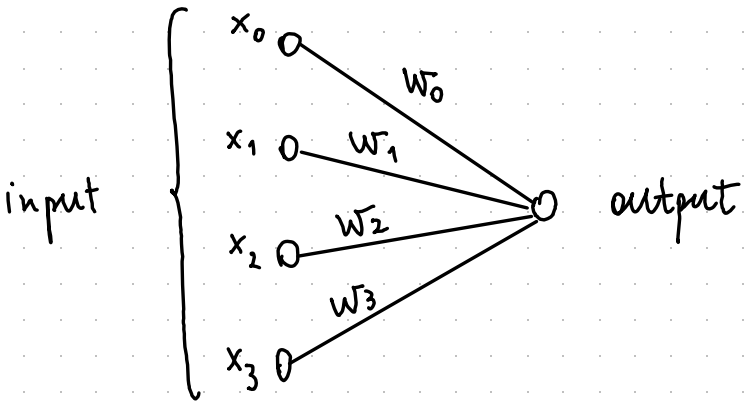
NEURAL NETWORKS

We saw linear regression.
The linear regression model works like this:

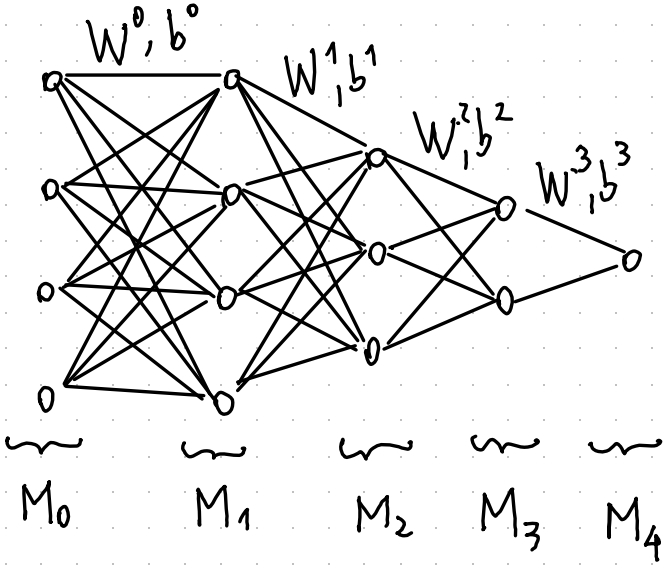
- $x \in \mathbb{R}^{1 \times M}$ feature vector
- $w \in \mathbb{R}^{M \times 1}$ weights
- $b \in \mathbb{R}^{1 \times 1}$ bias
- $y = xw + b \in \mathbb{R}^{1 \times 1}$ output

We can visualize it as follows



What if we want to add more complexity?

<u>Idea</u> (does not work) we can add more parameters in a layered structure



layer 0
$$\begin{cases} x^0 \text{ input} \\ y^0 = x^0 W^0 + b^0, \quad W^0 \in \mathbb{R}^{M_0 \times M_1}, \; b^0 \in \mathbb{R}^{1 \times M_1} \end{cases}$$

layer 1
$$\begin{cases} x^1 = y^0 \\ y^1 = x^1 W^1 + b^1, \quad W^1 \in \mathbb{R}^{M_1 \times M_2}, \; b^1 \in \mathbb{R}^{1 \times M_2} \end{cases}$$

...

Is this adding more complexity and expressivity to the model?

No. It's always a linear model, just with more parameters

$$y^l = x^l W^l + b^l =$$
$$= y^{l-1} W^l + b^l =$$
$$= (x^{l-1} W^{l-1} + b^{l-1}) W^l + b^l =$$
$$= x^{l-1} \underbrace{W^{l-1} W^l}_{\in \mathbb{R}^{M_{l-1} \times M_{l+1}}} + \underbrace{b^{l-1} W^l + b^l}_{\in \mathbb{R}^{1 \times M_{l+1}}}$$

$$= \dots$$

$$= x^0 W^0 W^1 \dots W^l + \text{bias}$$

How do we add more complexity?

With nonlinearities.

$$y^l = \phi(x^l W^l + b^l)$$

↑ Nonlinear function

See perceptron for the interpretation.