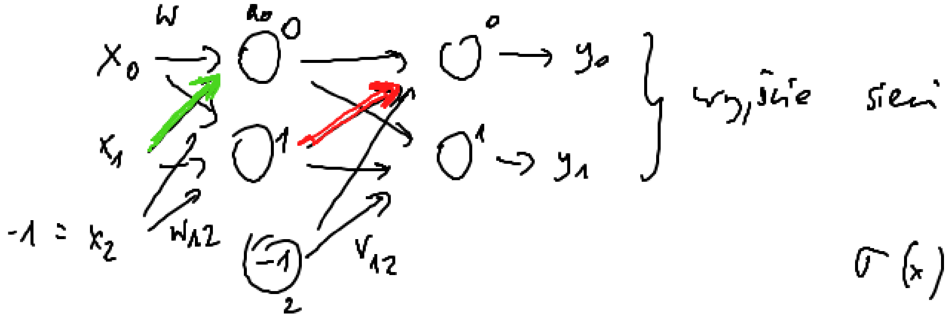
Propagacja wsteczna



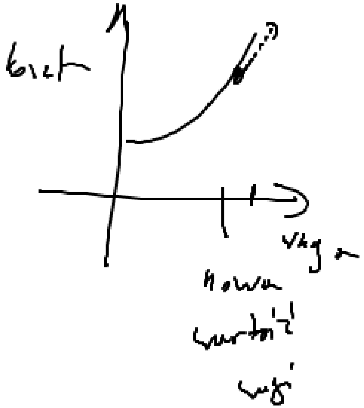
$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$a_j = \sigma\left(\sum_{k=0}^{2} w_{jk} x_k\right), \quad j = 0,1$$

$$\sigma(x) = x_+ = \max(x, 0) \quad \text{ReLU}$$

$$a_2 := -1$$

$$y_j = \sigma\left(\sum_{k=0}^{2} v_{jk} a_k\right), \quad j = 0,1$$



$$y_j = \sigma\left(\sum_{k=0}^{2} v_{jk}\, a_k\right)$$

$$L(t,y) = \frac{1}{2}\|t-y\|_2^2 = \frac{1}{2}(t_0-y_0)^2 + \frac{1}{2}(t_1-y_1)^2 \qquad t=(t_0,t_1) \qquad y=(y_0,y_1)$$

$$L(t,y) = -\sum_{i=0}^{1} t_i \log y_i \qquad (0\cdot\log 0 = 0)$$

Ustalmy j_0, k_0, $v := v_{j_0,k_0}$.

$$\frac{\partial L}{\partial v} = \frac{\partial L(t,y)}{\partial y_0}\cdot\frac{\partial y_0}{\partial v} + \frac{\partial L(t,y)}{\partial y_1}\frac{\partial y_1}{\partial v} = \frac{\partial L(t,y)}{\partial y_{j_0}}\frac{\partial y_{j_0}}{\partial v} = \underbrace{\boxed{\frac{\partial L(t,y)}{\partial y_{j_0}}\cdot\sigma'\left(\sum_{k=0}^{2} v_{j_0 k}\, a_k\right)}}_{\rightsquigarrow\ -\delta^{(2)}}\, a_{k_0}$$

$$\text{nowe } v = v - \underset{\substack{\uparrow\\ \text{stała uczenia}}}{c}\,\frac{\partial L}{\partial v}$$

$$a_m = \sigma\left(\sum_{n=0}^{2} w_{mn}\, x_n\right), \quad m=0,1$$

$c=0.01$

$c=0.1$

„Optimizer"

Ustalmy m_0, n_0, $w := w_{m_0,n_0}$.

