2 Classification Capacity   + Comment
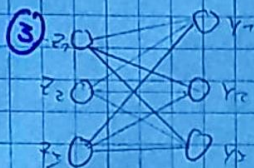
2.1 Simple Networks

(1)
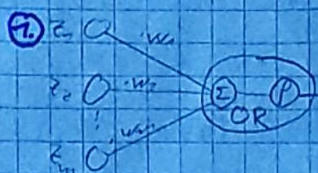
$$w_i = 1 \quad \forall i \in \{1,..,m\}, \quad P(\tilde{z}) = \begin{cases} 0 & \text{if } \tilde{z} \leq 0 \\ 1 & \text{else} \end{cases}$$

If one of the z_i is > 0 the sum of all z_i is ~~bigger~~ > 0.
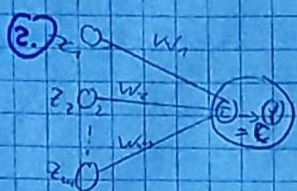
same as in sample solution

$$w_i = 10^{i-1} \quad \forall i \in \{1,..,n\}, \quad P(\tilde{z}) = \begin{cases} 0 & \text{if } \tilde{z}-c\cdot w \neq 0 \\ 1 & \text{else} \end{cases}$$
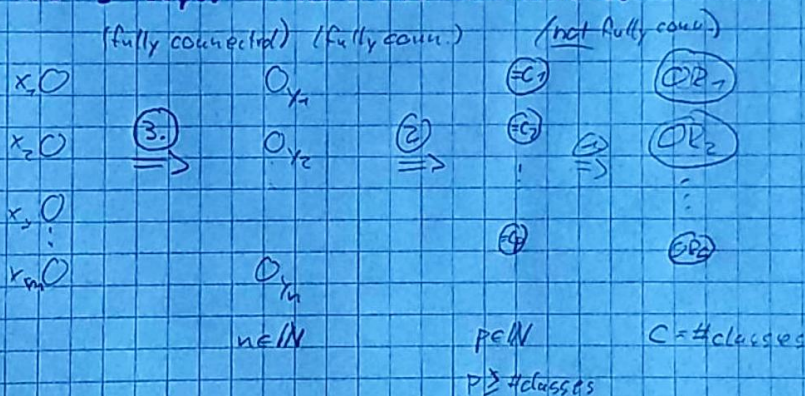
encode z as a decimal number and compare with c in same representation _different approaches but same result_

each ~~x~~ y-neuron has weight vector $y_i \sim B_i$

$$y_i = \text{step}(B_i \cdot z) = \begin{cases} 1 & B_i \cdot x > 1 \\ 0 & \text{else} \end{cases}$$

used definition of the lecture

2.2 3-layer Universal Classifier

(fully connected)   (fully conn.)   (not fully conn.)

$n \in \mathbb{N}$

$p \in \mathbb{N}$
$p \geq \#\text{classes}$

$C = \#\text{classes}$

First the data is split, such that each input vector is projected on a corner of a hyper-cube (dim n), where in one corner ~~eithare~~ are only members of one class. the c_i ~~encode~~ the layer reads of the bit representation such that on c_i is ~~an~~ always one and all c_j $j\neq i$ are 0. Since each corner only contains one class, but the members of one class can be projected onto different corners, we need

the OR-layer which will show the classifi-
cation of the instances. Each OR neuron repre-
sents one class, and is only connected
to the neurons of the previous layer, which
represents a corner (of the hypercube) of that
class. This way if the first layer is trained
there will be perfect classification of a training
set

Less explicit but same idea and order of
gates

1. logical OR operation on a binary input vector $z \in \{0,1\}^m$

$$\text{i.e. } z \rightarrow f(z) = \begin{cases} 0 & \forall i: z_i = 0 \\ 1 & \text{otherwise} \end{cases}$$

2. for an arbitrary but fixed binary vector $c \in \{0,1\}^m$ map the input vector $z \in \{0,1\}^m$ to

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

3. for the dataset X, Y displayed in Figure 1 (with feature vectors X_i and associated classes $Y_i \in$ { red minus, blue plus, green circle }) map every X_i onto the corners of a hypercube $\{0,1\}^m$ such that each corner contains only one class (you can map one class to multiple corners, but not multiple classes onto one corner). The dimension m of the hypercube is not a priori fixed and may be adjusted to fit the dataset. Draw the decision boundaries of your network in Figure 1 (you do not need to specify the precise equations of these boundaries) and indicate for each region to which hypercube corner it will be mapped. How can this be generalized to arbitrary many input dimensions and arbitrary (non degenerate) class distributions?
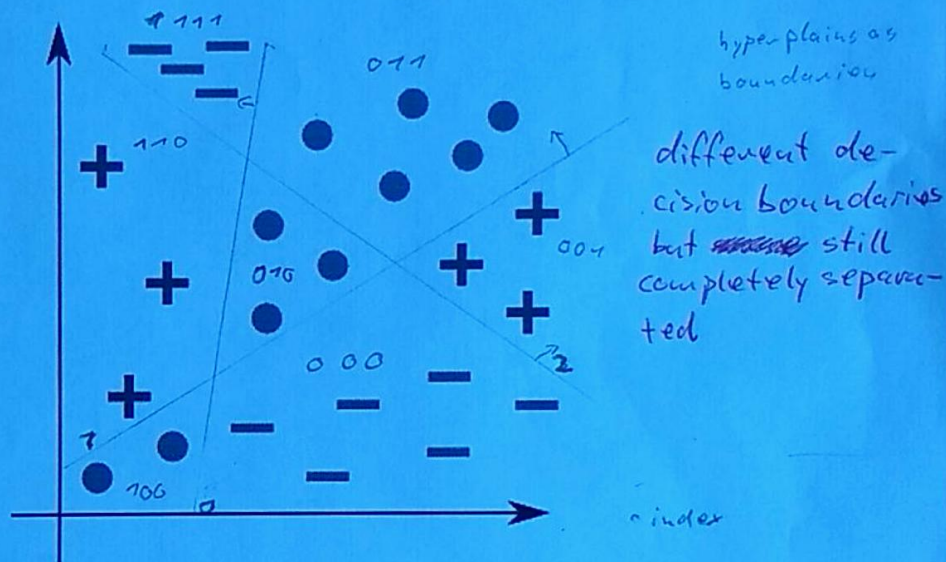

Abbildung 1: Example dataset with three classes (red minus, blue plus, green circle). Sketch the decision boundaries of your network here and draw the matching labeled hypercube.

2.2   3-layer Universal Classifier (5 Points)

Combine the networks from above into a universal classifier which classifies an arbitrary training set with zero training error. Draw a sketch of the network and explain in a few sentences how it works. Do you see any problems with this zero training loss classifier?