4 Neural Networks

p 6

$$f\left(\sum w_i x_i\right) = f(\vec{x}^T \vec{w}) \quad \text{with } \vec{x} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \quad \vec{w} = \begin{pmatrix} w_0 \\ x_1 \end{pmatrix}$$

p 8

belongs to neuron 1

$$o_1 = f(\vec{x}^T \vec{w}_1)$$

$$\vdots$$

$$o_k = f(\vec{x}^T \vec{w}_k)$$

elements in the vector

define: $W = \begin{bmatrix} - & \vec{w}_1 & - \\ & \vdots & \\ - & \vec{w}_k & - \end{bmatrix}$ & $\vec{F} = \begin{bmatrix} f(v_1) \\ \vdots \\ f(v_k) \end{bmatrix}$

then: $\underline{\vec{o} = \vec{F}(W\vec{x})}$

... p 11 (3)

compute an update

1) forward pass: we need to know "y"

$$y = w_3 \cdot \max(0, w_1 x_1 + w_2 x_2)$$

$$y = \underline{4}$$

$$\cdot \left.\frac{de}{dy}\right|_{y=4} = -2(2-4) = \underline{4}$$

$$\cdot \left.\frac{dy}{do_1}\right|_{w_3=1} = 1$$

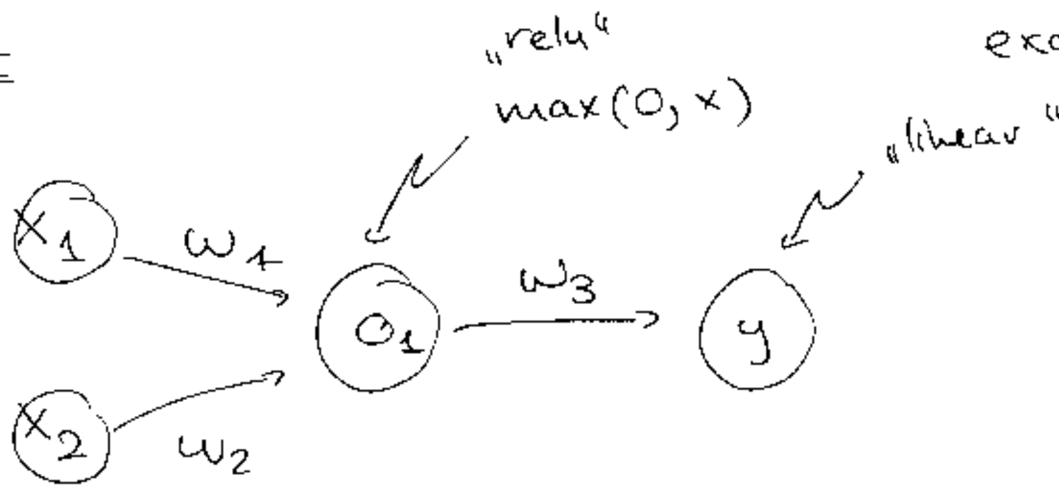
$$\cdot \left.\frac{do_1}{dw_1}\right|_{w_1, w_2, x_1, x_2} = 1 \cdot 2 = \underline{2}$$

$$\frac{de}{dw_1} = 4 \cdot 1 \cdot 2 = \underline{8}$$

$$w_1 \leftarrow w_1 - \eta \cdot \frac{de}{dw_1} = 1 - 0.8 = \underline{\underline{0.2}}$$

p 11

example ③



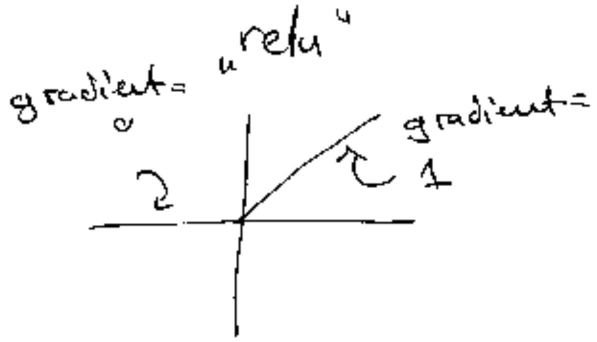
$$o_1 = \max(0,\ w_1 x_1 + w_2 x_2)$$

$$y = w_3 o_1$$

$$e = (y_{target} - y)^2$$

$$\frac{de}{dw_1} = \underbrace{\frac{de}{dy} \cdot \frac{dy}{do_1} \cdot \frac{do_1}{dw_1}}_{\text{„back propagation“}}$$

- $\frac{de}{dy} = -2(y_{target} - y)$



- $\frac{dy}{do_1} = w_3$

- $\frac{do_1}{dw_1} = \begin{cases} 1 \cdot x_1 & \text{if } w_1 x_1 + w_2 x_2 > 0 \\ 0 & \text{else} \end{cases}$

