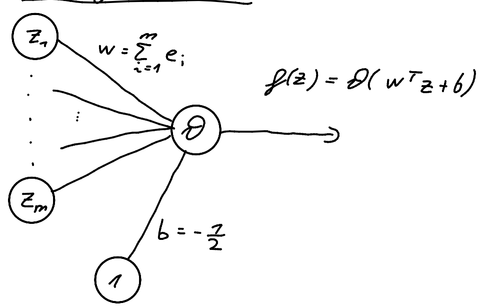
Exercise 2

07 May 2021 23:48

1 Classification Capacity

1.1 Simple Networks

1.) logical OR:



we def.: $w := \sum_{i=1}^m e_i$, $b := -\frac{1}{2}$, $\vartheta(\tilde z) = \begin{cases} 1 & \tilde z > 0 \\ 0 & \tilde z \leq 0 \end{cases}$

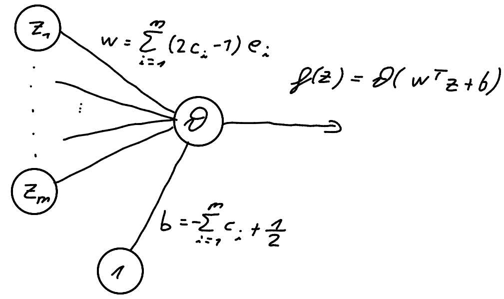
Then:

$$f(z) = \vartheta(w^T z + b)$$

$$= \vartheta(\underbrace{\sum_{i=1}^m z_i - \frac{1}{2}}) = \begin{cases} 0 & \text{if } \forall i \in [\![1, m]\!]: z_i = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\begin{cases} = -\frac{1}{2} & \text{if } \forall i \in [\![1, m]\!]: z_i = 0 \\ \geq 1 - \frac{1}{2} = \frac{1}{2} & \text{otherwise} \end{cases}$$

2.) logical AND:



we def.: $w := \sum_{i=1}^m (2c_i - 1) e_i$, $b := -\sum_{i=1}^m c_i + \frac{1}{2}$, $\vartheta(\tilde z) = \begin{cases} 1 & \tilde z > 0 \\ 0 & \tilde z \leq 0 \end{cases}$

Then:

$$f(z) = \vartheta(w^T z + b)$$

$$= \vartheta\left(\sum_{i=1}^m (2c_i z_i - z_i - c_i) + \frac{1}{2}\right)$$

$$\Rightarrow f(z) = \vartheta\left(\underbrace{\sum_{\substack{i=1 \\ c_i = 0}}^m \overbrace{-z_i}^{\leq 0}} + \underbrace{\sum_{\substack{i=1 \\ c_i = 1}}^m \overbrace{z_i - 1}^{\leq 0}} + \frac{1}{2}\right)$$

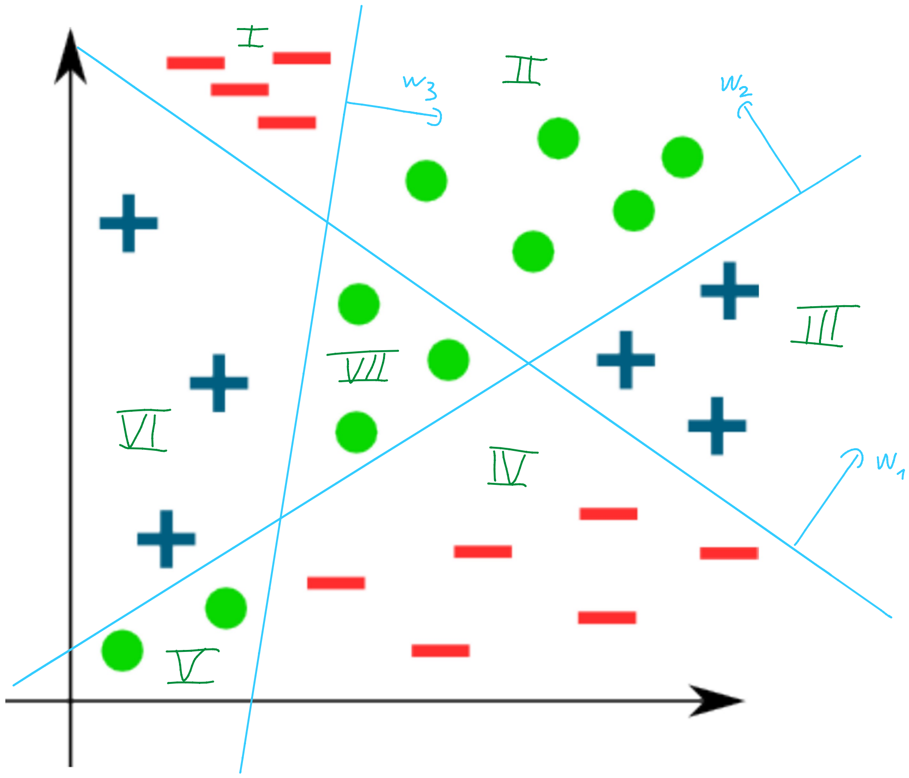
$\{ = 0$ if ...

