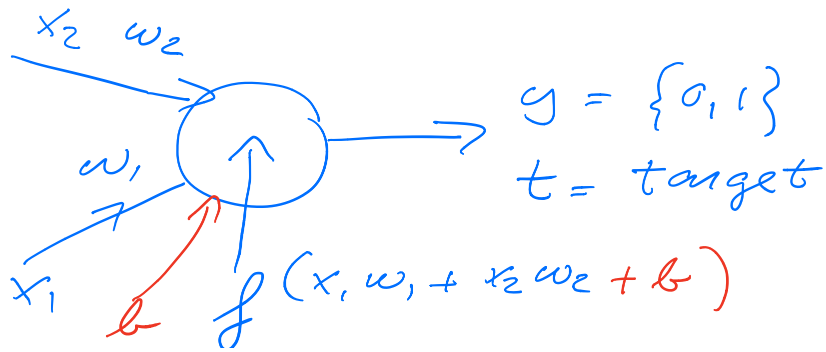
Lecture October 2

Feed Forward NN / Multi-layer perceptron.

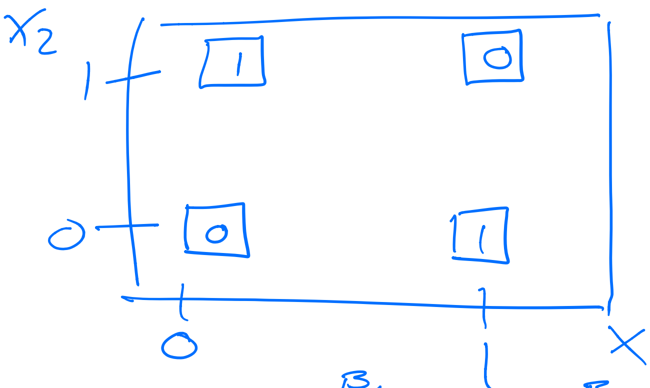
- Architecture
  - input layer
  - given # of hidden layers with various nodes/units/neurons
  - output
- Training $\Theta = \{w, b\}$
  - Feed Forward
  - Backpro (sets up gradients)
  - SGD
  - Model (MSE, cross-entropy) compare output with targets

Single perceptron



XOR

x_1	x_2	t
0	0	0
0	1	1
1	0	1
1	1	0



$$\overset{\beta_1}{x_1 w_1} + \overset{\beta_2}{x_2 w_2} + \overset{\beta_0}{\boxed{b}} = y$$

Linear regression

$$X = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \\ \begin{array}{ccc} \uparrow & x_1 & x_2 \\ \beta_0 = b & & \end{array} \end{array}$$

$$\hat{\beta} = (X^T X)^{-1} X^T y = (X^T X)^{-1} X^T t$$

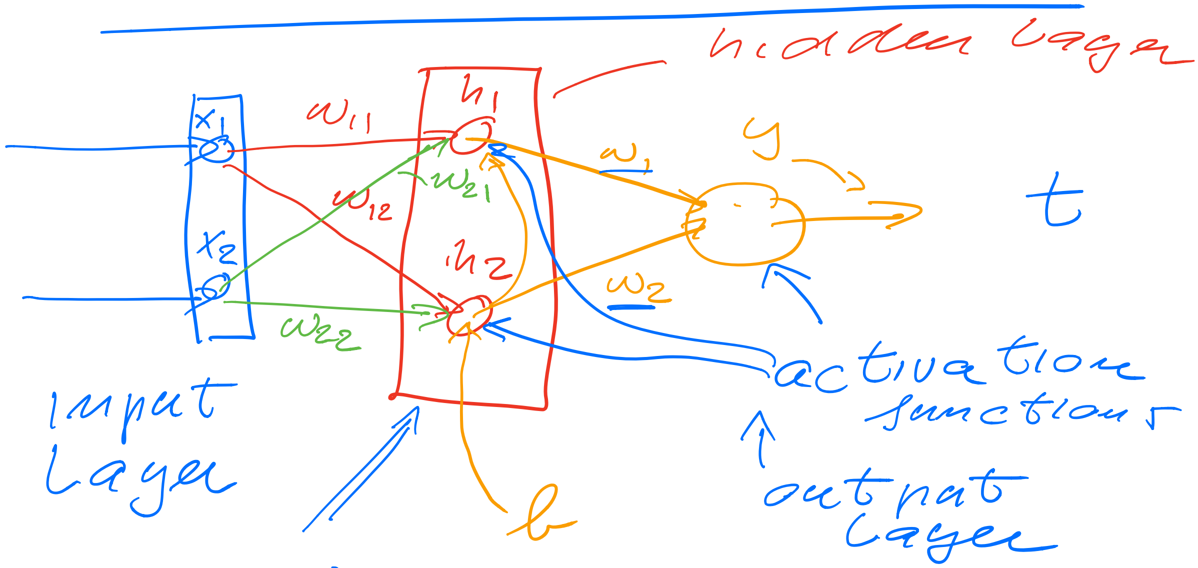
$$X^T X = \begin{bmatrix} 4 & 2 & 2 \\ 2 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix}$$

