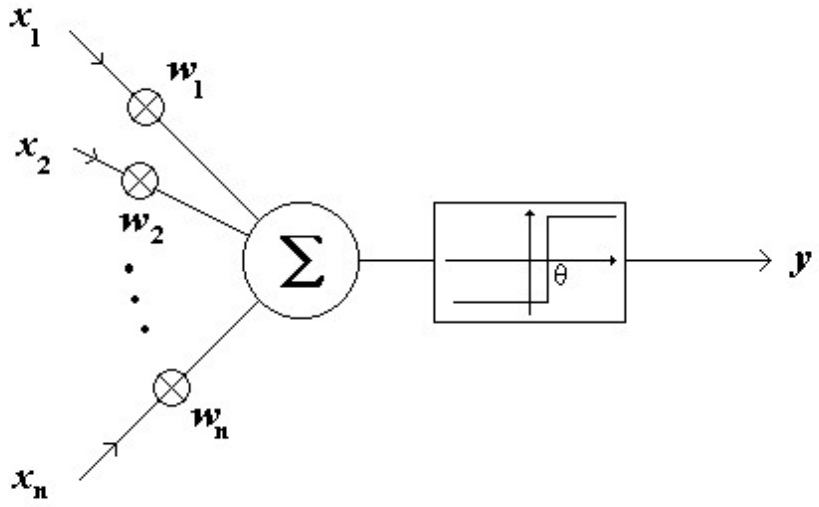
# Esplicitazione delle azioni: unità logiche a soglia

- L'unità logica a soglia (TLU) è un circuito elettrico, ma in pratica è pressoché sempre simulata tramite software
- **TLU diverse si differenziano** a due possibili livelli:
  1. *architettura*, ovvero il **numero** $n$ **di input**
  2. *parametri*, ovvero i **valori dei pesi** $w_1, \ldots, w_n$ **e della soglia** $\theta$
- **Ogni TLU** realizza (cioè, **calcola**) **una funzione**, la trasformazione $(x_1, \ldots, x_n) \rightarrow y$
- Poiché $y$ può assumere solo i valori 0 o 1, **ogni TLU calcola una funzione logica** sulle feature.

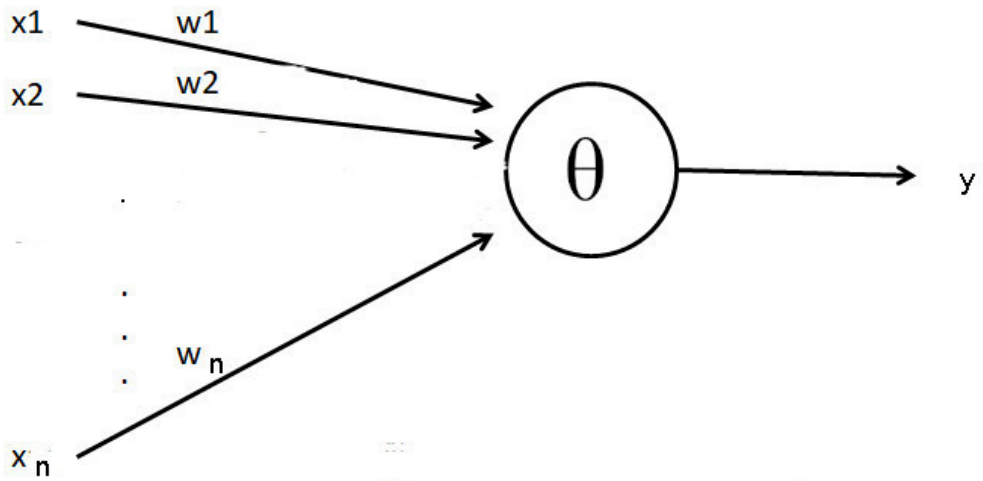


# Esplicitazione delle azioni: unità logiche a soglia

Rappresenteremo questa TLU




nel seguente modo, più compatto:

