(1) **Neuhálo II konz** 2018.10.04.

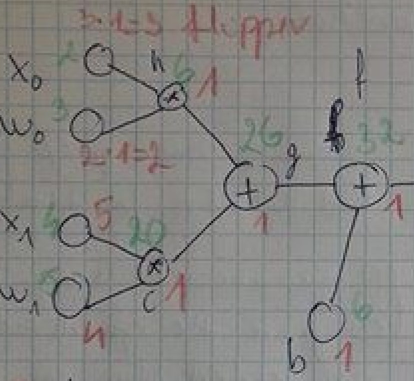
- ZH-ban:
  - Back prop számolás → számológép kell (ugyanolyan számolás)
  - Rosenblatt XOR, OR, AND megoldás 1-2 iterációval
  - Elméleti kérdések, pl mi az a feedforward?

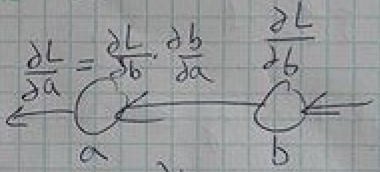
kiinduló globális gradiens

$\frac{\partial L}{\partial L} = 1$ grad fel* / grad out / grad vector

Loss: hiba fgv

- Számolási feladat:





$a' = \frac{\partial b}{\partial a}$ chain rule

■ : back prop
■ : feed forw.

$f = g + b$ $f' = \frac{\partial f}{\partial f} = 1$ $b' = \frac{\partial f}{\partial b} = \frac{\partial f}{\partial f} \cdot \frac{\partial f}{\partial b} = 1$

$g' = \frac{\partial f}{\partial g} = \frac{\partial f}{\partial f} \cdot \frac{\partial f}{\partial g} = 1$ $g = h + i$

$h = w_0 \cdot x_0$

$h' = \frac{\partial f}{\partial h} = \frac{\partial f}{\partial f} \cdot \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial h} = 1$

$ReLu' = \begin{cases} 1 & x > 0 \\ 0 & \text{else} \end{cases}$

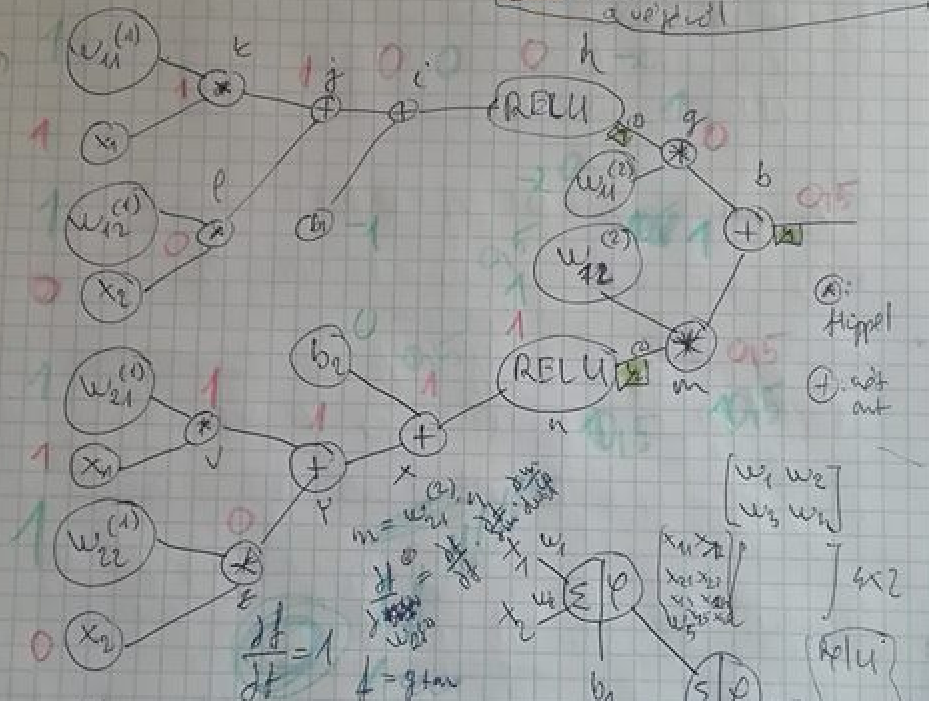
Lokális hiba:

$$\delta = \begin{cases} e'(1_i^{(m)}) \cdot (d_i - y_i^{(m)}) & \text{output layer} \\ e'(1_i^{(m)}) \cdot \left(\sum_j^N w_{ij} \cdot \delta_j^{(m+1)}\right) & \text{hidden layer} \end{cases}$$

outputban

$$w_{ij}^{(m)}(t) = w_{ij}(t) + \eta \cdot \delta_{ij}^{(m)} \cdot y_i^{(m-1)}$$

ZH: visszafelé propagáció (hatás) a végéről

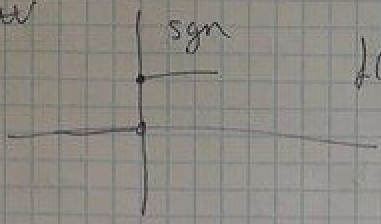


$$\frac{\partial f}{\partial f} \cdot \frac{\partial f}{\partial g} = \frac{\partial f}{\partial g} \qquad \frac{\partial f}{\partial m} = 1$$

A fa végén mindig 1 van.

