

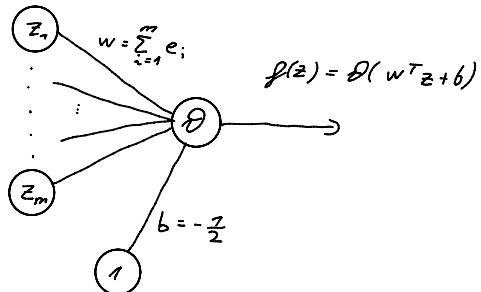
Exercise 2

07 May 2021 23:48

1 Classification Capacity

1.1 Simple Networks

① logical OR:

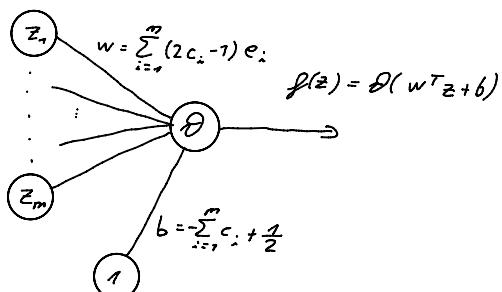


$$\text{we def: } w := \sum_{i=1}^m e_i, \quad b := -\frac{1}{2}, \quad \theta(\tilde{z}) = \begin{cases} 1 & \tilde{z} > 0 \\ 0 & \tilde{z} \leq 0 \end{cases}$$

then:

$$\begin{aligned} f(z) &= \theta(w^T z + b) \\ &= \theta\left(\sum_{i=1}^m z_i - \frac{1}{2}\right) = \begin{cases} 0 & \text{if } \forall i \in [1, m]: z_i = 0 \\ 1 & \text{otherwise} \end{cases} \\ &\begin{cases} = -\frac{1}{2} & \text{if } \forall i \in [1, m]: z_i = 0 \\ \geq 1 - \frac{1}{2} = \frac{1}{2} & \text{otherwise} \end{cases} \end{aligned}$$

② logical AND:



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

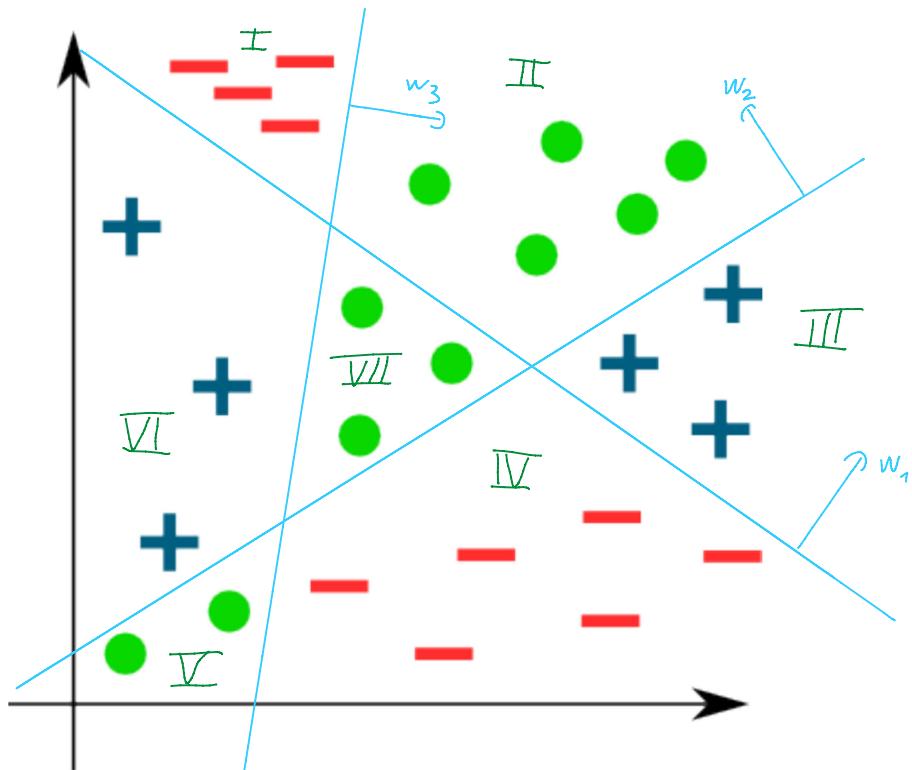
then:

$$\begin{aligned} f(z) &= \theta(w^T z + b) \\ &= \theta\left(\sum_{i=1}^m (2c_i z_i - z_i - c_i) + \frac{1}{2}\right) \end{aligned}$$

