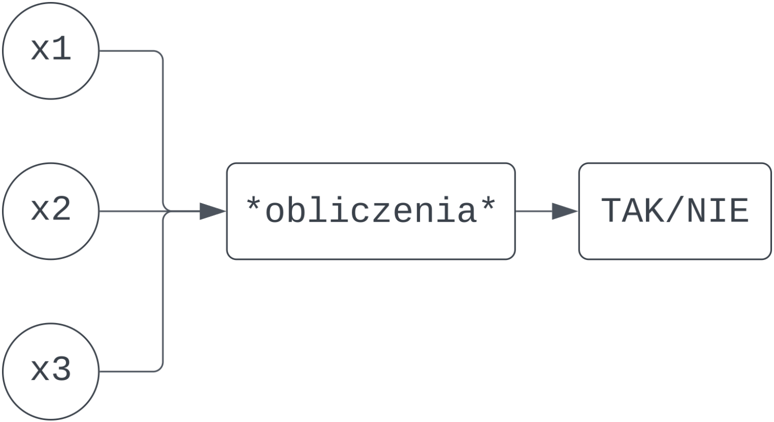
Perceptrony wielowarstwowe

Jak wyaproksymować (niemal) dowolną funkcję

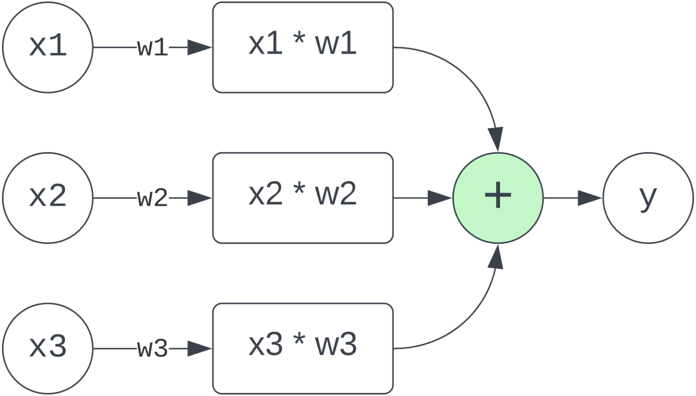
Michał Horodecki
Dominik Matuszek

Perceptron – przypomnienie

Co my w ogóle robimy



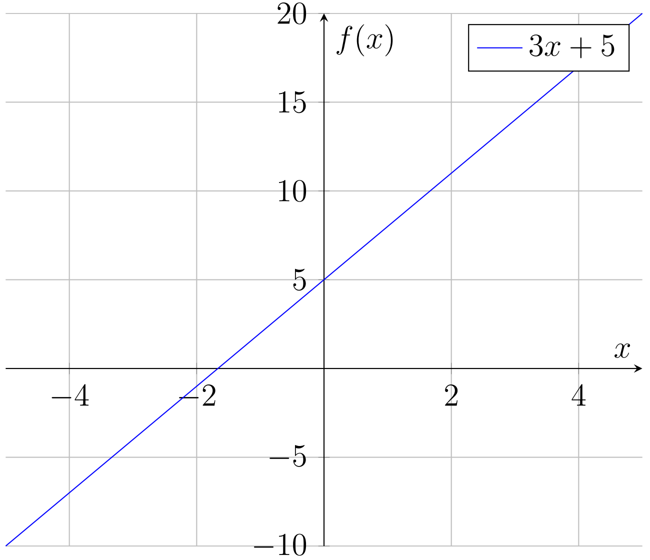
Schemat Perceptrona



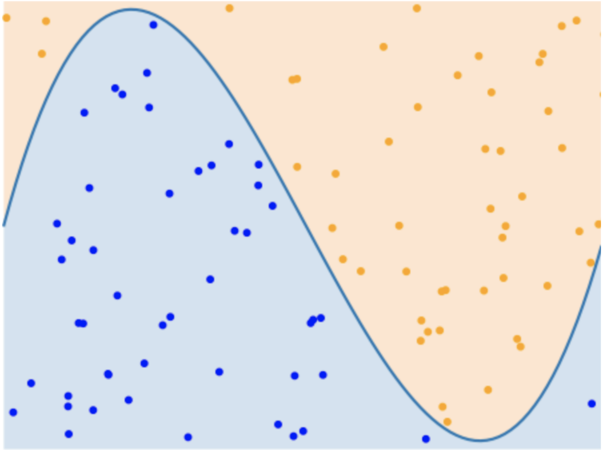
Formalniej

- $X = (x_1, x_2, x_3, \dots, x_n) \in \mathbb{R}^n$
