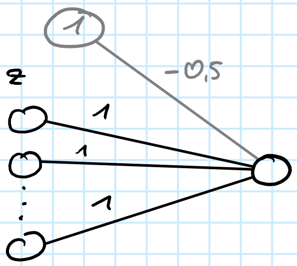
Task 1, 2 - Own feedback

Lars Kühmichel
Nicolas Wolf

Donnerstag, 6. Mai 2021 14:56

1.1

a) $z \in \{0,1\}^n \mapsto f(z) = \begin{cases} 0 & \text{if } z_i = 0 \ \forall i \\ 1 & \text{otherwise} \end{cases}$



$$\Rightarrow \tilde{z} = (1, z) \cdot \begin{pmatrix} -0.5 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \underbrace{\sum_i z_i}_{=0 \text{ iff } z=0} - 0.5$$

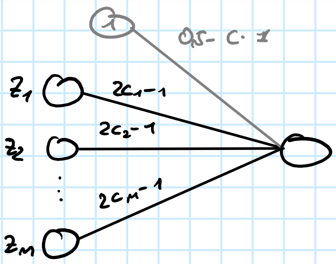
$$f(z) = \Theta(\tilde{z}) \rightarrow \begin{cases} 0 & \text{if } \tilde{z} \le 0 \\ 1 & \text{if } \tilde{z} > 0 \end{cases}$$

Comment: We added a bias of -0.5, so that we don't have to worry about $\tilde{z} = 0$. Otherwise equal.

b) let $c \in \{0,1\}^m$ be fixed, $z \in \{0,1\}^m \mapsto f(z) = \begin{cases} 1 & z = c \\ 0 & \text{otherwise} \end{cases}$

$\beta_i = \begin{cases} 1 & \text{if } c_i = 1 \\ -1 & \text{if } c_i = 0 \end{cases} = 2c - \mathbb{1}$ $\quad b = 0.5 - c.\text{sum}()$

activation function: $\varphi = \Theta$



$$\tilde{z} = z \cdot \beta + b = z \cdot (2c - 1) + b$$

$$= 2\, z \cdot c - z \cdot \mathbb{1} - c \cdot \mathbb{1} + 0.5$$

$$= 2 \cdot z \cdot c - (z + c) \cdot \mathbb{1} + 0.5$$

- if $z = c$: $z \cdot c = c \cdot c = \sum c_i^2 = \sum c_i \Rightarrow \tilde{z} = 0.5$
- if $z \neq c$: $\exists\, i: z_i \cdot c_i = 0$

$$\Rightarrow z \cdot c = \sum_i z_i c_i < \sum_i c_i^2 = \sum_i c_i = c \cdot \mathbb{1}$$

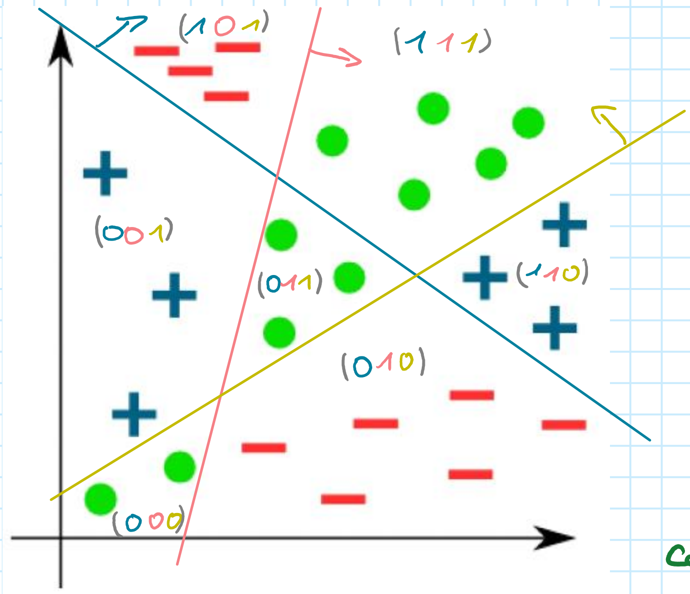
and $\sum_i z_i \cdot c_i < \sum_i z_i^2 = \sum_i z_i = z \cdot \mathbb{1}$

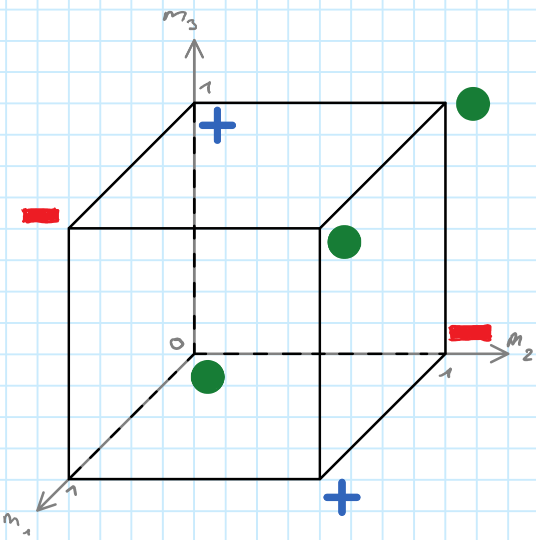
$$\Rightarrow 2 \cdot z \cdot c < (z + c) \cdot \mathbb{1}$$

$$\Rightarrow \tilde{z} < 0$$

Comment: We added 0.5 to the bias as above. Otherwise equal

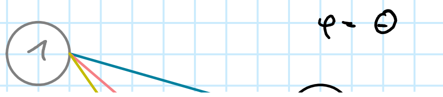
c)