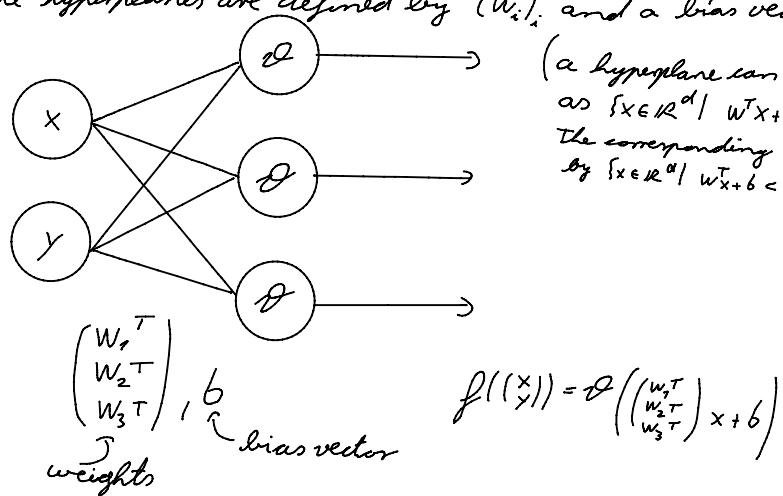
$$\Rightarrow f(z) = \theta\left(\underbrace{\sum_{\substack{i=1 \\ c_i=0}}^m -z_i}_{l=0} + \underbrace{\sum_{\substack{i=1 \\ c_i=1}}^m z_i - 1}_{l=0} + \frac{1}{2}\right)$$

$$\begin{aligned}
 & \left(\sum_{\substack{i=1 \\ c_i=0}}^n -z_i + \sum_{\substack{i=1 \\ c_i=1}} z_i - 1 + \frac{c}{2} \right) \\
 & \quad \left\{ \begin{array}{ll} = 0 & \text{if } z_i = 1 = c_i \\ \leq -1 & \text{otherwise} \end{array} \right. \quad \forall i \in [1, m], c_i = 1 \\
 & \left\{ \begin{array}{ll} = 0 & \text{if } z_i = 0 = c_i \\ \leq -1 & \text{otherwise} \end{array} \right. \quad \forall i \in [1, m], c_i = 0 \\
 & \Rightarrow \sum_{\substack{i=1 \\ c_i=0}}^n -z_i + \sum_{\substack{i=1 \\ c_i=1}} z_i - 1 + \frac{c}{2} \quad \left\{ \begin{array}{ll} \geq 0 & \text{if } z_i = c_i \quad \forall i \in [1, m] \\ \leq -\frac{1}{2} & \text{otherwise} \\ \leq 0 & \end{array} \right. \\
 & \Rightarrow f(z) = \begin{cases} 1 & \text{if } z=c \\ 0 & \text{otherwise} \end{cases}
 \end{aligned}$$

(3) m -dim cube:



Let $(w_i)_{i \in [1, 3]} \subseteq \mathbb{R}^2$ be according to the figure above. The hyperplanes are defined by (w_i) , and a bias vector $b \in \mathbb{R}^3$



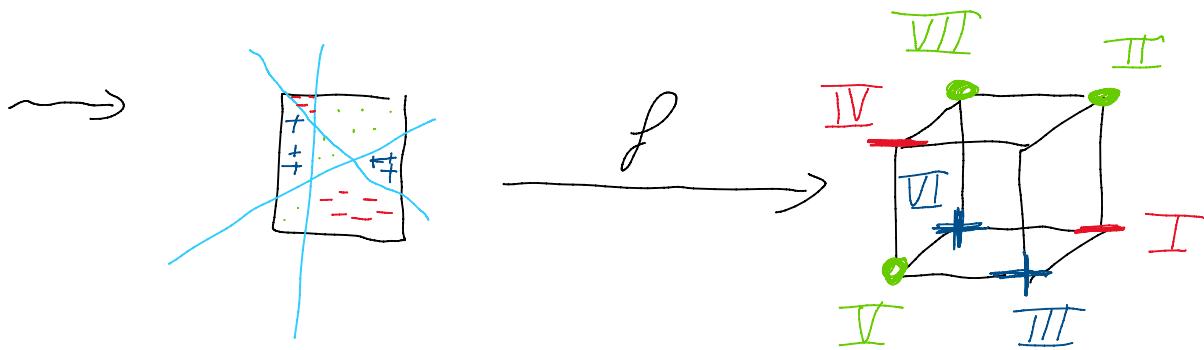
(a hyperplane can in general be written as $\{x \in \mathbb{R}^d \mid w^T x + b = 0\}$ for some $w \in \mathbb{R}^d \setminus \{0\}, b \in \mathbb{R}$. The corresponding half spaces are given by $\{x \in \mathbb{R}^d \mid w^T x + b < 0\}$, $\{x \in \mathbb{R}^d \mid w^T x + b > 0\}$)

