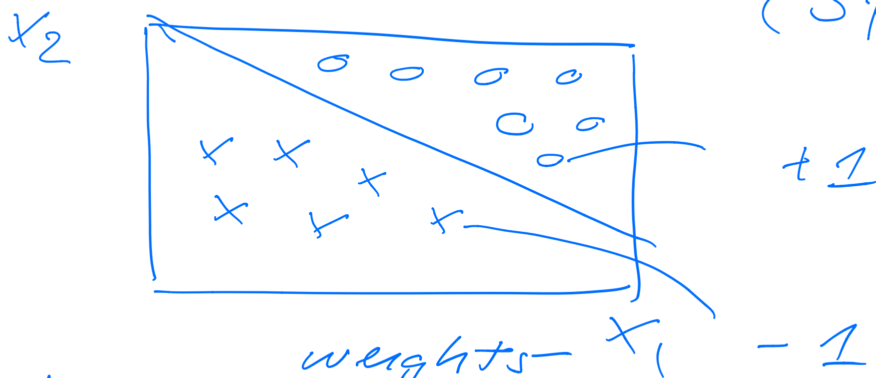
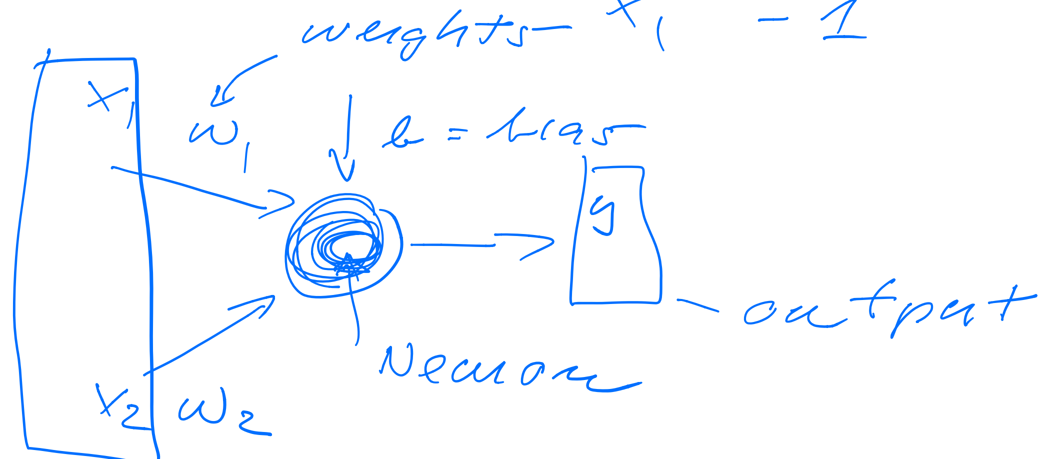
- Back propagation algo and Neural network (NN)

- single perceptron model (SPM)





