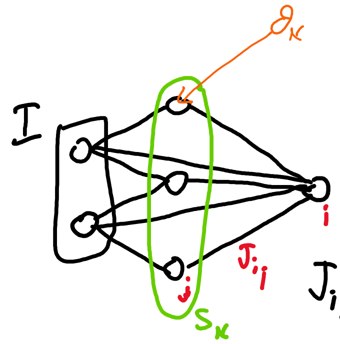
Perceptron

Monday, 17 March 2025 12:28



FITNESS FUNCTION

$$\boxed{f(\Omega) = \frac{1}{4N} \sum_K^{\sim} \left| O^T - O_K \right|^2 \cdot \text{cost}}$$

distance between theoretical output and observed

Use the DENSITY OF CONNECTIONS

because you want to find the best performing network with the least cost, i.e. # of links (and neurons)

$J_{ij} \in [-1,+1]$

$S_K \in [2, 5]$

$\in \mathbb{N}$

indicative

$J_{ij} \to J_{ij} + \xi$

mutation

Ω: space of possible networks

$$\Omega = \{ \underbrace{\{J_{ij}\}}_{\text{links}}, \underbrace{\{\vartheta_K\}}_{\text{thresholds}}, \underbrace{\{S_K\}}_{\text{\# of intra neurons}} \}$$

Objective: implement the XOR gate

A	B	O^T
0	0	0
0	1	1
1	0	1
1	1	0

I want my network to reproduce as best as possible this logic gate

How should I organize the problem?

So the expected result is the perceptron, where the activation function is a Heaviside, i.e. the threshold $\vartheta_K = 0 \ \forall K$.

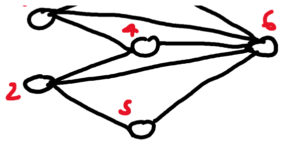
The idea is that I should have:

- a connectivity matrix J_{ij}, where $\forall i,j : J_{ij} = w_{ij}$ so each element is the weight of the (DIRECTED) link node $j \to$ node i.

If $J_{ij} = 0 \to$ no link

I think this is wrong: it should be $J_{ij} \in \{-1,$ and this represent only the presence or absence a link between two neurons; while the weights are a different variable, which trained through backpropag

For this matrix, I should number the neurons from left to right and from up to down, i.e.



Therefore, the dimension of J_{ij} depends on S_K, i.e. the # of interneurons, in particular, J_{ij} is a $N \times N$ matrix, where $N = \underbrace{2}_{\text{input neur.}} + \underbrace{S_K}_{\text{hidden neur.}} + \underbrace{1}_{\text{output neurons}}$.

- Here the input is binary: 0 or 1, and also the output.
- The threshold range should be from the maximum possible value of $w \cdot x$ to the minimum:

$$\theta_K \in [-\min(w), \max(w)]$$

Is it $\theta_K \in [-1,+1]$?

- Oh maybe $J_{ij} \in [-1,+1]$ and the perceptron is the best one be θ_K it's in the middle of the range.

  But so it would be the same if $J_{ij} \in [0,2]$ and $\theta_K = +1$ for the perceptron?

- I think I can go without bias here.
- cost: $\dfrac{\sum_{ij} g(J_{ij})}{2}$ where $g(x) = \begin{cases} 0 & \text{if } x=0 \\ 1 & \text{if } x \neq 0 \end{cases}$

SHOULD IT BE SQUARED?

DOMANDE:

1) Perché $J_{ij} \in [-1,+1]$?
2) un threshold ∀ network? o ∀ neurone?
3) Is the cost function right like this?
4) Should I use back propagation?

or is it

↳ or maybe it's $J_{ij} \in \{-1, +1\}$ where -1 : no links, $+1$: link

+1 : link for
-1 : // t
0 : no link
(i.e. $J_{ij} = \{-1,$

and then the weights are not computed through the EA, but by backpropagation?

5) Is N the total number of possible networks?
i.e. $|\Omega| = N$?

SKETCH OF THE ALGORITHM

So the idea being:

1) I create all the possible networks,
2) Select solutions
3) Train and backprop.
4) Calculate fitness value, i.e. loss for every solution
5) Selection
6) Mutation + random generation
7) Back to step 3.