2) Classification Capacity (15P)

2.1 Simple Networks

1. We use the step function Θ as an activation function:

$$\Theta(u) = \begin{cases} 1 & \text{if } u \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

   Then we simply have $f(y) = \Theta(Wy)$, with $1 \times m$ weight matrix W, whereby $W_{1,k} = 1$ for all $k = 1, ..., m$

2. We define the $1 \times m$ weight matrix through $W_{1,k} = 2c_k - 1$, and a 1×1 bias $b = -\sum_{k=1}^{m} c_k$, and again use the step function:

$$g(y) = \Theta(Wy + b) \tag{2}$$

   How this works can be seen by rewriting the function as follows (simplifies to W, b above):

$$g(y) = \Theta\left(\sum_k (c_k - z_k)(1 - 2c_k)\right) \tag{3}$$

   The $(c_k - z_k)$ term is ± 1 if $c_k \neq z_k$. The second term is ∓ 1, so the product of the terms contributes -1 to the sum whenever $c_k \neq z_k$, and 0 otherwise. So the sum can only be zero if every $c_k = z_k$.

3. We use three linear decision boundaries for this task. Decision boundary (i) can be expressed throgh

$$h^{(i)}(x) = \Theta(W^{(i)}x + b^{(i)}) \tag{4}$$

   where the 1×2 weight matrix contains the normal vector of the boundary, and the bias term is the negative distance from the origin, in direction of the normal (see figure).

   For the three boundaries in total, we therefore have a 3×2 weight matrix $W = [W^{(1)}, W^{(2)}, W^{(2)}]$, and a 3-vector $b = (b^{(1)}, b^{(2)}, b^{(3)})$. The output of

$$h(x) = \Theta(Wx + b) \tag{5}$$

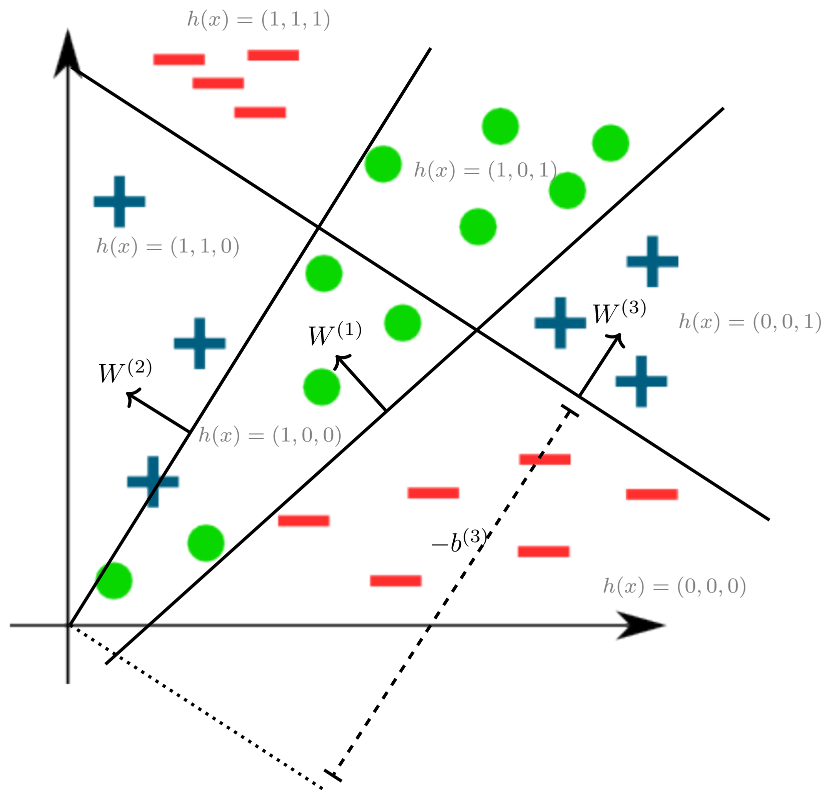
   will therefore be $\in \{0, 1\}^3$, indicating on which side of each boundary the point lies (see figure).

   We can sketch a simple algorithm that does this in general, in d-dimensional space, with a set $\mathcal{X} = \{x^{(i)}\}$ of N datapoints and K classes:

   (1) Find a d-dimensional linear decision boundary so that the points on one side only come from a single class (guaranteed to be possible, because we can just split off a single point in the worst case)

(2) If the points on the other side are also all from a sinlge class, we are done. Otherwise, go back to step (1) for these points.

This will alsways terminate in $\leq N$ iterations, because the number of remaining points is reduced by at least 1 each iteration.



2.2 3-layer Universal Classifier

For this task, we use a 3-layer network $f\Big(g\big(h(x)\big)\Big)$, with the one-layer networks from above.

- $h(x)$ maps the coordinates x to a $z \in 0, 1^3$ like in Question 2.1.3, as shown in the figure.
- $g(z)$ maps all 8 possible $z \in \{0, 1\}^3$ to a $\{0, 1\}^8$ vector y, which indicates which z was the input, i.e. what partition of x space the point lies in. Each row corresponds to the results from Question 2.1.2 (in binary sorted

order):

$$W = \begin{pmatrix} -1 & -1 & -1 \\ -1 & -1 & 1 \\ -1 & 1 & -1 \\ & \vdots & \\ 1 & 1 & 1 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 0 \\ -1 \\ -1 \\ \vdots \\ -3 \end{pmatrix} \tag{6}$$

- $f(y)$ combines all individual regions belonging to one class through the `or` operation, equivalent to Question 2.1.1. For the shown example (green circle: class 1, blue plus: class 2, red minus: class 3), the weight matrix will be

$$W = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

In practice, this approach will of course lead to overfitting, and poor generalization. For the 2D case shown, we at least see some generalization, but e.g. for image classification it is sure to fail completely.

3) Linear Activation Function

A linear activation function can only be scaling with some scalar s. We now construct a one-layer-network that simply computes

$$\tilde{Z} = AZ_0 \quad \text{with} \quad A \equiv s^L \prod_{l=1}^{L} B_l \tag{8}$$

We then show that $\tilde{Z} = Z_L$, with Z_L from the original network. This can be seen by recusively repeating the following operation:

$$\begin{aligned} Z_1 =& sB_1 Z_0 & \equiv A_1 Z_0 \quad (9) \\ Z_2 =& sB_2 Z_1 = sB_2 A_1 Z_0 & \equiv A_2 Z_0 \quad (10) \\ Z_3 =& sB_3 Z_2 = sB_3 A_2 Z_0 & \equiv A_3 Z_0 \quad (11) \\ \vdots & & \vdots \\ Z_L =& sB_L A_{L-1} Z_0 & = AZ_0 \quad (12) \end{aligned}$$

$\square$