- $W = (w_1, w_2, w_3, \dots, w_n) \in \mathbb{R}^n$; (wagi zawsze będą pionowe by default)
- $y = X \circ W$; oczywiście $y \in \mathbb{R}$

W praktyce jeszcze dodajemy tzw. *bias* do wartości y, by móc oddawać wszystkie zależności liniowe.


$f(x)$
$3x+5$
20
15
10
5
-5
-10
x
-4
-2
2
4

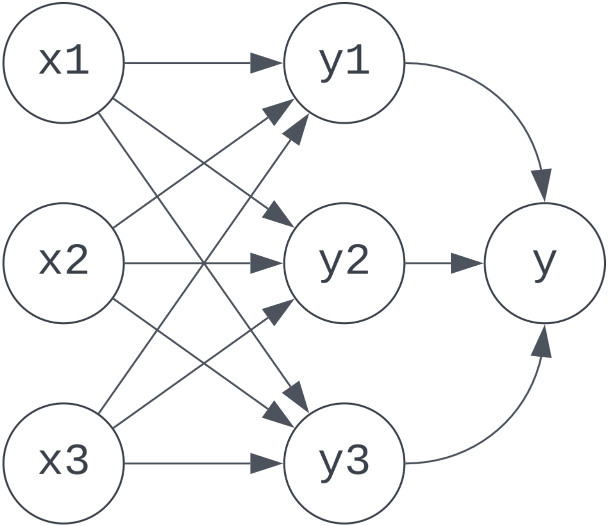
Problem z Perceptronem



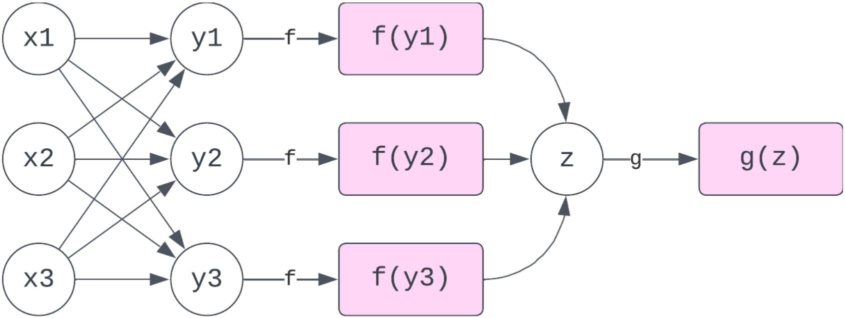
Rysunek: Perceptron nie potrafi oddać nieliniowych zależności