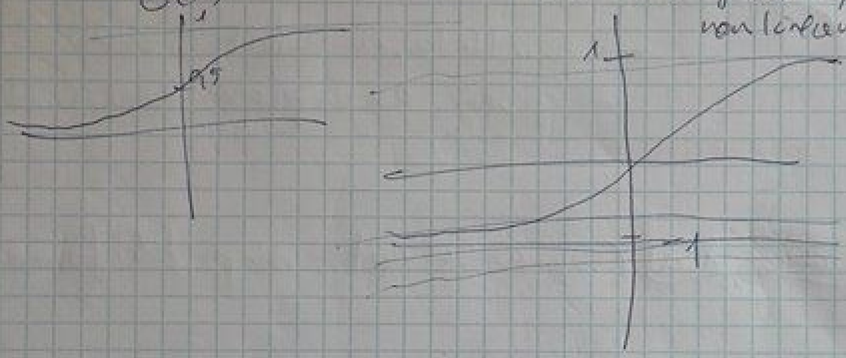
$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial f} \cdot \frac{\partial f}{\partial m} \cdot \frac{\partial m}{\partial n} \cdot \frac{\partial n}{\partial x} = 0.5 \qquad \frac{\partial f}{\partial n} = \frac{\partial f}{\partial f} \cdot \frac{\partial f}{\partial m} \cdot \frac{\partial m}{\partial n} = w_{21}^{(2)}$$

- fogalom tár: a kiemelt dolgok
  - Artificial neuron: node, ports, modell
  - activation func, bias

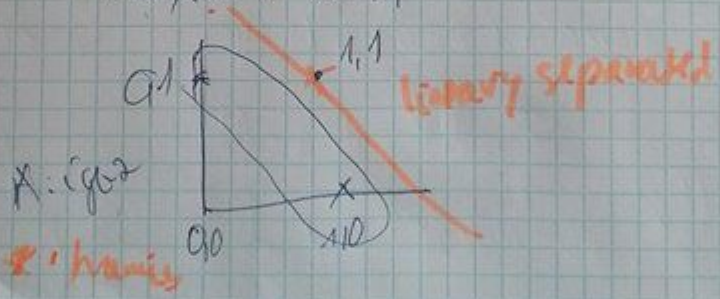


$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

sigmoid linearity vs sigmoid nonlinearity



- element set seperation
  (bináris fgv. lineárisan szeparálható-e
  bizonyítani is kell



4.

— Rosen-Bach algorithm
↳ táblás volt

How it works and we should use it.

— Perceptron convergence (nem kell bizonyítani)

~~Multistep~~

$w_0 = b$

x_1 x_2

XOR $x = [1\ \ 1\ -1]$

Loss function = error function

— multilayer what is it?
— given examples how is it working?

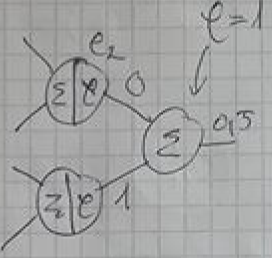
Some did (summa school: deep feed forward network)

Gyak anyagot nézd át!! Logikai fgv.

$$\varphi(u) = \frac{1}{1+e^{-u}} \qquad \varphi_a(u) = \frac{1}{1+e^{-au}}$$

$$\varphi'(u) = \varphi(u)(1-\varphi(u)) \qquad \varphi_a'(u) = a \cdot \varphi(u)(1-\varphi(u))$$

1.) FORWARD PROPAGATION
2.) ERROR CALCULATION
3.) LOCAL ERROR CALCULATION
4.) UPDATE WEIGHTS



(1) $y_3 = 0{,}5 \quad y_2 = 1 \quad y_1 = 0$ (másik oldal)

(2) $E_i = d_i - o_i \;\equiv\; E_1 = 1 - 0{,}5 = 1/2$

lokális hiba az 1. neuronon

(3) $e_1^{(2)} = e_3 = 1 \cdot 1/2 = 1/2 \leftarrow$ [illegible]

$e_1^{(1)} = e_2 = ReLU'(0) = 0 \leftarrow e_2$

$e_2^{(1)} = e_1 = ReLU'(1) \cdot 0{,}5 \cdot 0{,}5 = 0{,}25$

lok hiba ↑ súly ↑ visszapropagált hiba

(4) $w_{ij}^{(m)}(1) = w_{ij}^{(m)} + \eta \cdot e_i^{(m)} \cdot y_i^{(m-1)}$

$w_{12}^{(2)} = 0{,}5 + 1 + e_3 \cdot 1 = 1 \leftarrow$ erre javítunk

$w_{11}^{(2)} = -2 + 1 \cdot e_1^{(2)} \cdot 0 = -2$

ZH: • kapsz egy gráfot: deriváltakat kell számolni
• 4 cs ZH [illegible] hozzáld