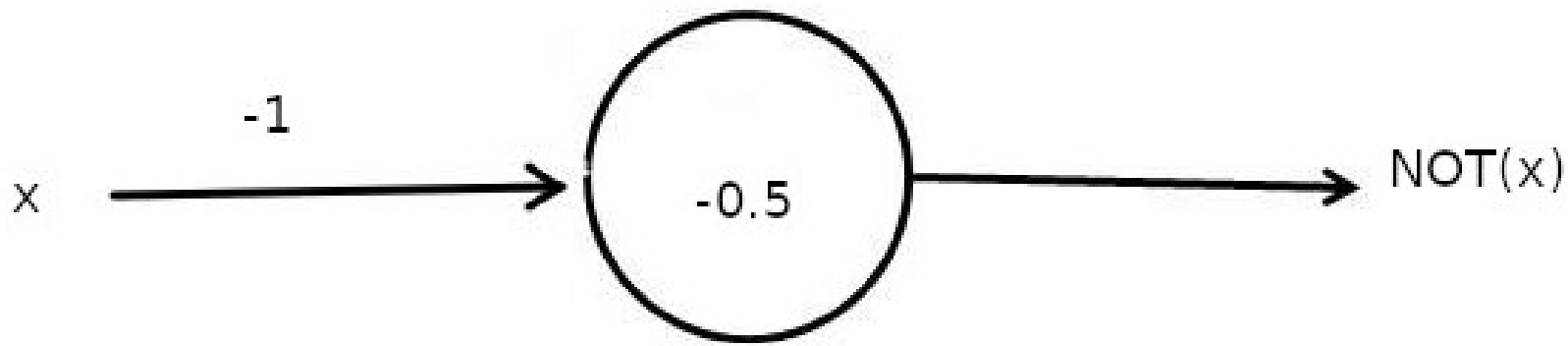


# Esplicitazione delle azioni: unità logiche a soglia

*Esempio* - Assumiamo che **la feature** $x$ **sia booleana** $(0/1)$. Ricordiamo che la tavola di verità della sua negazione logica *NOT* è la seguente:

| $x$ | $\neg x$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

Una TLU che calcoli $NOT(x)$ può essere così configurata:



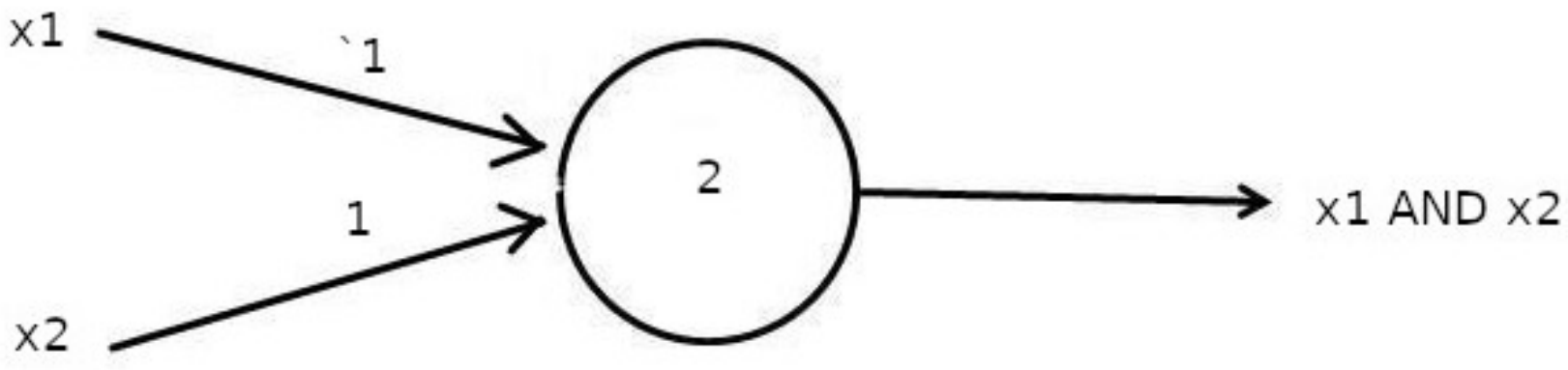

(naturalmente infinite altre configurazioni sono possibili scegliendo un peso $w < 0$ a piacere e fissando di conseguenza la soglia $\theta$ in modo che $w < \theta \leq 0$)

