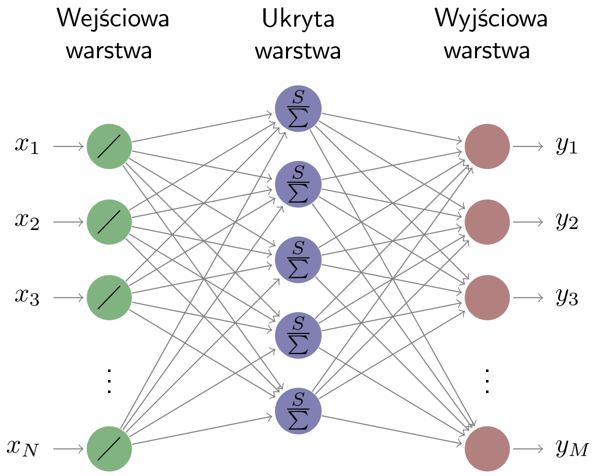
Wsteczna propagacja błędu

Informacje: Propagacja wsteczna – podstawowy algorytm uczenia nadzorowanego wielowarstwowych, jednokierunkowych sieci neuronowych. Podaje on przepis na zmianę wag dowolnych połączeń elementów przetwarzających rozmieszczonych w sąsiednich warstwach sieci. Oparty jest on na minimalizacji sumy kwadratów błędów (lub innej funkcji błędu) uczenia z wykorzystaniem optymalizacyjnej metody największego spadku. Dzięki zastosowaniu specyficznego sposobu propagowania błędów uczenia sieci powstałych na jej wyjściu, tj. przesyłania ich od warstwy wyjściowej do wejściowej, algorytm propagacji wstecznej stał się jednym z najskuteczniejszych algorytmów uczenia sieci `https://pl.wikipedia.org/wiki/Propagacja_wsteczna`.

Topologia sieci:



Funkcja błędu aproksymacji:

$$F = \sum_{p \in P} {}^{p}E \quad \text{, gdzie } p \text{ jest elementem w zbiorze trenującym } P$$

$$^{p}E = \frac{1}{2} \sum_{j \in M} \left(\hat{y}_m - y_m\right)^2 \quad \text{, gdzie } \hat{y}_m \text{ trenuje } m \text{ wyjście } y_m$$

Funkcje wag - współczynników połączeń między neuronami:

$$^{p} \triangle w_{hm} \sim -\nabla_w \cdot {}^{p}E \quad \text{, gdzie } w_{hm} \text{ jest wagą połączenia z neuronu } h \text{ do neuronu } m$$

$$\triangle w_{hm} = -\eta \frac{\partial E\left(w_{hm}\right)}{\partial w_{hm}} \quad \text{, gdzie } \eta \text{ jest współczynnikiem uczenia się}$$

Dla wyjściowych neuronów z ostatniej warstwy:

$net_m = \sum_{i=0}^{H} w_{im} \tilde{o}_i$

$o_m = y_m = f_m(net_m)$

$\frac{\partial E(w_{hm})}{\partial w_{hm}} = \frac{\partial E}{\partial net_m} \cdot \frac{\partial net_m}{\partial w_{hm}}$

$$\frac{\partial net_m}{\partial w_{hm}} = \frac{\partial}{\partial w_{hm}} \cdot net_m = \frac{\partial}{\partial w_{hm}} \sum_{i=0}^{H} w_{im} \tilde{o}_i = \sum_{i=0}^{H} \frac{\partial}{\partial w_{hm}} w_{im} \tilde{o}_i = \frac{\partial}{\partial w_{hm}} \tilde{o}_h w_{hm} = \boxed{\tilde{o}_h}$$

$$\frac{\partial E}{\partial net_m} = \frac{\partial E}{\partial y_m} \cdot \frac{\partial y_m}{\partial net_m} \ (= -\delta_m) = \frac{\partial E}{\partial y_m} \cdot \frac{\partial}{\partial net_m} f_m(net_m) = \underbrace{\frac{\partial E}{\partial y_m} \cdot \boxed{f_m'(net_m)}}_{=: -\delta_m}$$

$$\frac{\partial E}{\partial y_m} = \frac{\partial}{\partial y_m} \cdot \frac{1}{2} \sum_{j=1}^{M} \left(\hat{y}_j - y_j\right)^2 = \boxed{-(\hat{y}_m - y_m)}$$

$$\triangle w_{hm} = -\eta \frac{\partial E}{\partial w_{hm}} = -\eta \frac{\partial E}{\partial y_m} \frac{\partial y_m}{\partial net_m} \frac{\partial net_m}{\partial w_{hm}} = \eta (\hat{y}_m - y_m) f_m'(net_m) \tilde{o}_h$$

$\boxed{\delta_m = (\hat{y}_m - y_m) \cdot f'_m(net_m)}$

$\boxed{\triangle\, w_{hm} = \eta \cdot \delta_m \cdot \tilde{o}_h}$ Widrow-Hoff reguła / δ-reguła

DLa neuronów z ukrytych warstw między wejściem, a wyjściem:

