Exercise 2

1 Hand-crafted Network

1.) logical OR: $z \to f(z) = \varphi(z \cdot \beta + b) = \Theta(z \cdot \vec{1} - 0{,}5)$

$\hookrightarrow \varphi = \Theta$: Heaviside step function

$\hookrightarrow \beta = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} =: \vec{1}$

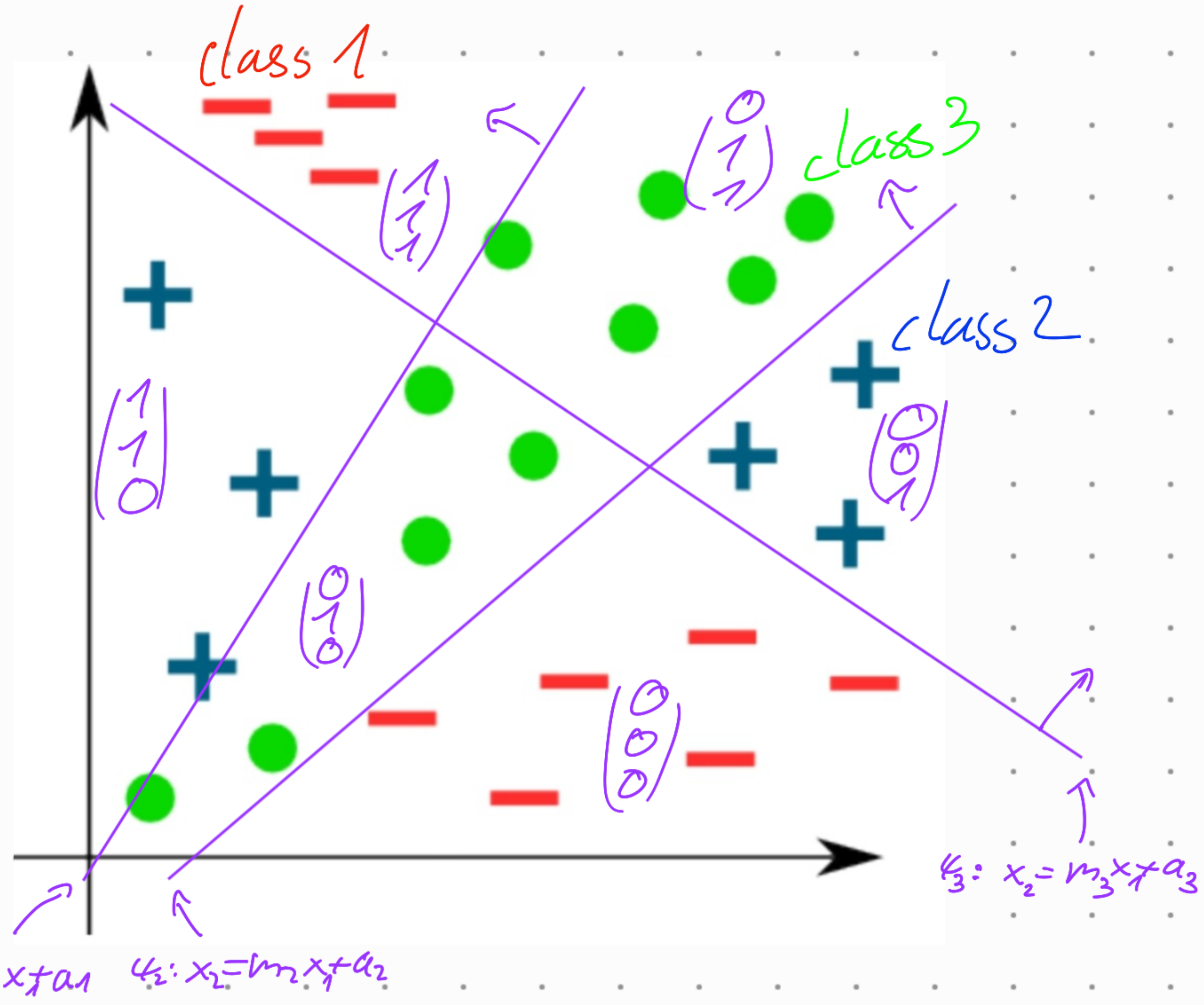
$\hookrightarrow b = -0{,}5$

2.) masked logical OR: $z \to g(z;c) = \Theta(z \cdot c - 0{,}5)$

$\hookrightarrow \beta = c$

3.) perfect match: $z \to h(z;c) = \Theta((2c - \vec{1}) \cdot z + b_p)$

$$b_p = -\sum_i c_i + 0{,}5$$



Layer 1:

