

# Uczenie głębokie — zarys

Bartosz Hanc

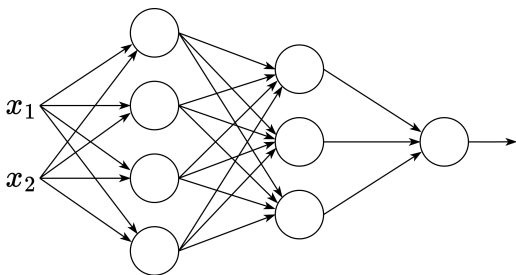
6 lutego 2025

## 1 Wprowadzenie

Historia sztucznych sieci neuronowych sięga lat 50. XX wieku, kiedy to Frank Rosenblatt zaproponował model perceptronu – prostej 3-warstwowej sieci, której zadaniem miało być rozpoznawanie znaków alfanumerycznych. Podstawową jednostką przetwarzającą w takiej sieci był neuron McCullocha–Pittsa, który realizował następujące przekształcenie  $F : \mathbb{R}^n \mapsto \mathbb{R}$

$$F(\mathbf{x}) = \varphi(\mathbf{w}^\top \mathbf{x} + b),$$

gdzie  $\mathbf{x} = (x_1, \dots, x_n)$  określa sygnały wchodzące do neuronu,  $\varphi$  jest pewną nieliniową funkcję zwaną funkcją aktywacji, a  $\mathbf{w}$  i  $b$  to pewne parametry (zwane również wagami) określające, które z sygnałów mają być wzmocnione, a które osłabione. Perceptron jest zbudowany z wielu takich neuronów połączonych ze sobą. Neurony te są zorganizowane w warstwy, w obrębie których nie ma połączeń, natomiast między sąsiednimi warstwami neurony są połączone „każdy z każdym”. Na Rysunku 1 przedstawiono schemat takiego prostego perceptronu wielowarstwowego. Strzałki pokazują przepływ sygnałów w sieci, natomiast każdy wierzchołek reprezentuje odwzorowanie nieliniowe  $F$  z właściwymi dla siebie wagami. Efektywnie perceptron wielowarstwowy realizuje więc pewne przekształcenie nieliniowe  $\mathbb{R}^n \mapsto \mathbb{R}^m$ .



Rysunek 1: Schemat prostego perceptronu wielowarstwowego realizującego odwzorowanie  $\mathbb{R}^2 \mapsto \mathbb{R}$

Podany powyżej opis konstrukcji sztucznych sieci neuronowych, choć często przywoływany we wprowadzeniach, nie jest zbyt użyteczny w nowoczesnym

podejściu do uczenia głębokiego. Bardziej przydatnym sformułowaniem, czym właściwie są sieci neuronowe jest następująca definicja: sieć neuronowa to dowolny skierowany graf acykliczny, którego wierzchołki reprezentują dowolną sparametryzowaną funkcję o argumentach i wartościach tensorowych, a skierowane krawędzie reprezentują przepływ tych tensorów. Sieć neuronowa realizuje więc pewne przekształcenie wejściowych tensorów (przykładowo może to być 4-wymiarowy tensor reprezentujący zbiór kolorowych obrazków). Trening sieci neuronowej będzie polegał na takiej zmianie jej parametrów, aby sieć realizowała określone zadanie.

Pojęcie tensora w uczeniu maszynowym nie ma wiele wspólnego z tensorami używanymi w algebrze, czy w fizyce. W naszym kontekście tensor to po prostu wielowymiarowa tablica liczb. Standardowo w tym tekście tensory będziemy oznaczać literami wytłuszczczonymi, np. **A**, natomiast odnosząc się do ustalonego elementu tensora **A** będziemy zapisywać  $A_{\alpha_1, \dots, \alpha_n}$  lub  $A_\alpha$ , gdzie  $\alpha$  będzie oznaczać wielowskaźnik  $(\alpha_1, \dots, \alpha_n)$ .

Mamy więc graf skierowany posiadający  $N$  wierzchołków. Dla każdego wierzchołka  $i \in [N]$  wprowadzimy następujące wielkości:

- $\mathcal{N}_i$  – zbiór następników, tj. wierzchołków  $j \in [N]$ , dla których istnieje krawędź skierowana  $(i, j)$ ,
- $\mathcal{P}_i$  – zbiór poprzedników, tj. wierzchołków  $j \in [N]$ , dla których istnieje krawędź skierowana  $(j, i)$ ,
- $\mathbf{F}^{(i)}(\cdot; \boldsymbol{\theta}^{(i)})$  – funkcję tensorową realizowaną przez dany wierzchołek, sparametryzowaną przez  $\boldsymbol{\theta}^{(i)}$ ,
- $\mathbf{v}^{(i)}$  – wartość funkcji  $\mathbf{F}^{(i)}$  przechowywaną aktualnie w wierzchołku (w szczególności może to być tzw. null –  $\emptyset$ ).

Obliczenie wartości  $\mathbf{v}^{(i)}$  możemy wyrazić rekurencyjnie jako

$$\mathbf{v}^{(i)} = \mathbf{F}^{(i)} \left[ \left( \mathbf{v}^{(j)} \right)_{j \in \mathcal{P}_i}; \boldsymbol{\theta}^{(i)} \right].$$

Jeśli teraz wartości  $v$  będziemy obliczać w kolejności zwróconej przez sortowanie topologiczne wierzchołków grafu to przy propagacji sygnału w przód odwiedzimy każdy wierzchołek jedynie raz.

## 2 Funkcje straty

## 3 Metody optymalizacji

## 4 Wsteczna propagacja błędu

## 5 Inicjalizacja

## 6 Regularyzacja

## 7 Architektury

### 7.1 CNN

### 7.2 RBM

### 7.3 GAN

### 7.4 DDPM

### 7.5 Autoencoder

### 7.6 Transformer

## 8 Interpretowalność