Multi Level Perceptron

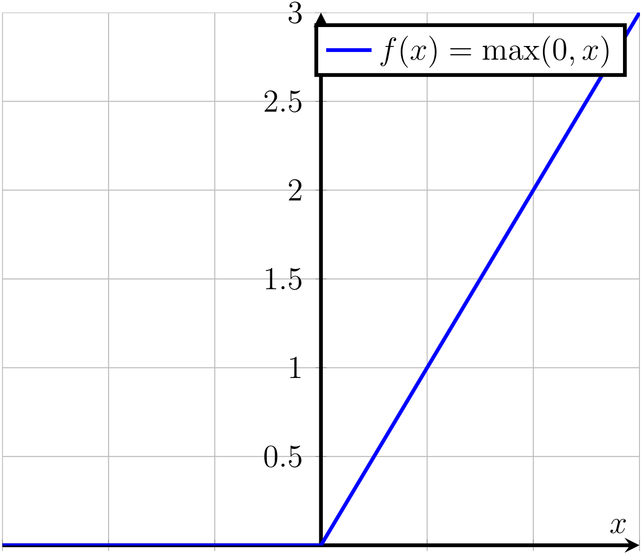
Pomysł na MLP

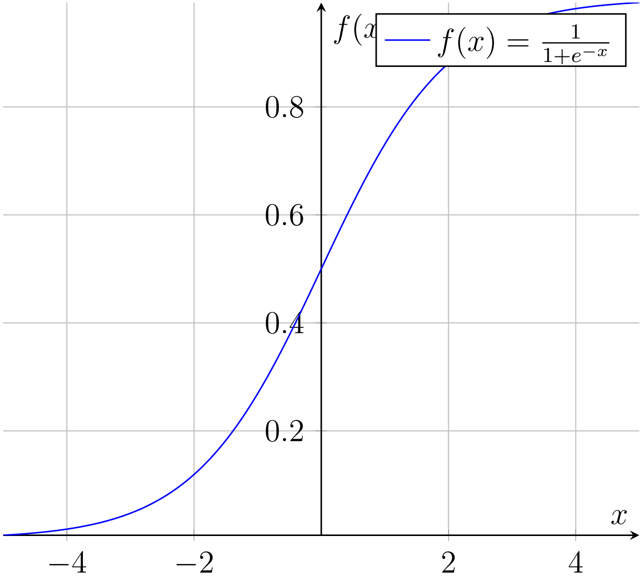


Funkcje aktywacji



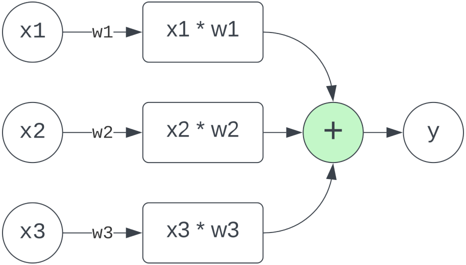
ReLU




$f(x)$
$f(x) = \frac{1}{1+e^{-x}}$
0.8
0.6
0.4
0.2
x
-4
-2
2
4

Forward Propagation

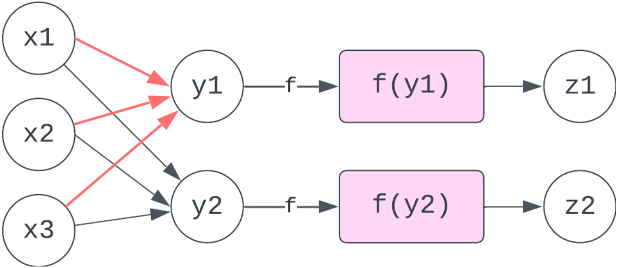
Forward Propagation w Perceptronie



$y = x \circ w$ gdzie

- x to poziomy wektor wejściowy
- w to pionowy wektor wag

Forward Propagation w MLP



Rysunek: Pojedyncza warstwa w MLP

Forward Propagation w MLP – Oznaczenia

Oznaczamy:

- $s^{(i)}$ – liczba perceptronów w i-tej warstwie.
- $x^{(i)} \in \mathbb{R}^{s}_{i-1}$ – poziomy wektor wejściowy i-tej warstwy
- macierz wag $W^{(i)} \in \mathbb{R}^{s^{(i-1)} \times s^{(i)}}$, gdzie $W^{(i)}_{j,\,k}$ – waga między x_j a y_k na styku warstw $i-1$ oraz i
- $y^{(i)} \in \mathbb{R}^{s^{(i)}}$ – poziome wartości przed aktywacją w i-tej warstwie
- $f^{(i)}$ – funkcja aktywacji i-tej warstwy
- $z^{(i)} \in \mathbb{R}^{s^{(i)}}$ – poziomy wektor wyjściowy i-tej warstwy, w „klasycznym” przypadku $x^{(i+1)} = z^{(i)}$