# Esplicitazione delle azioni: unità logiche a soglia

*Esempio* - **Le feature $x_1$ e $x_2$ siano booleane**. La tavola di verità della loro congiunzione logica *AND* è:

| $x_1$ | $x_2$ | $x_1 \wedge x_2$ |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |

Una TLU che calcoli $x_1$ *AND* $x_2$ può essere così configurata:

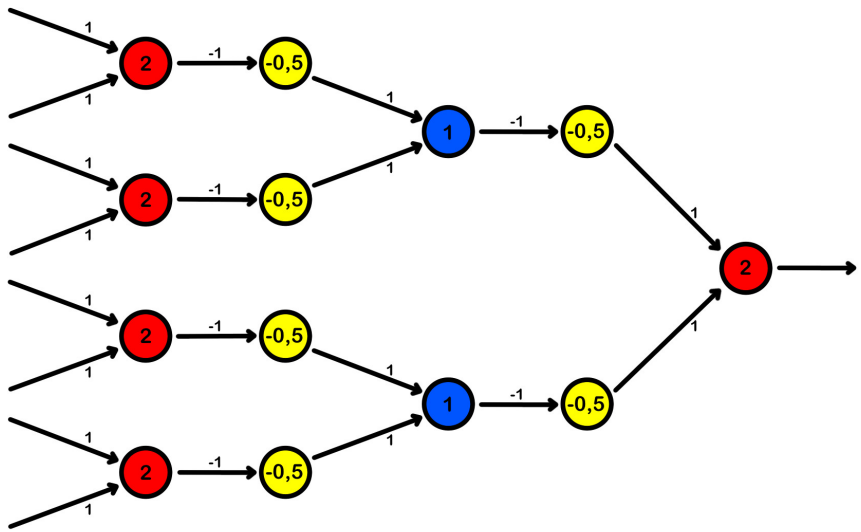



Infinite altre configurazioni sono possibili al variare di $w_1 > 0$, $w_2 > 0$ e, di conseguenza, di $\theta$ (con $\theta > w_1$, $\theta > w_2$, e $\theta \leq w_1 + w_2$). **Esercizio**: progetta una TLU che calcoli la disgiunzione logica *OR* tra $x_1$ e $x_2$.

# Esplicitazione delle azioni: unità logiche a soglia

Abbiamo dimostrato che le TLU sono in grado di calcolare *AND*, *OR* e *NOT*.

Come si è fatto per le MdT, questo costituisce informalmente la base dell'argomento per cui **componendo in cascata un numero sufficiente di TLU** è possibile realizzare qualsiasi funzione calcolata da un computer digitale.

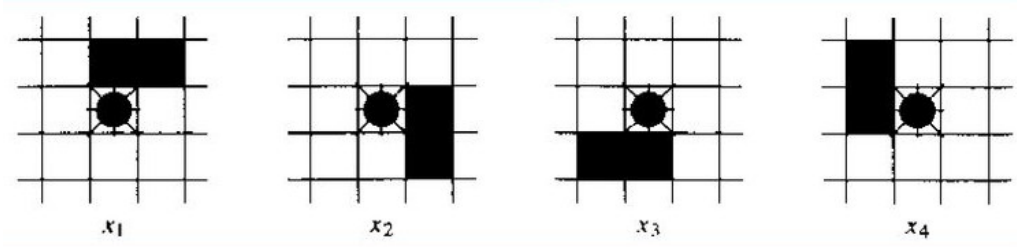



Ad esempio: • = *AND*, • = *NOT* e • = *OR*.