$\hat{}$ [0.5]

$$\beta = \begin{bmatrix} \cdot \\ 0 \\ 0 \end{bmatrix} \qquad b = 1/2, \quad w_1 = w_2 = 0$$

$$y = x_1 w_1 + x_2 w_2 + b = 1/2$$

simple neural network



$$f^{(1)}(\underline{WX + b})$$

$$f^{(2)}(\underline{h}\, w)$$

$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \qquad w = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

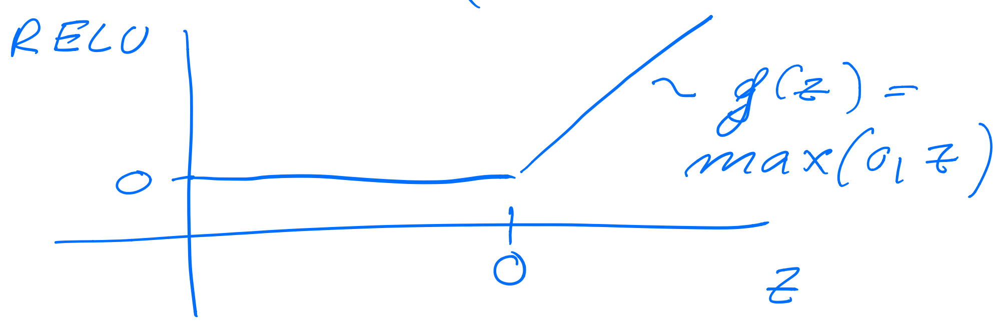
$$XW + b = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

$$f^{(1)}(XW + b)$$

- activation function

$f^{(1)}$ - RELU

$$= \max(0,\ WX + b)$$



$f^{(1)}$ becomes

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} = h$$

$$f^{(2)} = h \cdot w = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = y \equiv t$$

Backpropagation algo

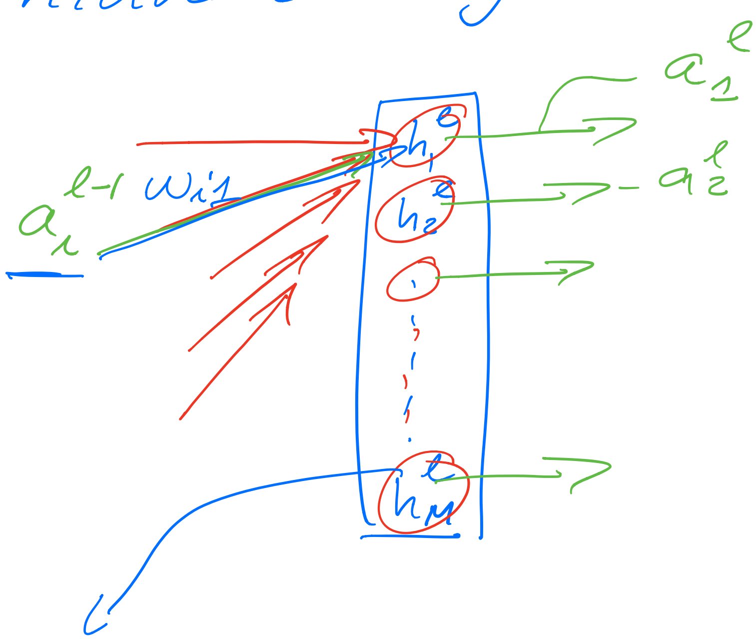
Backprop: chain rule to obtain the gradients

SGD: training of weights and biases: $\Theta = \{w, b\}$

Regression case

$$\underline{C(\Theta)} = \frac{1}{2}\sum_{i=1}^{n}(\underbrace{y_i}_{a_i^L} - t_i)^2$$

hidden layer -l-



$$f(z_j^l) = a_j^l$$

$$\underline{z_j^l} = \sum_i w_{ij}^l a_i^{l-1} + b_j^l$$

$$z^l = [W^l]^T a^{l-1} + b^l$$

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

Chain Rule

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

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

$$= a_j^l(1-a_j^l) =$$

$$f(z_j^l)(1-f(z_j^l))$$

$$C(W,b) = \frac{1}{2}\sum(a_i^L - t_i)^2$$

$l = L$ = output layer.

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

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

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

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

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

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

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

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

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

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

$L \to l$

$$\delta_j^l = \frac{\partial C}{\partial z_j^l}$$

can we define it in terms of layer $l+1$?

$$\delta_j^l = \sum_k \frac{\partial C}{\partial z_k^{l+1}} \cdot \frac{\partial z_k^{l+1}}{\partial z_j^l}$$

$$= \sum_k \delta_k^{l+1} \cdot \frac{\partial z_k^{l+1}}{\partial z_j^l}$$

$$z_j^{l+1} = \sum_{i=1}^{M_l} w_{ij}^{l+1} a_i^l + b_j^{l+1}$$

$$\delta_j^l = \sum_k \delta_k^{l+1} w_{kj}^{l+1} f'(z_j^l)$$

$$\boxed{w^l \leftarrow w^l - \eta \delta_j^l a_k^{l-1}}$$

SGD

$$\boxed{\begin{aligned} w_{jk} &\leftarrow w_{jk} - \gamma \delta_j x_k \\ b_j^e &\leftarrow b_j^e - \gamma \delta_j^e \end{aligned}}$$