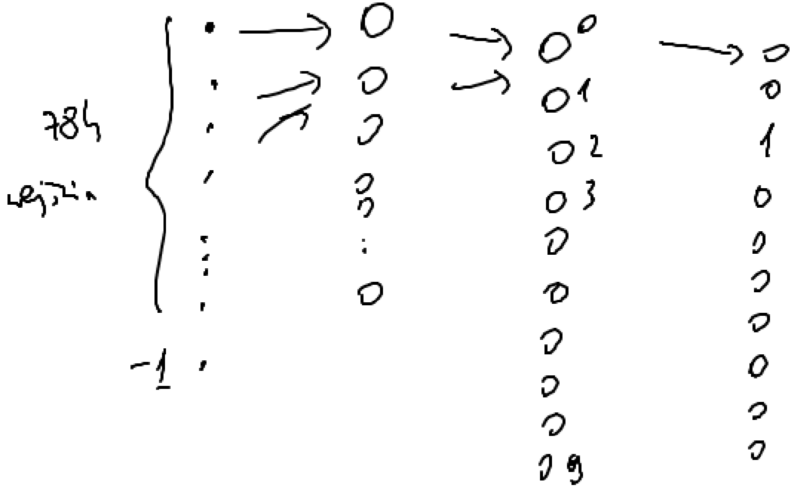
$$\frac{\partial L}{\partial w} = \sum_{j=0}^{1}\frac{\partial L(t,y)}{\partial y_j}\cdot\frac{\partial y_j}{\partial w} = \sum_{j=0}^{1}\frac{\partial L(t,y)}{\partial y_j}\,\sigma'\left(\sum_{k=0}^{2} v_{jk}\, a_k\right)\cdot\sum_{k=0}^{1} v_{jk}\frac{\partial a_k}{\partial w} = \sum_{j=0}^{1}\boxed{\frac{\partial L(t,y)}{\partial y_j}\,\sigma'\left(\sum_{k=0}^{2} v_{jk}\, a_k\right)}\, v_{jm_0}\cdot\frac{\partial a_{m_0}}{\partial w} =$$

$$= \underbrace{\overbrace{\sum_{j=0}^{1}\boxed{\frac{\partial L(t,y)}{\partial y_j}\,\sigma'\left(\sum_{k=0}^{2} v_{jk}\, a_k\right)}\, v_{jm_0}}^{-\delta^{(1)}}\cdot\,\sigma'\left(\sum_{n=0}^{2} w_{mn}\, x_n\right)}_{\rightsquigarrow\ -\delta^{(1)}}\cdot\, x_{n_0}$$

$$\text{nowe } w = w - c\,\frac{\partial L}{\partial w}$$

Zadanie

Oczekiwany np. dla cyfry 2 wyjścia:



Dane: (x_i, t_i), $i=0,1,\dots,M-1$

x_i – wejście, t_i – oczekiwane wyjście („etykieta")

Łączny koszt: $\sum_{i=0}^{M-1} L(t_i, y_i)$

y_i – wyjście sieci dla x_i

Używanie łącznego kosztu jest zbyt kosztowne obliczeniowo.

- 'online learning': bierzemy po kolei dane $x_0, x_1, \dots, x_{M-1}$
  i uaktualniamy wagi po każdej

- kompromis: 'batch learning': bierzemy grupki danych $(x_0, x_1, \dots, x_{b-1}), \dots$
  i uaktualniamy wagi po każdej grupie

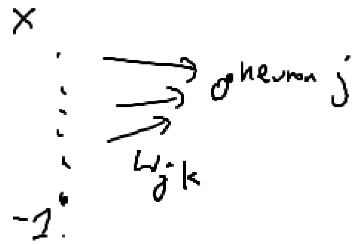
- danych mamy zwykle za mało, dlatego używamy ich wielokrotnie:
  warto dane wejściowe przetasować po każdej epoce

```
for epoch in ... :
    for x in ... :
        ...
    shuffle(x)
```

• inicjacja wag:

wagi powinny zostać zainicjowane w różny sposób dla różnych neuronów

— losowo



Nasze zadanie:

np. jeśli $w \sim N(0, \underset{\text{wariancja}}{1})$ (iid) itd

Lepiej: $w \sim N\left(0, \left(\frac{1}{\sqrt{785}}\right)^2\right)$

wyjście z neuronu j w warstwie 1: $= \sigma\left(\sum_{k=0}^{785} w_{jk} x_k\right)$

x_k – dane obrazka, $\in [-1,1)$

gdyby połowa x_k była równa 0, a połowa 1:

$\sigma(\underbrace{390 \text{ wartości } w_{jk}})$

$\sim N(0, 390) = N(0, \sqrt{390}^2)$

$\sigma \approx 20$

N(0,1)