$$\left(\underbrace{\sum_{\substack{i=1 \\ c_i=0}} -z_i}_{\begin{cases} = 0 & \text{if } z_i = 0 = c_i \quad \forall i \in [\![1,m]\!], c_i = 0 \\ \leq -1 & \text{otherwise} \end{cases}} + \underbrace{\sum_{\substack{i=1 \\ c_i=1}} z_i - 1}_{\begin{cases} = 0 & \text{if } z_i = 1 = c_i \quad \forall i \in [\![1,m]\!], c_i = 1 \\ \leq -1 & \text{otherwise} \end{cases}} + \frac{1}{2} \right)$$

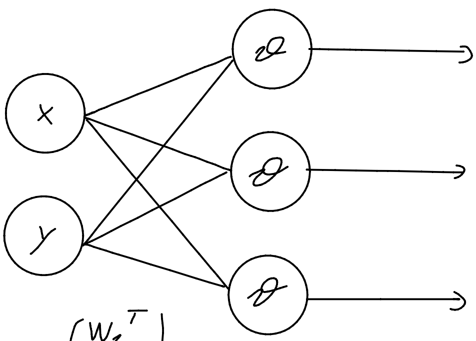
$$\Rightarrow \sum_{\substack{i=1 \\ c_i=0}}^{m} -z_i + \sum_{\substack{i=1 \\ c_i=1}}^{m} z_i - 1 + \frac{1}{2} \begin{cases} \overbrace{= \frac{1}{2}}^{>0} & \text{if: } z_i = c_i \quad \forall i \in [\![1,m]\!] \\ \underbrace{\leq -\frac{1}{2}}_{\leq 0} & \text{otherwise} \end{cases}$$

$$\Rightarrow f(z) = \begin{cases} 1 & \text{if } z = c \\ 0 & \text{otherwise} \end{cases}$$

(3.) m-dim cube:



Let $(w_i)_{i \in [\![1,3]\!]} \subseteq \mathbb{R}^2$ be according to the figure above.

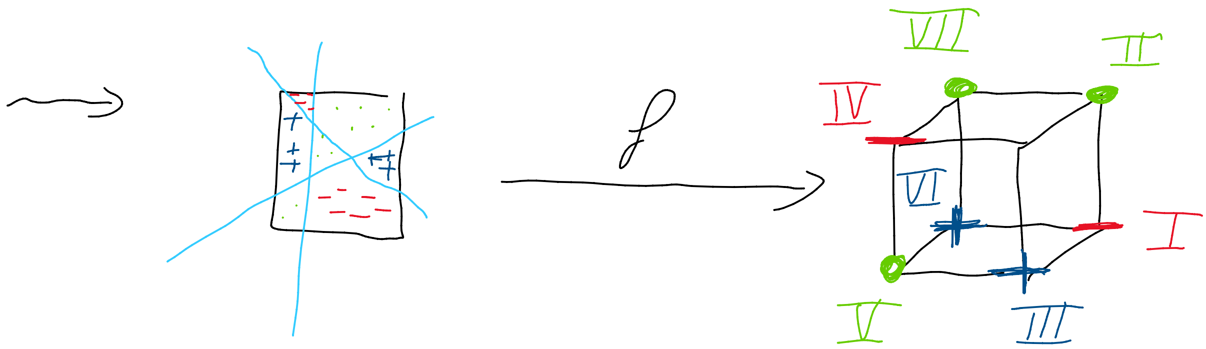


$$\begin{pmatrix} W_1^T \\ W_2^T \\ W_3^T \end{pmatrix}$$

$$f\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) = \vartheta\left(\begin{pmatrix} W_1^T \\ W_2^T \\ W_3^T \end{pmatrix} x\right)$$

The regions are characterised by

	$w_1^T\binom{x}{y}$	$w_2^T\binom{x}{y}$	$w_3^T\binom{x}{y}$	$f\left(\binom{x}{y}\right)$
I	>0	>0	<0	$(1,1,0)^T$
II	>0	>0	>0	$(1,1,1)^T$
III	>0	<0	>0	$(1,0,0)^T$
IV	<0	<0	>0	$(0,0,1)^T$
V	<0	<0	<0	$(0,0,0)^T$
VI	<0	>0	<0	$(0,1,0)^T$
VII	<0	>0	>0	$(0,1,1)^T$



In a more general setting we have

$((x_i, y_i))_{i\in[\![1,N]\!]} \subseteq \mathbb{R}^d \times [\![1, C]\!]$, where $d \geq 1$, $C>0$.

