Zaawansowane Metody Inteligencji Obliczeniowej

Lab 7: Metoda Spadku Wzdłuż Gradientu

Michał Kempka Marek Wydmuch

22 kwietania 2021

Fundusze Europejskie Polska Cyfrowa | Rzeczpospolita Polska | Unia Europejska Europejski Fundusz Rozwoju Regionalnego

"Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)",
projekt finansowany ze środków Programu Operacyjnego Polska Cyfrowa POPC.03.02.00-00-0001/20

1 Wprowadzenie/przypomnienie

1.1 Pochodna

Pochodna funkcji — miara szybkości zmian wartości funkcji względem zmian jej argumentów. Dla funkcji rzeczywistej $y = f(x)$ ($f : \mathbb{R} \to \mathbb{R}$ – funkcja o dziedzinie $\mathbb{R}$ i przeciwdziedzinie $\mathbb{R}$), pochodna w punkcie x_0 (o ile istnieje):

$$\lim_{\Delta x \to 0} = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \,. \tag{1}$$

Pochodna oznaczana jest często jednym z poniższych oznaczeń:

$$\lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} \,, \frac{dy}{dx} \,, \frac{d}{dx} f(x) \,, f'(x) \,.$$

Jeśli pochodna funkcji $f : (a, b) \to \mathbb{R}$ istnieje w każdym punkcie przedziału (a, b), mówimy, że funkcja f jest różniczkowalna w przedziale (a, b). Różnicując funkcję f otrzymujemy pierwszą pochodną f', która może być również różniczkowalna w przedziale (a, b), różniczkując pierwszą pochodną f' otrzymujemy drugą pochodną f''.

1.2 Pochodna funkcji złożonej

$$h(x) = f(g(x))$$
$$h'(x) = (f(g(x)))' = f'(g(x)) \cdot g'(x)$$

Stosując inną notację. Jeśli zmienna z zależy od zmiennej y, która z kolei zależy od x:

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}$$

1.3 Pochodna cząstkowa

Pochodna cząstkowa — dla danej funkcji wielu zmiennych jest to pochodna względem jednej z jej zmiennych przy ustaleniu pozostałych, oznaczana często jednym z poniższych oznaczeń:

$$\frac{\partial f}{\partial x}, f'_x \,.$$

1.4 Gradient

Wektor pochodnych cząstkowych funkcji $f : \mathbb{R}^n \to \mathbb{R}$ oznaczany jako $\nabla f(\boldsymbol{x})$, który jest funkcją $\mathbb{R}^n \to \mathbb{R}^n$:

$$\nabla f(\boldsymbol{x}) = \nabla f \left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \right) = \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \frac{\partial f(\boldsymbol{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \tag{2}$$

1.5 Hesjan (Macierz Hessego)

Hesjan (ang. Hessian), Macierz Hessego – symetryczna, kwadratowa macierz drugich pochodnych funkcji $f : \mathbb{R}^n \to \mathbb{R}$ oznacza jako $\nabla^2 f(\boldsymbol{x})$ lub $H(\boldsymbol{x})$, która jest funkcją $\mathbb{R}^n \to \mathbb{R}^{n \times n}$

$$H(\boldsymbol{x}) = \nabla^2 f(\boldsymbol{x}) = \nabla^2 f \left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \right) = \begin{bmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial^2 x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial^2 x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial^2 x_n} \end{bmatrix}$$