Forward propagation w MLP – wzory

Dla k-tego perceptrona na i-tej warstwie mamy:

$$y_k^{(i)} = \sum_{j=1}^{s_{i-1}} x_j^{(i)} \cdot W_{j,k}^{(i)}$$

Forward propagation w MLP – wzory

Dla k-tego perceptrona na i-tej warstwie mamy:

$$y_k^{(i)} = \sum_{j=1}^{s_{i-1}} x_j^{(i)} \cdot W_{j,k}^{(i)}$$

co jest mnożeniem wektora x z k-tą kolumną W

$$y_k^{(i)} = x^{(i)} \circ W_{*,k}^{(i)}$$

Forward propagation w MLP – wzory

Dla k-tego perceptrona na i-tej warstwie mamy:

$$y_k^{(i)} = \sum_{j=1}^{s_{i-1}} x_j^{(i)} \cdot W_{j,k}^{(i)}$$

co jest mnożeniem wektora x z k-tą kolumną W

$$y_k^{(i)} = x^{(i)} \circ W_{*,k}^{(i)}$$

Co można zapisać jako mnożenie macierzy:

$$y^{(i)} = x^{(i)} \cdot W^{(i)}$$

$$
\begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & \ldots & x_n^{(i)} \end{bmatrix} \cdot \begin{bmatrix} W_{1,\,1}^{(i)} & W_{1,\,2}^{(i)} & \ldots & W_{1,\,s^{(i)}}^{(i)} \\ W_{2,\,1}^{(i)} & W_{2,\,2}^{(i)} & \ldots & W_{2,\,s^{(i)}}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ W_{s^{(i-1)},\,1}^{(i)} & W_{s^{(i-1)},\,2}^{(i)} & \ldots & W_{s^{(i-1)},\,s^{(i)}}^{(i)} \end{bmatrix}
$$

$$
= \begin{bmatrix} y_1^{(i)} & y_2^{(i)} & \ldots & y_{s^{(i)}}^{(i)} \end{bmatrix}
$$

Aktywacja

$$z_k^{(i)} = f^{(i)}\left(y_k^{(i)}\right)$$

Co zapisujemy w skrócie jako

$$z^{(i)} = f^{(i)}\left(y^{(i)}\right)$$

I na koniec wyjście trafia jako wejście do kolejnej warstwy:

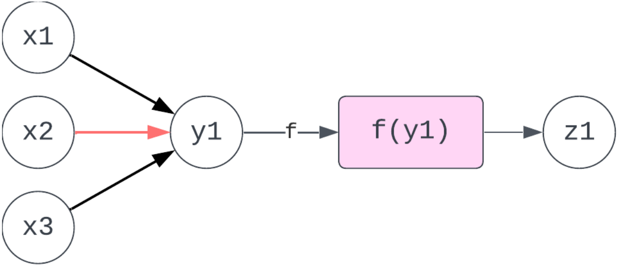
$$x^{(i+1)} = z^{(i)}$$

Backward Propagation

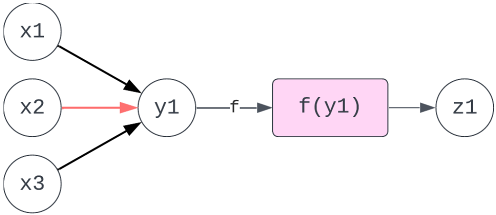
Backward Propagation – Zarys

Chcemy zastosować algorytm gradient descent.

Aby poprawić model musimy dla każdej wagi $w \in \mathbb{R}$ w sieci policzyć pochodną $\frac{\partial L}{\partial w}$ po funkcji straty L.