Then we add linear decision boundaries until the regions defined by these m boundaries contain instances from at most one class.

Remark: by placing the decision rule between two points it can be ensured that $m<\infty$.

The NN defined by

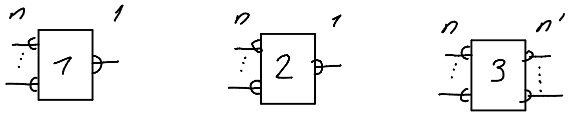
$$x \longmapsto f(x) = \Theta\left(\begin{pmatrix} w_1^T \\ \vdots \\ w_m^T \end{pmatrix} x\right)$$

elementwise application of Heaviside function

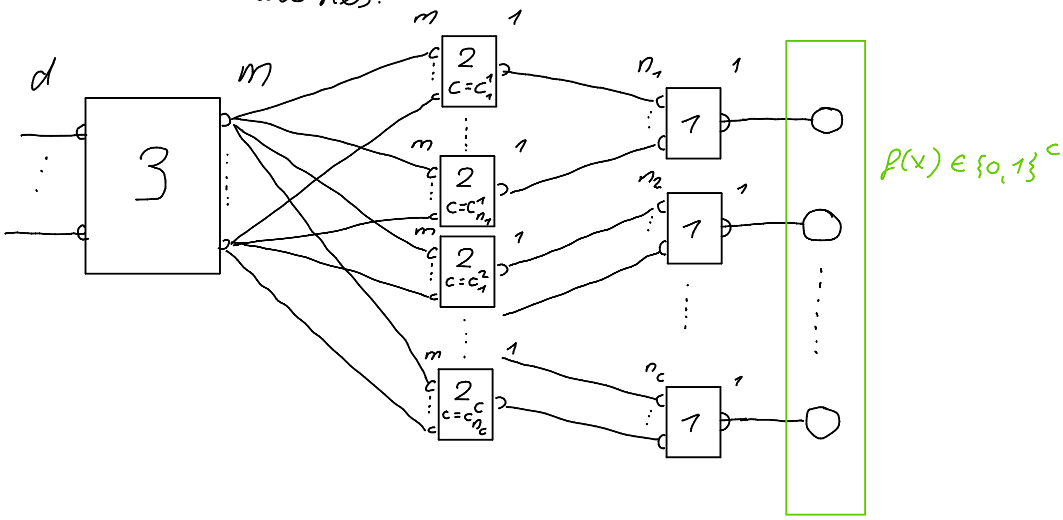
will map each region to one unique corner of an m-dimensional hypercube. Because there is no more than one class in each region, there is at most one class in each corner.

1.2:

We denote the networks from 1.1 with



Let $c^i_1,\dots,c^i_{n_i} \in \mathbb{R}^m, i \in [[1,c]]$ be the vectors corresponding to the corners of the i-th class, where $m < \infty$ is the number of decision boundaries.



The first part (network 3) distributes each instance to one corner which corresponds to a unique class. The second part (networks 2) convert the spatial information (corner of instance) to a one hot code (every component of the output vector corresponds to one corner). The third part (networks 1) perform a logical OR on all corners corresponding to certain class (i-th component of the output is one, if the input for any of the n_i corners of class i is one). The output therefore is a one-hot-code for the class of the input instance.

2 Linear Activation Function

Let $L > 0$ and $(\phi_\ell)_{\ell \in [[1,L]]}$ linear maps.

for $\ell \in [[1,L]]$: $x \longmapsto x \cdot B_\ell + b_\ell =: \tilde{\phi}_\ell(x)$ is obviously linear.

Therefore, as a composition of a finite number of linear functions, the NN

$$f(x) = \phi_L \circ \tilde{\phi}_L \circ \phi_{L-1} \circ \tilde{\phi}_{L-1} \cdots \phi_1 \circ \tilde{\phi}_1$$

is linear. We define $\Phi := \phi_L \circ \tilde{\phi}_L \circ \ldots \phi_1 \circ \tilde{\phi}_1$

Then:

$$f(x) = \Phi(x) = \Phi(\mathbb{1} \cdot x + 0)$$

But $\Phi(\mathbb{1}x+0)$ describes a NN with one layer and bias vector $b=0$ and weights $W = \mathbb{1}$.

$\blacksquare$