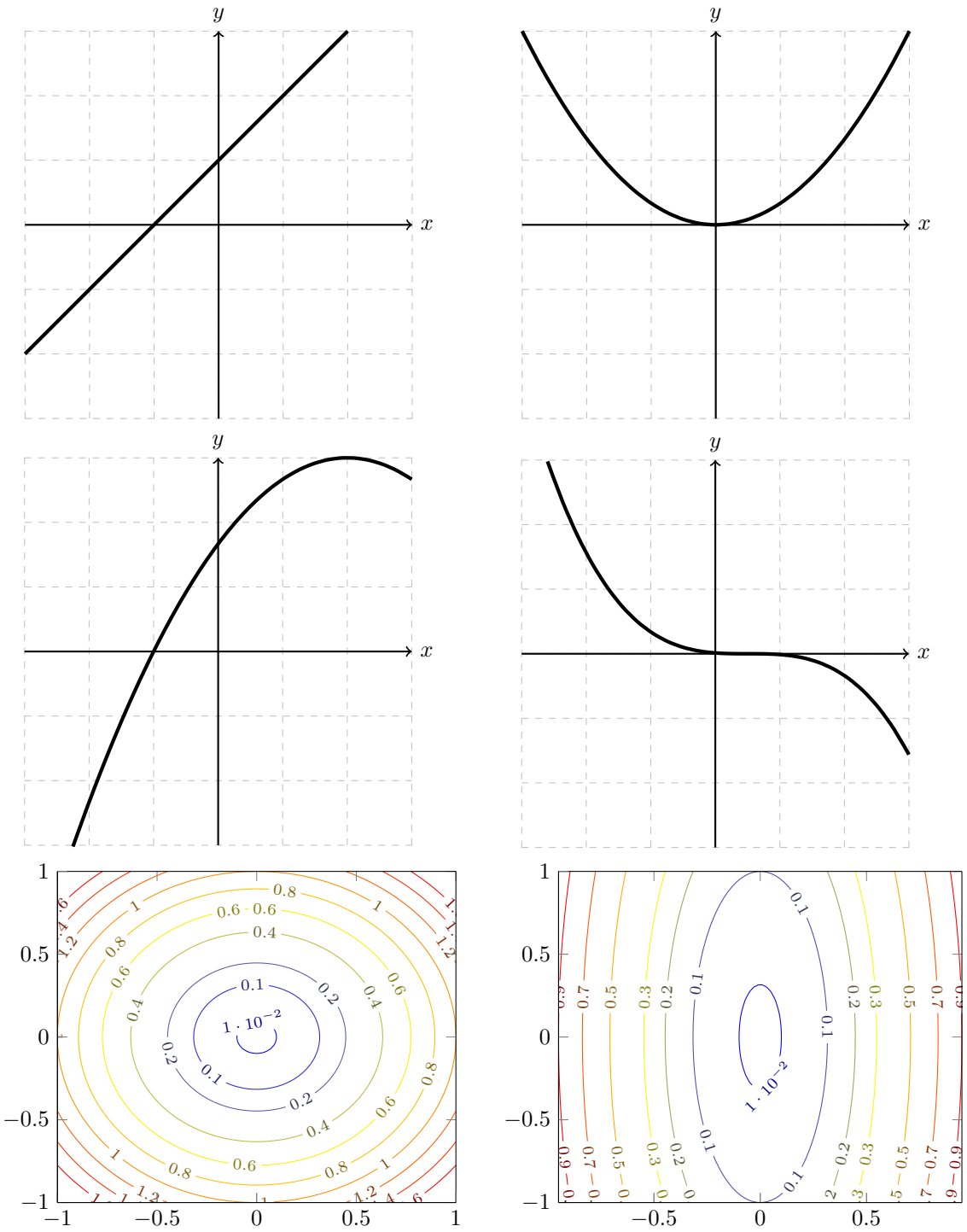
1.6 Znajdowanie minimum funkcji różniczkowalnej

Dwukrotnie różniczkowalna funkcja $f : \mathbb{R}^n \to \mathbb{R}$ ma minimum lokalne w punkcie x^* jeśli $\nabla f(x^*) = 0$, hesjan jest dodatnio określony (tj. $\forall_{x \neq 0} x^T H(x^*) x > 0$.

2 Zadania dotyczące gradientu

2.1 Rysowanie gradientów

Dla poniższych funkcji narysuj kierunki gradientów za pomocą strzałek dla różnych punktów. Postaraj się zachować zależność wielkości pomiędzy różnymi punktami.



3 Regresja Liniowa

Mamy daną macierz $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{n \times m}$, gdzie wiersz i jest wektorem $\boldsymbol{x}_i$ reprezentującym pojedynczy punkt w danych i zawierajacym m wartości określających jego cechy. Dodatkowo mamy dany wektor $\boldsymbol{y} \in \mathbb{R}^n$, w którym wartość y_i odpowiada cechą w wektorze $\boldsymbol{x}_i$. Chcielibyśmy znaleźć **model** (często nazywany hipotezą i dlatego oznaczany jako $h(\boldsymbol{x})$), która najlepiej przewiduje wartość y_i ($\forall_{i \in \{1,2 \ldots n\}} h(\boldsymbol{x}_i) \approx y_i$). Jak nazwa wskazuje w wypadku regresji liniową naszym modelem jest funkcja liniowa o współczynnikach $\boldsymbol{w}$ i wyrazie wolnym b, które w tym kontekście raczej nazywamy wagami (ang. weights) i biasem (ang. bias). Alternatywnie często bias jest omijany i zamiast tego dopakowana jest kolumna do macierzy X, zawierająca 1 dla każdego wiersza, dla uproszczenia my będziemy stosować to podejście. By sformalizować ten problem

najczęściej używany jest błąd średnio-kwadratowy (ang. mean square error (MSE)):

$$\arg\max_{\boldsymbol{w}\in\mathbb{R}^m} \mathcal{L}(\boldsymbol{w}) = \frac{1}{2n}\sum_{i=1}^{n}(\overbrace{\boldsymbol{x}_i^T\boldsymbol{w}}^{h(\boldsymbol{x}_i)} - y_i)^2 = \mathbb{E}[(X\boldsymbol{w} - y)^2] \tag{3}$$