$net_h = \sum_{i=0}^{H} w_{ih}\tilde{o}_i$

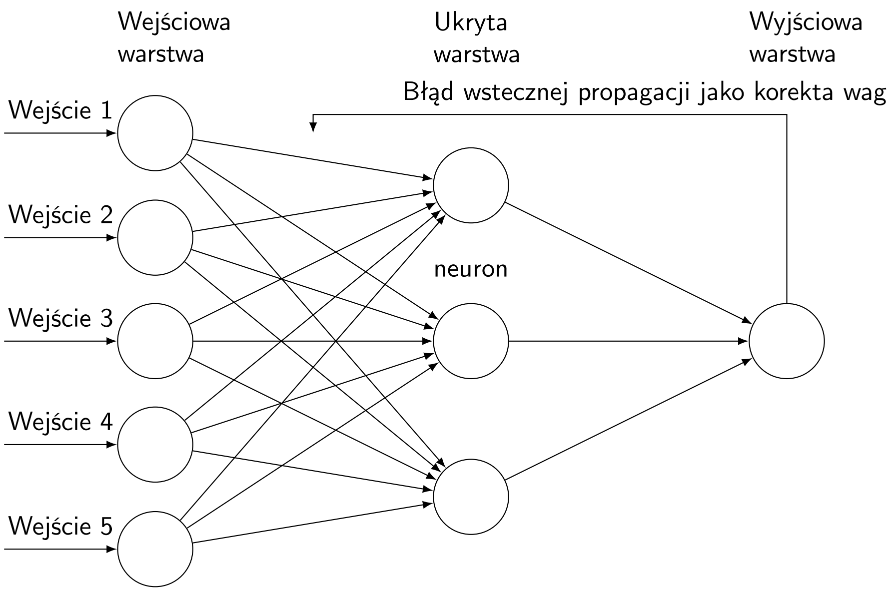
$\frac{\partial E}{\partial w_{kh}} = \frac{\partial E}{\partial net_h} \cdot \frac{\partial net_h}{\partial w_{kh}}$

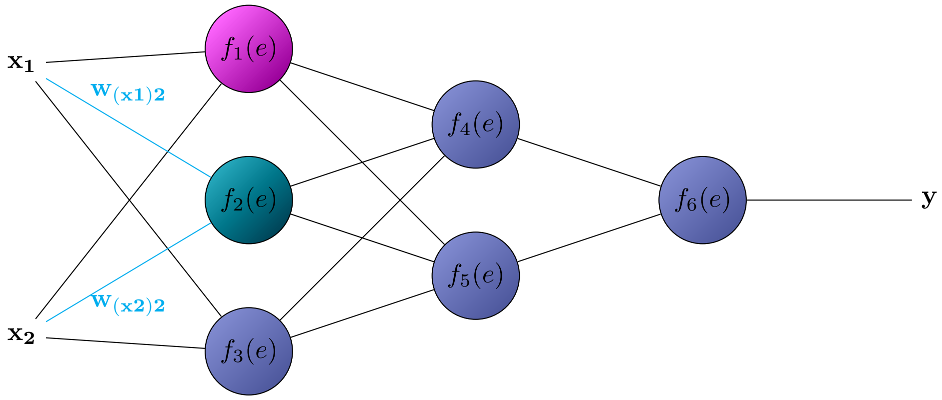
$$\delta_h = -\frac{\partial E}{\partial net_h} = -\frac{\partial E}{\partial o_h} \cdot \frac{\partial o_h}{\partial net_h}$$

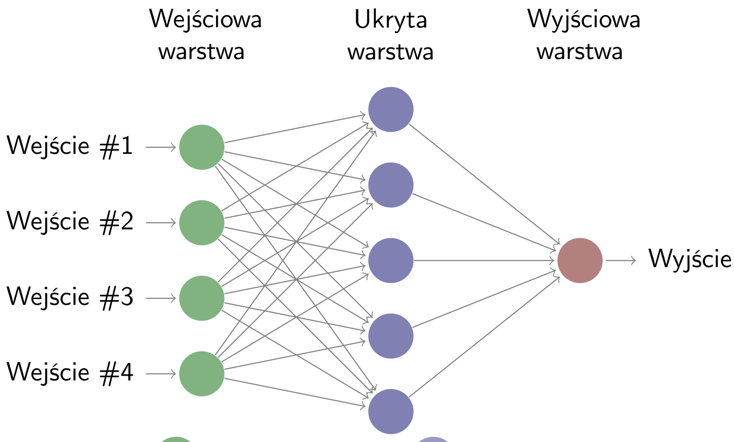
$$-\frac{\partial E}{\partial o_h} = -\frac{\partial E\left(\underline{net}_{l=1}, \underline{net}_{l=2}, \ldots, \underline{net}_{l=L}\right)}{\partial o_h} = \sum_{l=1}^{L}\left(-\frac{\partial E}{\partial \underline{net}_l}\right) \cdot \frac{\partial \underline{net}_l}{\partial o_h} = \sum_{l=1}^{L} \underline{\delta}_l \cdot \frac{\partial}{\partial o_h} \sum_{j=0}^{H} \underline{w}_{jl} \cdot o_j = \sum_{l=1}^{L} \underline{\delta}_l \cdot \underline{w}_{hl}$$

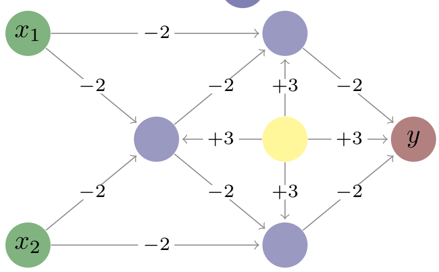
$\boxed{\delta_h = \sum_{l=1}^{L} (\underline{\delta}_l \cdot \underline{w}_{hl}) \cdot f'\,(net_h)}$

$\boxed{\triangle\, w_{kh} = \eta \cdot \delta_h \cdot \tilde{o}_k}$






Wejściowa warstwa
Ukryta warstwa
Wyjściowa warstwa
Wejście #1
Wejście #2
Wejście #3
Wejście #4
Wyjście


x_1
x_2
y
−2
+3