input Layer

$$y = x_1 w_1 + x_2 * w_2 + b$$

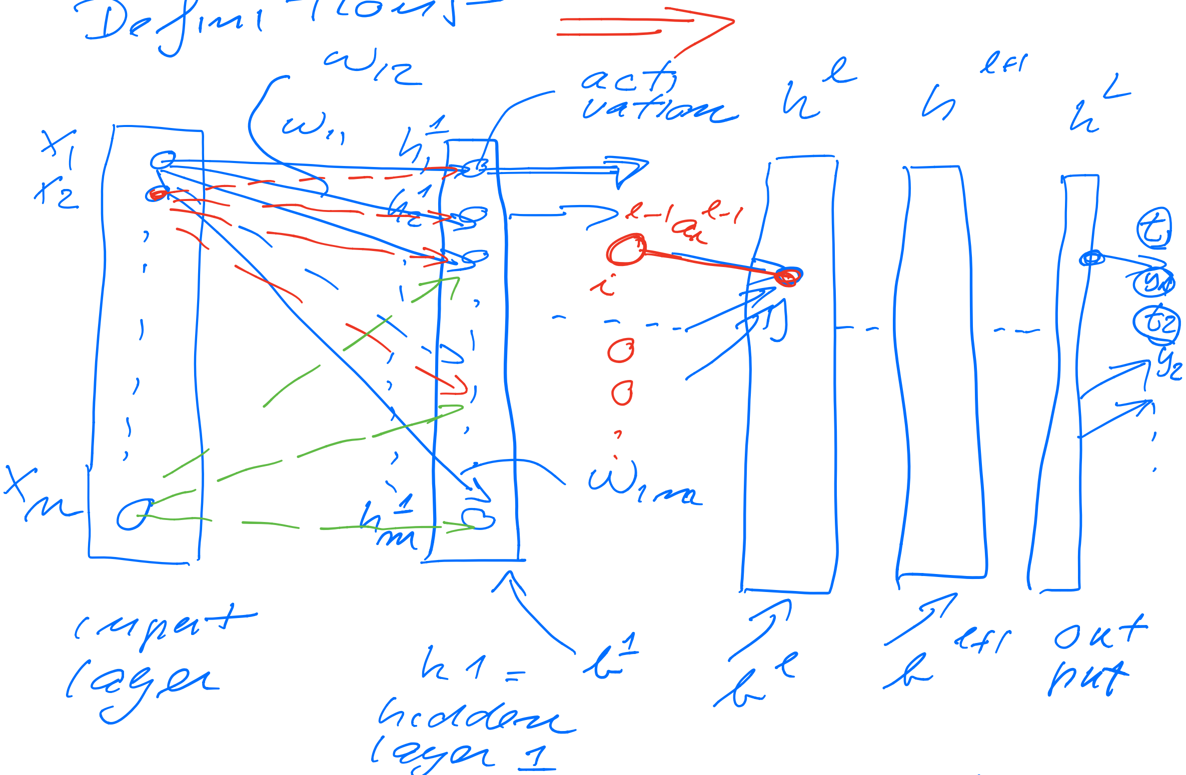
$$y = \{-1, 1\}$$

SPM can be used to fit a first-order polynomial, How can we train more complicated functions? Universal approx theorem states that a neural network with one or more hidden

Layers can approximate functions beyond first-order

Definitions

Feed Forward



targets/outputs are labeled, t_i

Regression case

Cost function $C(W, b) = \frac{1}{2}\sum_{i=1}^{n}(y_i - t_i)^2$

weight matrix

From the network

Input to node j in layer l

$$\underline{z_j^l} = \sum_{i=1}^{M_{l-1}} w_{ij}^l \boxed{a_i^{l-1}} + b_j^l$$

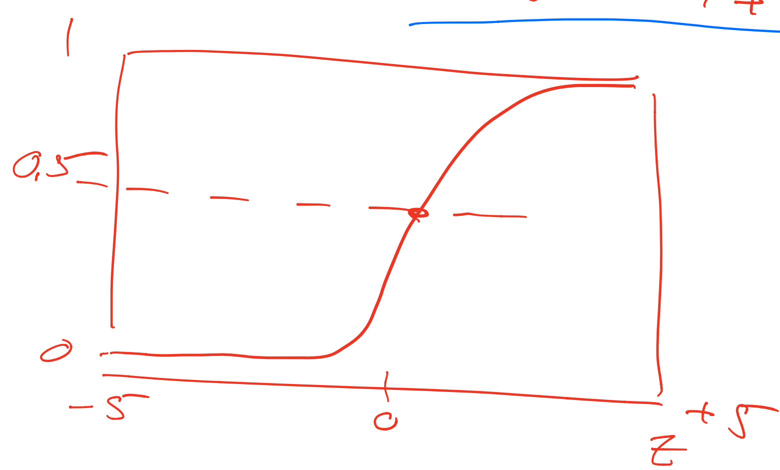
output
from node i
in layer $l-1$

$$\boxed{\text{output } a_j^l} = f(z_j^l)$$

activation function

classical case
sigmoid

$$f(z_j^l) = \frac{1}{1+e^{-z_j^l}}$$



We want $\dfrac{\partial C}{\partial w_{ij}}$ & $\dfrac{\partial C}{\partial b_j}$

- chain rule massage

$$\frac{\partial z_j^l}{\partial w_{ij}^l} = a_i^{l-1} = f(z_i^{l-1})$$

∂w_{ij}

$$\frac{\partial z_j^l}{\partial a_i^{l-1}} = w_{ij}^l$$

$$\frac{\partial a_j^l}{\partial z_j^l} = f(z_j^l)(1 - f(z_j^l))$$

output layer $l = L$

$$C(W^L, b^L) = \frac{1}{2}\sum_{i=1}^{m}(y_i - t_i)^2$$
$$= \frac{1}{2}\sum_{i=1}^{m}(a_i^L - t_i)^2$$

$$\frac{\partial C}{\partial w_{jk}^L} = (a_j^L - t_j)\frac{\partial a_j^L}{\partial w_{jk}^L}$$

$$\frac{\partial a_j^L}{\partial w_{jk}^L} = \frac{\partial a_j^L}{\partial z_j^L}\frac{\partial z_j^L}{\partial w_{jk}^L}$$

$$= a_j^L(1 - a_j^L)\, a_k^{L-1}$$

$$\frac{\partial C}{\partial w_{jk}^L} = (a_j^L - t_j)\, a_j^L(1 - a_j^L)\, a_k^{L-1}$$

$$\delta_j^L = a_j^L(1-a_j^L)(a_j^L - t_j)$$

$$= f'(z_j^L)\frac{\partial C}{\partial a_j^L}$$

$$\boxed{\frac{\partial C}{\partial w_{jk}^L} = \delta_j^L a_k^{L-1}}$$

$$\delta_j^L = \frac{\partial C}{\partial z_j^L} = \frac{\partial C}{\partial a_j^L}\frac{\partial a_j^L}{\partial z_j^L}$$

$$\boxed{\delta_j^L = \frac{\partial C}{\partial b_j^L}\frac{\partial b_j^L}{\partial z_j^L} = \frac{\partial C}{\partial b_j^L}}$$