Pytanie: Rozwiąż ten problem analitycznie (znajdzie miejsce gdzie gradient błędu jest równy 0).

Niestety, wiele problemów optymalizacji (bardziej złożone modele, inne funkcje błędu) nie posiadają (lub nie zostały jeszcze znalezione) rozwiązań analitycznych tak jak regresja liniowa z MSE. Zamiast tego stosujemy inne metody optymalizacji, jedną z najpopularniejszych dla problemów ciągłych jest Metoda Spadku Wzdłuż Gradientu.

4 Metoda Spadku Wzdłuż Gradientu

Popularną metodą na znalezienie $\arg\max_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}); f : \mathbb{R}^n \to \mathbb{R}$ jest Metoda Spadku Wzdłuż Gradientu (ang. Gradient Descent (GD)). W metodzie GD, w każdym kroku poruszamy się w kierunku wskazywanym przez gradient. Wielkość o jaką się poruszamy jest regulowana przez wartość nazywaną wielkością kroku (ang. step-size, albo learning rate) i jest oznaczana często jako α (nasze oznaczenie) albo η. W najprostszej postaci α jest stała dla każdej współrzędnej. W różnych rozszerzenich/wariantach metody GD (np. AdaGrad, Adam, Momentum) różne wartości są używane dla różnych współrzędnych.

$$x_{t+1} = x_t - \alpha\nabla f(x_t),\ \eta \in \mathbb{R}^+ \tag{4}$$

Pytanie: Załóżmy, że chcemy rozwiązać problem regresji liniowej przedstawiony w Sekcji 3 za pomocą GD. W czasie t mamy wagi $\boldsymbol{w}_t$, jak wygląda aktualizacja dla tego kroku z użyciem danej stałej α.

Algorytm 1: (Batch) Gradient Descent

Argumenty:
Zbiór danych $X, \boldsymbol{y}$ o długości n
Parametry (wagi) W
Funkcja straty (błędu) $\mathcal{L}(y_i, h(\boldsymbol{x}_i, W)) \in \mathbb{R}$
Ilość iteracji $T \in \mathbb{N}$
Zbiór malejących wartości wielkości kroku: $\{\alpha_0, \alpha_1 \dots \alpha_{T-1}\}$
for $t \leftarrow 0$ **to** $T-1$ **do**
    $g_t \leftarrow 0$
    for $i \leftarrow 0$ **to** $n-1$ **do**
        $g_t \leftarrow g_t + \frac{1}{n}\nabla\mathcal{L}(y_i, h(\boldsymbol{x}_i, W))$ (względem W)
    $\theta \leftarrow \theta - \eta_t g_t$

5 Stochastyczny spadek wzdłuż gradientu

Jedno z najczęściej stosowanych odmian metody spadku wzdłuż gradientu jest Stochastyczny Spadek Wzdłuż Gradientu (ang. Stochastic Gradient Descent (SGD)), który zamiast oblicza $\nabla\mathcal{L}$ dla wszystkich punktów w danych, w danym kroku oblicza $\nabla\mathcal{L}$ tylko dla pojedynczej pary $\boldsymbol{x}_i, y_i$. Obliczenie obserwacji na podstawie jednej obserwacji jest prostsze, szybsze, nie wymaga dostępu do całej macierzy X w momencie liczenia gradientu. Zazwyczaj proces optymalizacji podzielony jest na epoki, w jednej epoce dokonuje się jednej aktualizacji dla każdego punktu w danych.
Pytanie: Jak będzie wyglądać aktualizacja regresji liniowej przedstawiony w Sekcji 3 za pomocą SGD. W czasie t mamy wagi $\boldsymbol{w}_t$, jak wygląda aktualizacja dla tego kroku z użyciem danej stałej α.