The regions are shown at x and y

weights *was vector*

The regions are characterised by

	$w_1^T(x) + b_1$	$w_2^T(x) + b_2$	$w_3^T(x) + b_3$	$f((x))$
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

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

Then we add linear decision boundaries until the regions defined by these 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

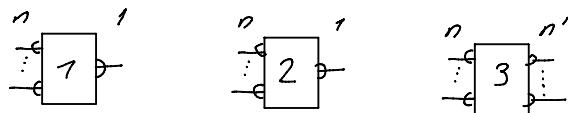
$$x \mapsto f(x) = \underset{P}{\partial} \left(\begin{pmatrix} w_1^T \\ \vdots \\ w_m^T \end{pmatrix} x + b \right)$$

elementwise application
of Heaviside function

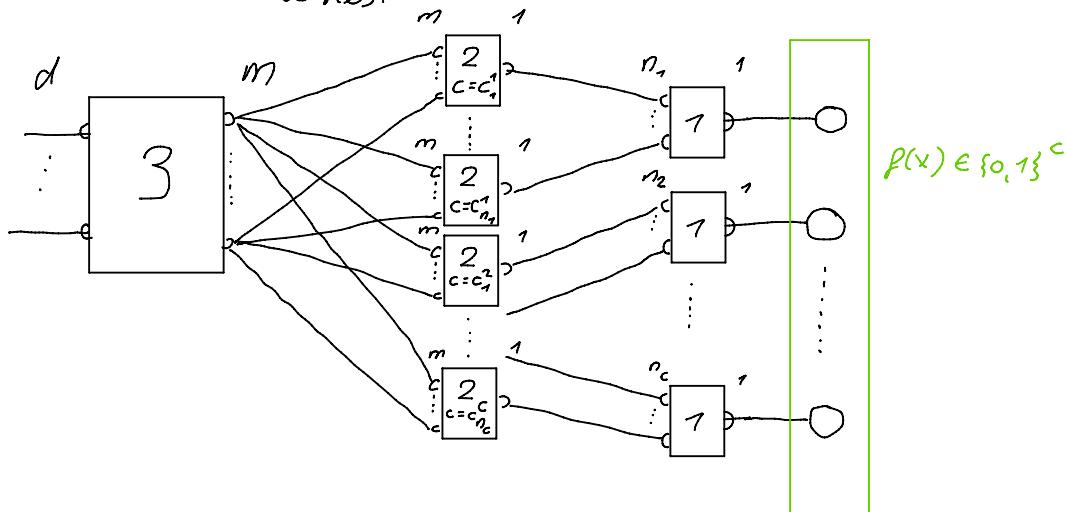
will map each region to one unique corner of an m -dimensional hypercube. Because there is no more than one class in each region, there is at most one class in each corner.

7.2:

We denote the networks from 7.1 with



Let $c_{1,1}^i, \dots, c_{n_i}^i \in \mathbb{R}^m$, $i \in [1, c]$ be the vectors corresponding to the corners of the i -th class, where $n < \infty$ 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 (network 2) converts the spatial information (corner of instance) to a one-hot code (every component of the output vector corresponds to one corner).

The third part (network 1) performs 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 $(\phi_e)_{e \in [1, L]}$ linear maps.

for $e \in [1, L]$: $x \mapsto x \cdot \beta_e + b_e =: \tilde{\phi}_e(x)$
is obviously linear.

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

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

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

$$f(x) = \bar{\Phi}(x) = \bar{\Phi}(\Pi x + 0)$$

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