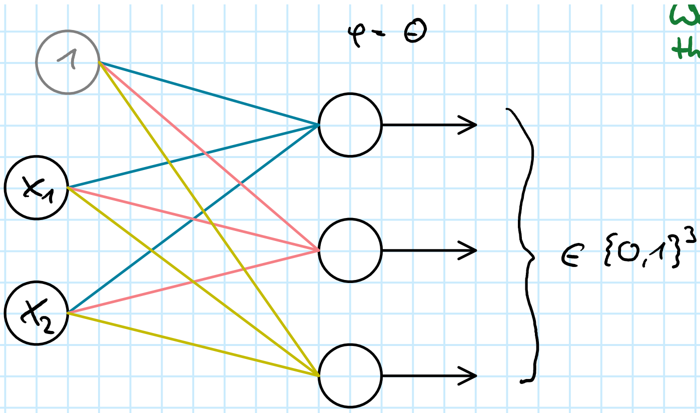


Comment: Slightly different decision boundaries but essentially the same.

We forgot to describe, how to generalize this to other training sets.

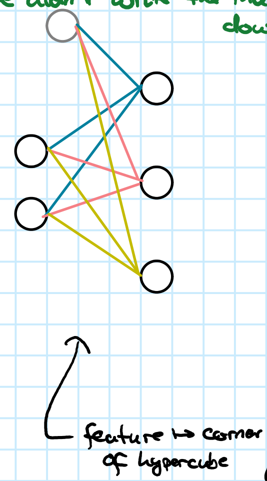


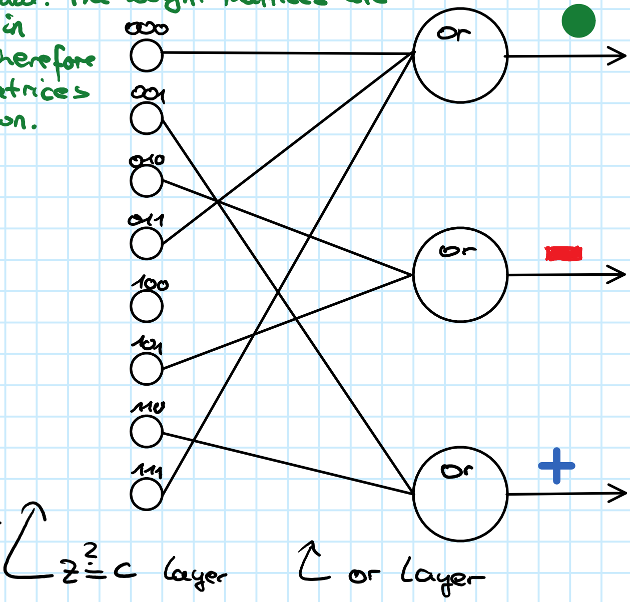
We forgot to describe, how to generalize this to other training sets.



1.3 • 000, 011, 111 ▬ 010, 101 ✚ 001, 110

Comment: Essentially equal. The weight matrices are described above or in the text below. Therefore we didn't write the matrices down.





- First, we map the features onto the corners of $\{0,1\}^m$ (for the given dataset: $m=3$). Since each corner contains maximal 1 class, we can use the network from 1.1.2 to test, if the output equals $c \in \{0,1\}^m$, which we do for every combination. Only one of those 2^m nodes will output a 1. Since we know the class labels of the hypercube corners, we combine the outputs using the or-network, s.t. one or-node is linked to all $z \overset{?}{=} c$-nodes with the same class label (the weights to the other $z \overset{?}{=} c$-nodes are set to 0).

- For this example, we only need a 3-dimensional hypercube, but for other data sets m could be larger which may lead to overfitting

② • Lemma: Let $f: \mathbb{R}^l \to \mathbb{R}^m$, $g: \mathbb{R}^n \to \mathbb{R}^l$ linear. Then $f \circ g: \mathbb{R}^n \to \mathbb{R}^m$ is linear

Prove: Let $v, w \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$

g linear f linear

$$(f \circ g)(\lambda v + w) = f(g(\lambda v + w)) \overset{\downarrow}{=} f(\lambda g(v) + g(w)) \overset{\downarrow}{=} \lambda f(g(v)) + f(g(w))$$

- If φ_ℓ is linear, then $z_{\ell-1} \mapsto z_\ell$ is linear

$$z_\ell = \varphi_\ell(\tilde{z}_\ell) = \varphi_\ell\Big(\underbrace{(1, z_{\ell-1}) \cdot \begin{pmatrix} b_\ell \\ B_\ell \end{pmatrix}}_{\text{linear}}\Big)$$

Comment: The sample solution uses linear functions, but the proof can be generalized to linear mappings, as we did. Our proof follows mainly the same idea but the sample solution's proof is better since it is a constructive proof as opposed to ours.

- Induction: $z_1(z_0)$ is linear
  - assume $z_\ell(z_0)$ is linear
  - $z_{\ell+1} = \underset{\text{linear}}{\varphi_{\ell+1}}\Big(\underbrace{(1, \underbrace{z_\ell(z_0)}_{\text{linear}}) \cdot \begin{pmatrix} b_{\ell+1} \\ B_{\ell+1} \end{pmatrix}}_{\text{linear}}\Big) \Rightarrow z_{\ell+1}(z_0)$ linear

- $\Rightarrow Y = z_L(X)$ is linear

  $\Rightarrow$ can find transformation matrix B, s.t. $Y = X \cdot B$

  $\Rightarrow$ use B as weights for 1-layer network