Algorytm 2: Online Stochastic Gradient Descent

Argumenty:
Zbiór danych $X, \boldsymbol{y}$ o długości n
Parametry (wagi) W
Funkcja straty (błędu) $\mathcal{L}(y_i, h(\boldsymbol{x}_i, W)) \in \mathbb{R}$
Ilość iteracji $T \in \mathbb{N}$
Zbiór malejących wartości wielkości kroku: $\{\alpha_0, \alpha_1 \dots \alpha_{T-1}\}$
for $t \leftarrow 0$ **to** $T - 1$ **do**
    Pomieszaj kolejność wierszy w $X, \boldsymbol{y}$
    for $i \leftarrow 1$ **to** n **do**
        $g_{ti} \leftarrow \frac{1}{n} \nabla \mathcal{L}(y_i, h(\boldsymbol{x}_i, W))$ (względem W)
        $\theta \leftarrow \theta - \alpha_t g_{ti}$

Algorytm 3: Mini-batch Stochastic Gradient Descent

Argumenty:
Zbiór danych $X, \boldsymbol{y}$ o długości n
Parametry (wagi) W
Funkcja straty (błędu) $\mathcal{L}(y_i, h(\boldsymbol{x}_i, W)) \in \mathbb{R}$
Ilość iteracji $T \in \mathbb{N}$
Zbiór malejących wartości wielkości kroku: $\{\alpha_0, \alpha_1 \dots \alpha_{T-1}\}$
Wielkość mini-batcha $b \in \mathbb{N}$
for $t \leftarrow 1$ **to** T **do**
    Pomieszaj kolejność wierszy w $X, \boldsymbol{y}$
    for $b_i \leftarrow 1$ **to** $\lceil \frac{n}{b} \rceil$ **do**
        $g_{tb_i} \leftarrow 0$
        for $j \leftarrow b_i \cdot b$ **to** $\max((b_i + 1) \cdot b, n) - 1$ **do**
            $g_{tb_i} \leftarrow \frac{1}{n} \nabla \mathcal{L}(y_i, h(\boldsymbol{x}_i, W))$ (względem W)
        $\theta \leftarrow \theta - \alpha_t g_{tb_i}$

6 Sieci neuronowe i propagacja wsteczna

Przechodząc obok całej tej analogii o neuronach, sieci neuronowe to po prostu złożenie wielu funkcji, jak np:

$$
\begin{aligned}
f^1(\boldsymbol{x}) &= \sigma^1(\boldsymbol{x}W^1) \\
f^2(\boldsymbol{x}) &= \sigma^2(\boldsymbol{x}W^2) \\
&\dots \\
f^L(\boldsymbol{x}) &= \sigma^L(\boldsymbol{x}W^L) \\
h(\boldsymbol{x}) &= f^L(f^{L-1}(\dots f^2(f^1(\boldsymbol{x}))\dots))
\end{aligned}
$$

Każdą z takich funkcji nazywamy zazwyczaj warstwą sieci neuronowej, a σ funkcją **aktywacji** (ang. activation function), która jest nieliniowa.

Co za tym idzie z powodzeniem moglibyśmy policzyć dla każdej pary $\boldsymbol{x}_i, y_i$ składową gradientu $\nabla \mathcal{L}(y_i, h(\boldsymbol{x}_i))$ – pochodną cząstkową dla każdej wagi $\frac{\partial \mathcal{L}}{W^l_{jk}}$, dzięki znajomości pochodnych funkcji złożonej. Oznaczmy:

$$
\begin{aligned}
\boldsymbol{z}^l &= W^l \boldsymbol{a}^{l-1} \\
\boldsymbol{a}^l &= \sigma(\boldsymbol{z}^l)
\end{aligned}
$$

