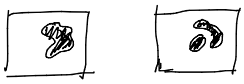
Problema: Classificare immagini in una griglia $n \times n$ bit
- ↳ CONNESSE
- ↳ NON CONNESSE



Minsky & Papert: $\psi_{conn}(X) = \begin{cases} 1 & \text{se } X \text{ è connesso} \\ -1 & \text{se } X \text{ non è connesso} \end{cases}$ $\quad X \subset \mathbb{R}^2$

↓

Formalismo dei predicati logici

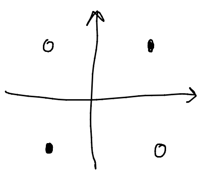
ψ_{CONN} → NON è locale

→ NON è di ordine finito ⇒ NON RISOLUBILE DA UN PERCEPTRON

Prototipo di problema non risolubile: XOR

$\vec{v} \in \mathbb{R}^D \qquad XOR(\vec{v}) = \prod_{i=1}^{D} v_i$

$D = 2$



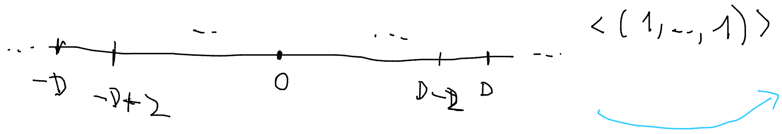
Chiaramente non ris. per $D=2$ ⇒ non ris. $\forall D$

Abbiamo visto però che aggiungendo uno strato hidden con 2 neuroni si riesce a risolvere il problema in $D=2$

? Quanti neuroni nello strato hidden servono per risolvere lo XOR in D dimensioni?

R: D neuroni.

Idea: Proiettiamo gli esempi sul sottospazio generato da $(1, \dots, 1)$



Dobbiamo suddividere in D parti questa retta → usiamo D piani

Tornando al problema delle aree connesse...

Mostriamo che il problema delle aree connesse non è risolubile da un perceptron:

Per ...

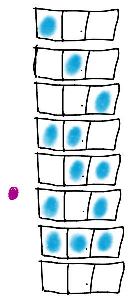
risolubile da un perceptron:

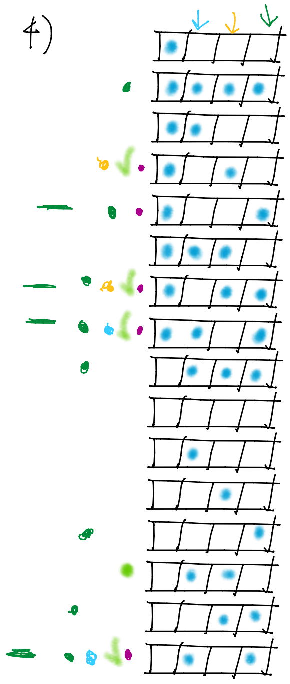
Per semplificare consideriamo aree piane formate da una sola riga di bit (vettori)

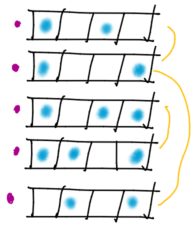
1) 

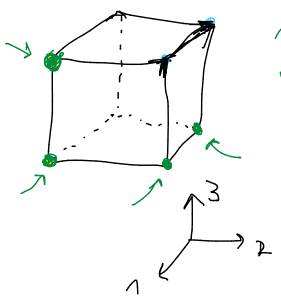
2) 

} non ci sono aree sconnesse

3) 

4) 





Sembra che i vertici in verde NON possono essere separati dagli altri vertici del cubo con un piano.

Implementazione di un NN con 1 strato di neuroni hidden:

- generare esempi giusti e sbagliati
  - posiz. casuale nella matrice
  - # casuale di passi compreso tra 1 e 'nstep'
  - mov. casuale ↑, ↓, →, ←

→ • sovrascrivo l'immagine connessa con un'altra immagine connessa con l'attenzione che non si sovrappongano (search)

⊙ scegliere il n° di neuroni hidden