Rysunek: Rozważamy wagę w między x_2 a y_1



$$\begin{aligned}
\frac{\partial L}{\partial w} &= \frac{\partial y_1}{\partial w} \cdot \frac{\partial z_1}{\partial y_1} \cdot \frac{\partial L}{\partial z_1} \\
&= \frac{\partial (w \cdot x_2 + c)}{\partial w} \cdot \frac{\partial (f(y_1))}{\partial y_1} \cdot \frac{\partial L}{\partial z_1} \\
&= x_2 \cdot f'(y_1) \cdot \frac{\partial L}{\partial z_1}
\end{aligned}$$

:)

Oznaczenia:

- $dz^{(i)} = \left[\frac{\partial L}{\partial z_1^{(i)}} \cdots \frac{\partial L}{\partial z_{s^{(i)}}^{(i)}} \right]$
- $dy^{(i)} = \left[\frac{\partial L}{\partial y_1^{(i)}} \cdots \frac{\partial L}{\partial y_{s^{(i)}}^{(i)}} \right]$
- $df^{(i)} = \left[\frac{\partial z_1^{(i)}}{\partial y_1^{(i)}} \cdots \frac{\partial z_{s^{(i)}}^{(i)}}{\partial y_{s^{(i)}}^{(i)}} \right]$
- $dW^{(i)}$ t. że[1] : $dW_{j,k}^{(i)} = \frac{\partial L}{\partial W_{j,k}^{(i)}}$

[1] Rozpisanie takiej macierzy na slajdzie mogłoby być średnio praktyczne, więc definiujemy ją tak jak definiujemy

df

$df^{(i)}$ mamy za darmo dla dowolnego i bo jest to po prostu:

$$\left[\frac{\partial f^{(i)}(y_1^{(i)})}{\partial y_1^{(i)}} \ldots \frac{\partial f^{(i)}(y_{s(i)}^{(i)})}{\partial y_{s(i)}^{(i)}}\right] = \begin{bmatrix} f^{(i)\prime}(y_1^{(i)}) & \ldots & f^{(i)\prime}(y_{s^i}^{(i)}) \end{bmatrix}$$

Mozna to prosto policzyć jeśli wiemy jaki jest wzór funkcji aktywacji f.

dy

$dy^{(i)}$ to jest z kolei (dla określonego k t. że $1 \leq k \leq s^{(i)}$):

$$\begin{aligned} dy_k^{(i)} &= \frac{\partial z_k^{(i)}}{\partial y_k^{(i)}} \cdot \frac{\partial L}{\partial z_k^{(i)}} \\ &= df_k^{(i)} \cdot dz_k^{(i)} \end{aligned}$$

Jeśli przez $*$ oznaczymy mnożenie punktowe po współrzędnych to możemy skrócić ten zapis do

$$dy^{(i)} = df^{(i)} * dz^{(i)}$$

Wiedząc, że $df^{(i)}$ można zawsze prosto policzyć, jeśli będziemy w stanie policzyć $dz^{(i)}$, będziemy mieć również $dy^{(i)}$ w pakiecie.

dW

$$
\begin{aligned}
dW_{j,k}^{(i)} &= \frac{\partial L}{\partial W_{j,k}^{(i)}} \\
&= \frac{\partial y_k^{(i)}}{\partial W_{j,k}^{(i)}} \cdot \frac{\partial L}{\partial y_k^{(i)}} \\
&= \frac{\partial \left(x_j^{(i)} \cdot W_{j,k}^{(i)} + c \right)}{\partial W_{j,k}^{(i)}} \cdot \frac{\partial L}{\partial y_k^{(i)}} \\
&= x_j^{(i)} \cdot dy_k^{(i)}
\end{aligned}
$$