Teraz wyznaczmy pochodne cząstkowe względem W^{1-L}:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial W^1} &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^L} \cdot \frac{\partial \boldsymbol{a}^L}{\partial \boldsymbol{z}^L} \cdot \frac{\partial \boldsymbol{z}^L}{\partial \boldsymbol{a}^{L-1}} \cdot \frac{\partial \boldsymbol{a}^{L-1}}{\partial \boldsymbol{z}^{L-1}} \cdot \ldots \cdot \frac{\partial \boldsymbol{z}^3}{\partial \boldsymbol{a}^2} \cdot \frac{\partial \boldsymbol{a}^2}{\partial \boldsymbol{z}^2} \cdot \frac{\partial \boldsymbol{z}^2}{\partial \boldsymbol{a}^1} \cdot \frac{\partial \boldsymbol{a}^1}{\partial \boldsymbol{z}^1} \cdot \frac{\partial \boldsymbol{z}^1}{\partial W^1} \\
\frac{\partial \mathcal{L}}{\partial W^2} &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^L} \cdot \frac{\partial \boldsymbol{a}^L}{\partial \boldsymbol{z}^L} \cdot \frac{\partial \boldsymbol{z}^L}{\partial \boldsymbol{a}^{L-1}} \cdot \frac{\partial \boldsymbol{a}^{L-1}}{\partial \boldsymbol{z}^{L-1}} \cdot \ldots \cdot \frac{\partial \boldsymbol{z}^3}{\partial \boldsymbol{a}^2} \cdot \frac{\partial \boldsymbol{a}^2}{\partial \boldsymbol{z}^2} \cdot \frac{\partial \boldsymbol{z}^2}{\partial W^2} \\
\ldots &= \ldots \\
\frac{\partial \mathcal{L}}{\partial W^{L-2}} &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^L} \cdot \frac{\partial \boldsymbol{a}^L}{\partial \boldsymbol{z}^L} \cdot \frac{\partial \boldsymbol{z}^L}{\partial \boldsymbol{a}^{L-1}} \cdot \frac{\partial \boldsymbol{a}^{L-1}}{\partial \boldsymbol{z}^{L-1}} \cdot \frac{\partial \boldsymbol{z}^{L-1}}{\partial \boldsymbol{a}^{L-2}} \cdot \frac{\partial \boldsymbol{a}^{L-2}}{\partial \boldsymbol{z}^{L-2}} \cdot \frac{\partial \boldsymbol{a}^{L-2}}{\partial W^{L-1}} \\
\frac{\partial \mathcal{L}}{\partial W^{L-1}} &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^L} \cdot \frac{\partial \boldsymbol{a}^L}{\partial \boldsymbol{z}^L} \cdot \frac{\partial \boldsymbol{z}^L}{\partial \boldsymbol{a}^{L-1}} \cdot \frac{\partial \boldsymbol{a}^{L-1}}{\partial \boldsymbol{z}^{L-1}} \cdot \frac{\partial \boldsymbol{z}^{L-1}}{\partial W^{L-1}} \\
\frac{\partial \mathcal{L}}{\partial W^{L}} &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^L} \cdot \frac{\partial \boldsymbol{a}^L}{\partial \boldsymbol{z}^L} \cdot \frac{\partial \boldsymbol{z}^L}{\partial W^L}
\end{aligned}
$$

Liczenie gradientu bezpośrednio w taki sposób jest nieefektywne, zauważmy, że do wyliczenia pochodnej cząstkowej dla wag w każdej warstwie po drodze używamy tych samych pochodnych cząstkowych, możemy więc wyliczyć pochodną cząstkową $\mathcal{L}$ względem wag w kolejnych warstwach w sposób rekurencyjny:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial W^l} &= \overbrace{\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^L} \cdot \frac{\partial \boldsymbol{a}^L}{\partial \boldsymbol{z}^L} \cdot \frac{\partial \boldsymbol{z}^L}{\partial \boldsymbol{a}^{L-1}} \cdot \frac{\partial \boldsymbol{a}^{L-1}}{\partial \boldsymbol{z}^{L-1}} \cdot \ldots \cdot \frac{\partial \boldsymbol{z}^{l+1}}{\partial \boldsymbol{a}^l}}^{\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^l}} \cdot \frac{\partial \boldsymbol{a}^l}{\partial \boldsymbol{z}^l} \cdot \frac{\partial \boldsymbol{z}^l}{\partial W^l} \\
\frac{\partial \mathcal{L}}{\partial W^l} &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^l} \cdot \frac{\partial \boldsymbol{a}^l}{\partial \boldsymbol{z}^l} \cdot \frac{\partial \boldsymbol{z}^l}{\partial W^l},
\end{aligned}
$$

gdzie:

$$
\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^l} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^{l+1}} \cdot \frac{\partial \boldsymbol{a}^{l+1}}{\partial \boldsymbol{z}^{l+1}} \cdot \frac{\partial \boldsymbol{z}^{l+1}}{\partial \boldsymbol{a}^l}
$$

Taki sposób liczenia gradientów w sieciach neuronowych nazywamy propagacją wsteczną (ang. backpropagation).

Literatura

[1] Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall, third edition.

[2] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. The MIT Press, second edition.

Fundusze Europejskie Polska Cyfrowa
Rzeczpospolita Polska
Unia Europejska Europejski Fundusz Rozwoju Regionalnego

"Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)",
projekt finansowany ze środków Programu Operacyjnego Polska Cyfrowa POPC.03.02.00-00-0001/20