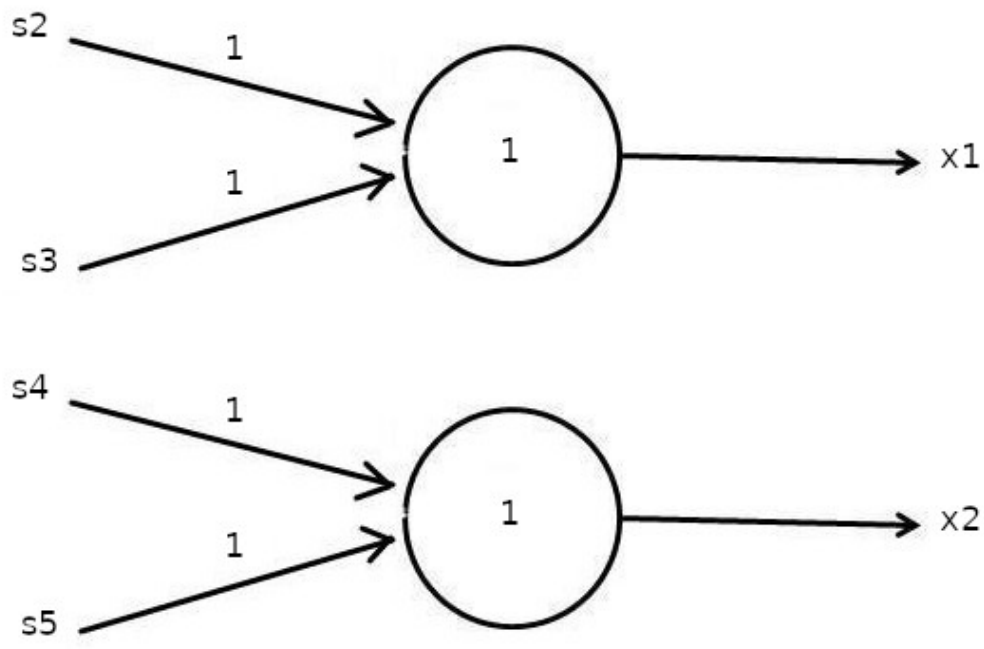
Le **reti di TLU vanno nella direzione caldeggiata dai Churchland**: macchine distribuite e parallele alternative alla MdT che si "ispirano al cervello".

# Esplicitazione delle azioni: unità logiche a soglia

Come usare **reti di TLU per il robot nel mondo a griglia**?




i.e. $x_1 = s_2 \vee s_3$, $x_2 = s_4 \vee s_5$, $x_3 = s_6 \vee s_7$ e $x_4 = s_8 \vee s_1$, ergo