Z czego (po dokładnym przemyśleniu) wyjdzie

$$dW^{(i)} = \left(x^{(i)} \right)^T \cdot dy^{(i)}$$

dz

To teraz zastanówmy się ile to jest ten $dz_j^{(i)}$ $(j \in \{1, \ldots, s^{(i)}\})$.
Przypomnijmy, że $z^{(i)} = x^{(i+1)}$.

$$
\begin{aligned}
dz_j^{(i)} &= \frac{\partial L}{\partial z_j^{(i)}} \\
&= \frac{\partial L}{\partial x_j^{(i+1)}}
\end{aligned}
$$

Tutaj pojawia się pewien predykament w liczeniu pochodnej. $x_j^{(i+1)}$ jest składnikiem (po przemnożeniu przez jakąś wagę) każdego $y_k^{(i+1)}$ (dla dowolnego $k \in \{1, \ldots, s^{(i+1)}\}$.

Dowód przez machanie rękami I

Twierdzimy że:

$$\frac{\partial L}{\partial x_j^{(i+1)}} = \sum_{k=1}^{s^{(i+1)}} \left(\frac{\partial y_k^{(i+1)}}{\partial x_j^{(i+1)}} \cdot \frac{\partial L}{\partial y_k^{(i+1)}} \right)$$

Dowód przez machanie rękami I

Twierdzimy że:

$$\frac{\partial L}{\partial x_j^{(i+1)}} = \sum_{k=1}^{s^{(i+1)}} \left(\frac{\partial y_k^{(i+1)}}{\partial x_j^{(i+1)}} \cdot \frac{\partial L}{\partial y_k^{(i+1)}} \right)$$

Zauważamy teraz, że

$$\frac{\partial y_k^{(i+1)}}{\partial x_j^{(i+1)}} = \frac{\partial (W_{j,k}^{(i+1)} \cdot x_j^{(i+1)} + c)}{\partial x_j^{(i+1)}} = W_{j,k}^{(i+1)}$$

oraz

$$\frac{\partial L}{\partial y_k^{(i+1)}} = dy_k^{(i+1)}$$

Najbardziej formalny dowód

Dowód przez machanie rękami II

Dostajemy

$$\frac{\partial L}{\partial x_j^{(i+1)}} = \sum_{k=1}^{s^{(i+1)}} \left(dy_k^{(i+1)} \cdot W_{j,k}^{(i+1)} \right)$$

Dowód przez machanie rękami II

Dostajemy

$$\frac{\partial L}{\partial x_j^{(i+1)}} = \sum_{k=1}^{s^{(i+1)}} \left(dy_k^{(i+1)} \cdot W_{j,k}^{(i+1)} \right)$$

Co oczywiście się zwija do mnożenia macierzy:

$$\begin{aligned} dz^{(i)} &= \left[\frac{\partial L}{\partial x_1^{(i+1)}} \cdots \frac{\partial L}{\partial x_{s^{(i+1)}}^{(i+1)}} \right] \\ &= dy^{(i+1)} \cdot \left(W^{(i+1)} \right)^T \end{aligned}$$

Wielki Powrót Analizy Matematycznej

Twierdzenie (Pochodna funkcji wielowymiarowej)

Niech $f : \mathbb{R}^n \to \mathbb{R}^m$ *będzie funkcją różniczkowalną i niech* $f = g \circ h$*. Wtedy* $\mathbb{R}^{m \times n} \ni D_f(a) = D_g(h(a)) \cdot D_h(a)$

W naszym przypadku mamy:

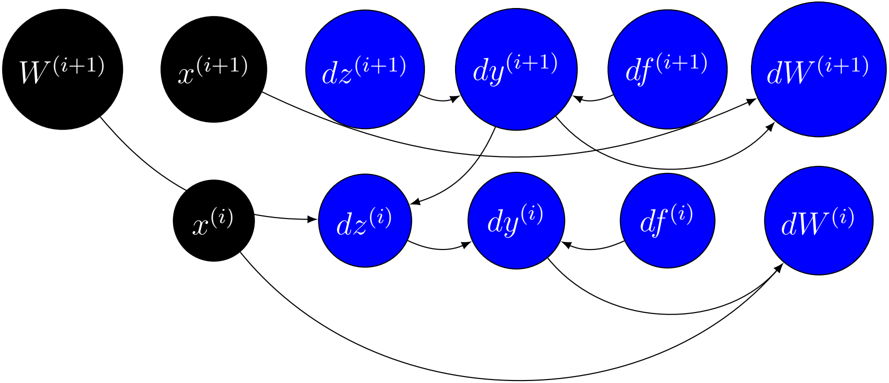
- $f\left(x^{(i+1)}\right)$ – funkcja wyrażająca stratę w zależności od $x^{(i+1)}$
- $g\left(y^{(i+1)}\right)$ – funkcja wyrażająca stratę w zależności od $y^{(i+1)}$
- $h\left(x^{(i+1)}\right) = y^{(i+1)} = x^{(i+1)} \cdot W^{(i+1)}$

$$\begin{aligned} dz^{(i)} &= D_f(x^{(i+1)}) = D_g(y^{(i+1)}) \cdot D_h(x^{(i+1)}) \\ &= dy^{(i+1)} \cdot \left(W^{(i+1)}\right)^T \end{aligned}$$

Backpropagation – podsumowanie

- $dz^{(i)} = \left[\frac{\partial L}{\partial z_1^{(i)}} \cdots \frac{\partial L}{\partial z_{s^{(i)}}^{(i)}}\right] = dy^{(i+1)} \cdot \left(W^{(i+1)}\right)^T$
- $dy^{(i)} = \left[\frac{\partial L}{\partial y_1^{(i)}} \cdots \frac{\partial L}{\partial y_{s^{(i)}}^{(i)}}\right] = df^{(i)} * dz^{(i)}$
- $df^{(i)} = \left[\frac{\partial z_1^{(i)}}{\partial y_1^{(i)}} \cdots \frac{\partial z_{s^{(i)}}^{(i)}}{\partial y_{s^{(i)}}^{(i)}}\right] = \begin{bmatrix} f^{(i)\prime}(y_1^{(i)}) & \dots & f^{(i)\prime}(y_{s^i}^{(i)}) \end{bmatrix}$
- $dW^{(i)} = \left(x^{(i)}\right)^T \cdot dy^{(i)}$

Graf zależności (prawie)



Składamy wszystko do kupy

Idziemy z obliczaniem pochodnych „od tyłu". Jeśli ostatnia warstwa naszego MLP ma indeks λ, to na początku chcemy policzyć $dW^{(\lambda)}$.
Strata obliczana jest dla wyjścia z tej warstwy $(L(z^{(\lambda)}))$, zatem „za darmo" dostajemy $dz^{(\lambda)} = \left[\frac{\partial L}{\partial z_1^{(\lambda)}} \cdots \frac{\partial L}{\partial z_{s^{(\lambda)}}^{(\lambda)}}\right]$.
Wartości numeryczne tego wektora będą zależeć od przyjętej funkcji straty.

MLP – Algorytm

Algorytm MLP

- Bierzemy wejście do naszej sieci i ustawiamy jako $x^{(0)}$
- Obliczamy wartości kolejnych warstw $y^{(i+1)} = z^{(i)} \cdot W^{(i)}$
- Obliczamy funkcję straty na wyjściu ostatniej warstwy $L(z^{(\lambda)})$
- Korzystając z definicji L ręcznie obliczamy $dz^{(\lambda)}$
- korzystając z wyprowadzonych wzorów, idąc od tyłu (tj. od λ do 0) obliczamy kolejne $dy^{(i)}, df^{(i)}, dW^{(i)}, dz^{(i)}$
- Aktulizujemy wagi $W^{(i)} := W^{(i)} - \eta \cdot dW^{(i)}$