P 13

- Prediction
- Backpropagation for optimization
- yes, for the error computation

P 14

- Non, functions aren't convex
- Slope: we move along the steepest slope until we reach the lowest point of the valley!
- Gradient $\nabla f = \begin{bmatrix} \frac{df}{dx_1} \\ \vdots \end{bmatrix}$

P 15

- $o_1 = f(\vec{w}_1^T \vec{x})$; $y = \vec{w}_2^T \vec{o}$

$$\vec{w}_1 = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \quad \vec{w}_2 = \begin{bmatrix} w_3 \\ w_4 \end{bmatrix}$$

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad o = \begin{bmatrix} x_3 \\ o_1 \end{bmatrix}$$

$$o_1 = f(w_1 x_1 + w_2 x_2)$$

$$y = w_3 x_3 + w_4 o_1$$

$$e = (y_{target} - y)^2$$

$$\frac{de}{dw_1} = \frac{de}{dy} \cdot \frac{dy}{do_1} \cdot \frac{do_1}{dw_1}$$

... p15

$$\frac{de}{dy} = -2(\bar{y} - y)$$

$$\frac{dy}{do_1} = w_4$$

$$\frac{do_1}{dw_1} = \begin{cases} x_1 & \text{if } w_1x_1 + w_2x_2 > 0 \\ 0 & \text{else} \end{cases}$$

update:

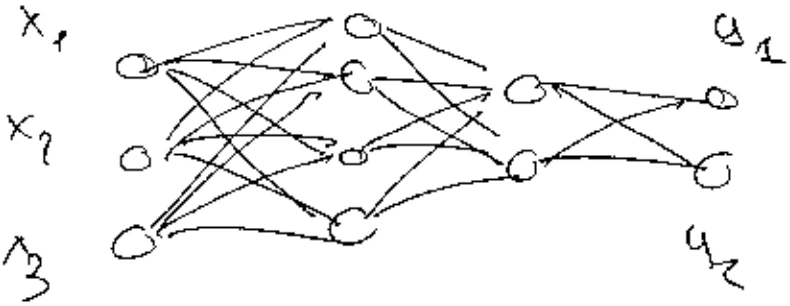
$$y = 1 \cdot 1 + 1 \cdot f(1 \cdot 1 + 1 \cdot 1) = \underline{3}$$

$$\frac{de}{dw_1} = 2 \cdot 1 \cdot 1 = \underline{\underline{2}}$$

$$\nabla e = \begin{bmatrix} \frac{de}{dw_1} \\ \frac{de}{dw_2} \\ \frac{de}{dw_3} \\ \frac{de}{dw_4} \end{bmatrix} \begin{array}{l} \longleftarrow \text{already done} \\ \longleftarrow \text{only changes } \frac{do_1}{dw_2} \\ \longleftarrow \frac{de}{dy} \cdot \frac{dy}{dw_3} \\ \longleftarrow \frac{de}{dy} \cdot \frac{dy}{dw_4} \end{array}$$

$$\left.\begin{array}{l} \frac{do_1}{dw_2} = \begin{cases} x_2 & \text{if } w_1x_1 + w_2x_2 > 0 \\ 0 \end{cases} \\ \frac{dy}{dw_3} = x_3 \\ \frac{dy}{dw_4} = o_1 \end{array}\right\} \text{all missing terms}$$

P 1 6



"Connecting neuron 1 to neuron 1"

"Connecting neuron 2 to neuron 1" etc.

$$\vec{o}_1 = \vec{f}(w_1 \vec{x}) \qquad w_1 = \begin{bmatrix} w_{11} & w_{21} & w_{31} \\ \vdots & & \\ w_{14} & \cdots & w_{34} \end{bmatrix}_1$$

$$\vec{o}_2 = \vec{f}(w_2 \vec{o}_1) \qquad w_2 = \begin{bmatrix} w_{11} & w_{21} & w_{31} & w_{41} \\ w_{12} & w_{22} & w_{32} & w_{42} \end{bmatrix}_2$$

$$\vec{y} = \vec{f}(w_3 \vec{o}_2 \qquad w_3 = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix}_3$$

$$\nabla e = \begin{bmatrix} \frac{de}{dw_{11}^1} \\ \frac{de}{dw_{21}^1} \\ \vdots \\ \frac{de}{dw_{22}^3} \end{bmatrix}$$

denotes the layer

$\Rightarrow$ simply all weights in w_1, w_2, w_3

P 18

⇝ nothing new; do as in prev. examples

⇒ one thing to note: We are using the logistic function as activation not the ReLU:

$$\boxed{\sigma'(x) = \sigma(x)\,(1 - \sigma(x))}$$

(also has an easy gradient)