First Layer checks if a datapoint lies above or below the lines ℓ_1, ℓ_2, ℓ_3
neuron k checks if x_k above or below ℓ_j

$$\rightarrow z_1^k = \Theta(x \underbrace{\begin{pmatrix}-m_k\\1\end{pmatrix}}_{:=\beta_1^k} - a_k)$$

Layer 2:

use perfect match neurons to determine which decision region the input is in:

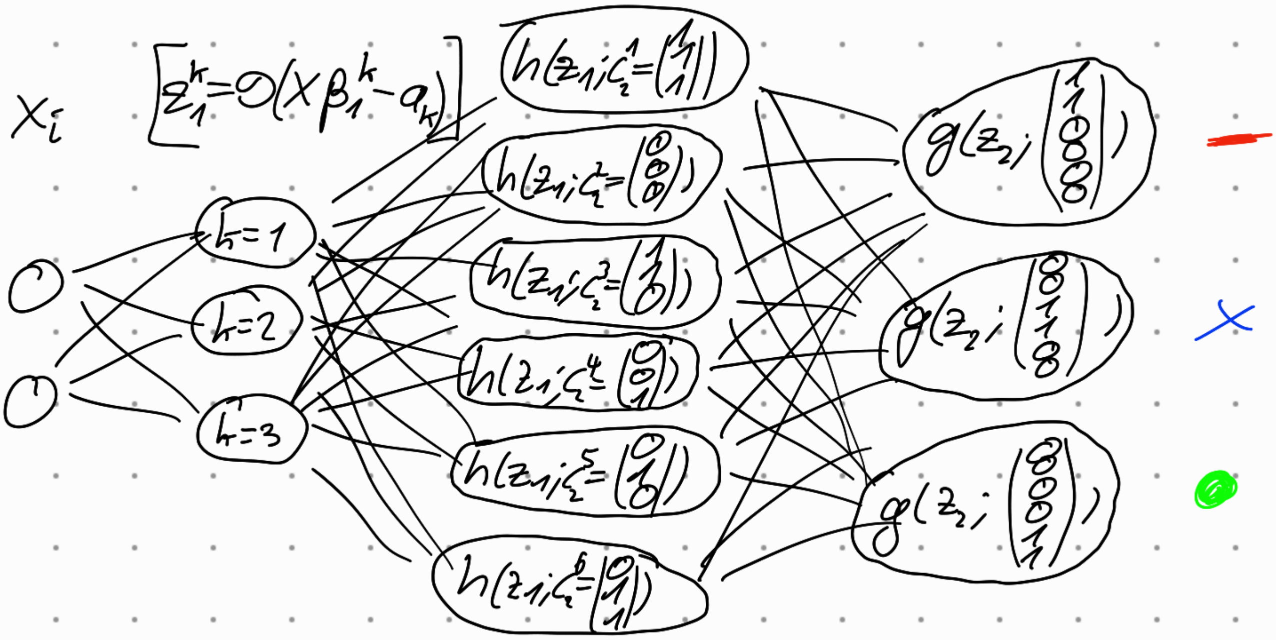
$$z_2^k = h(z_1, c_2^k)$$ for each hypercube corner c^k as indicated in the plot

Layer 3:

use masked logical OR neurons to associate the decision regions with the one-hot encoded labels

$$z_3^k = g(z_2; c_3^k);$$ c_3^k: has 1s in the coordinates associated with the hypercube corners that belong to label k

network:



Layer 0 1 2 3

For arbitrary dimensions D the decision boundaries will be hyperplanes and the number of neurons will depend on the label distributions.

A zero training loss classifier is by definition over-trained. So for noisy data, the results on the test set will be bad. Also, the complexity of the network gets quiet high for larger dimensions and number of labels.

$\boxed{2}$ Linear Activation Function

for identity activation function

$$z_0 = X \;;\; z_l = \varphi_l(z_{l-1} \cdot B_l + b_l) \overset{\downarrow}{=} z_{l-1} \cdot B_l + b_l$$

$$\Rightarrow z_1 = z_0 \cdot B_1 + b_1 = X \cdot B_1 + b_1$$

$$z_L = z_{L-1} B_L + b_L = (z_{L-2} \cdot B_{L-1} + b_{L-1}) \cdot B_L + b_L$$

$$= z_{L-2} B_{L-1} B_L + b_{L-1} B_L + b_L$$

insert $z_{L-2} = z_{L-3} \cdot B_{L-2} + b_{L-2}$ and so on

$$= X \prod_{i=1}^{L} B_i + \sum_{i=1}^{L} b_i \prod_{k=i+1}^{L} B_k$$

$$= X B^* + b^* \quad \text{with} \quad B^* = \prod_{i=1}^{L} B_i, \quad b^* = \sum_{i=1}^{L} b_i \prod_{k=i+1}^{L} B_k$$

which is a 1-layer NN $\square$