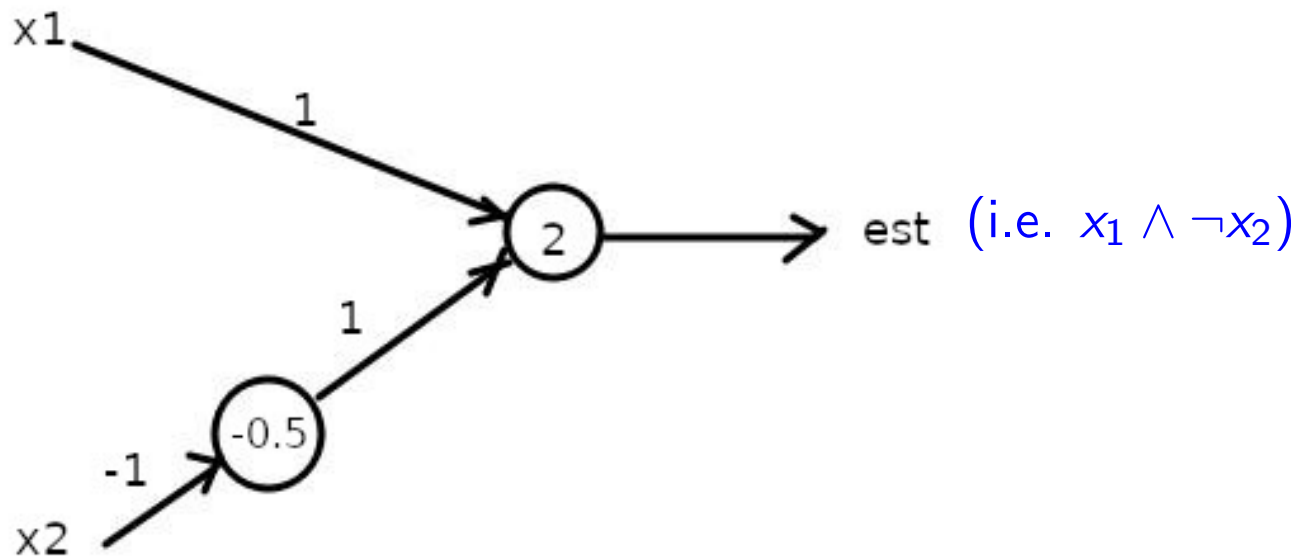

*Esercizio*: analogamente, progetta TLU che estraggano le feature $x_3$ e $x_4$.

# Esplicitazione delle azioni: unità logiche a soglia

A partire dalle TLU che estraggono le feature possiamo ora costruire **reti di TLU che verifichino le condizioni logiche corrispondenti alle 4 azioni**. Ricordando che

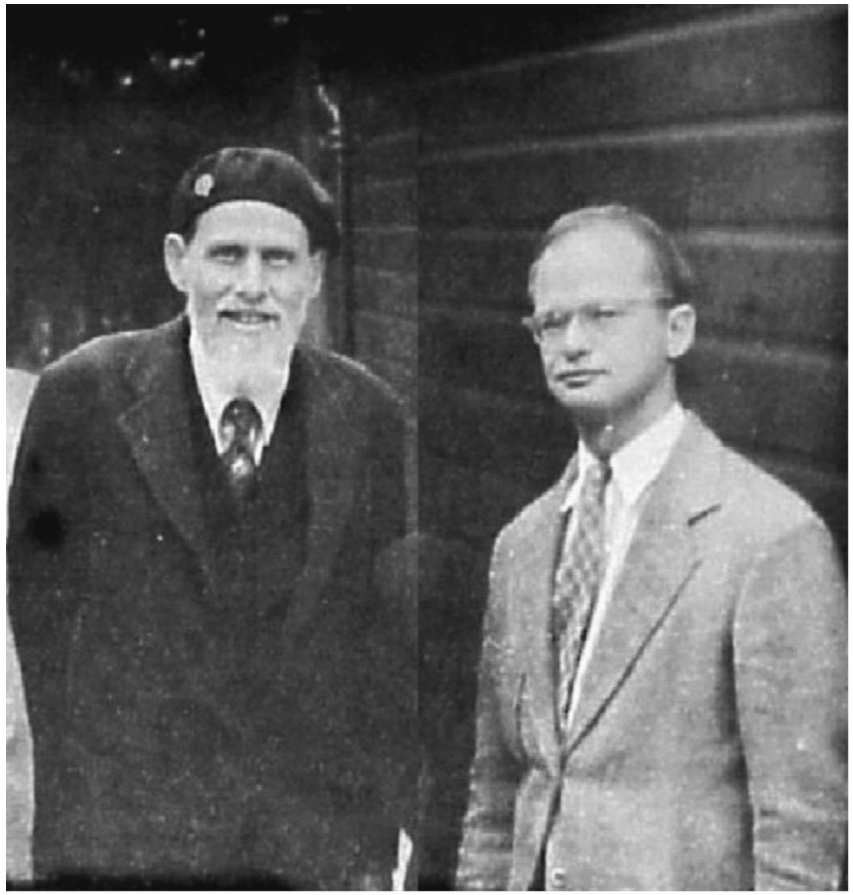
(1) se $x_1 = 1$ e $x_2 = 0$ allora **est**
(2) se $x_2 = 1$ e $x_3 = 0$ allora **sud**
(3) se $x_3 = 1$ e $x_4 = 0$ allora **ovest**
(4) se $x_4 = 1$ e $x_1 = 0$ allora **nord**

per l'azione **est** ho, ad esempio:




*Esercizio*: progetta analoghe reti per **sud**, **ovest** e **nord**.

# Il "neurone" di McCulloch e Pitts

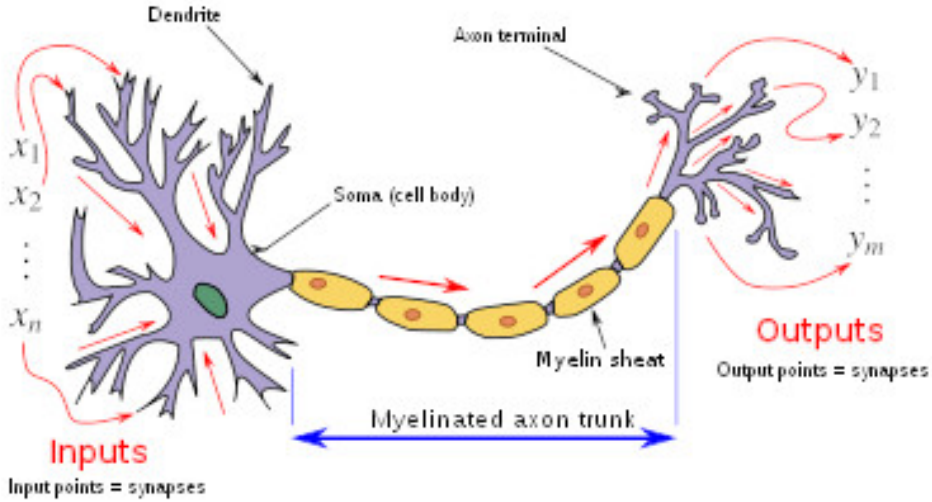


**Warren McCulloch (1898-1969) e Walter Pitts (1923-1969)**

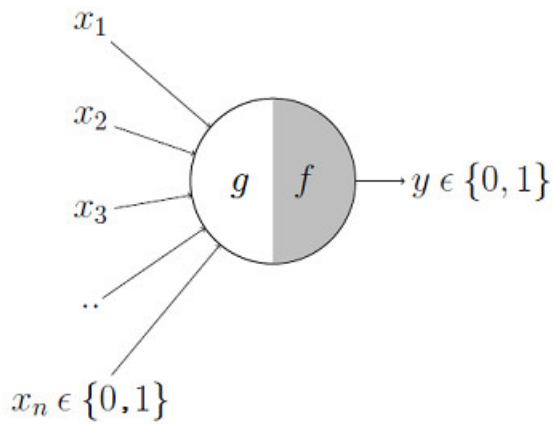
W. McCulloch e W. Pitts. "*A Logical Calculus of Ideas Immanent in Nervous Activity*", Bulletin of Mathematical Biophysics 5:115 - 133 (1943).

# Il "neurone" di McCulloch e Pitts

Ispirandosi al neurone biologico:




McCulloch e Pitts propongono un formalismo di calcolo:




che prende il loro nome e che è di fatto una TLU ante litteram.

**Esercizio**: guardare il film-capolavoro *A Clockwork Orange* di Stanley Kubrick (UK/USA, 1971) e chiedersi cosa c'entri (associazionismo, stimoli e risposte condizionate/incondizionate, cani di Pavlov e cervelli, a.k.a. *gulliver*).