

# WSI 25Z Ćwiczenie 5: Sztuczne sieci neuronowe

Yan Korzun

4 lutego 2026

## 1 Cel ćwiczenia

Celem ćwiczenia była implementacja perceptronu wielowarstwowego (MLP) oraz algorytmu optymalizacji gradientowej z wykorzystaniem metody propagacji wstecznej. Następnie zaimplementowany model został wykorzystany do klasyfikacji jakości wina na podstawie jego parametrów fizykochemicznych. Dane wejściowe pochodziły z pliku `data.csv`, a kolumną przewidywaną była `quality`.

## 2 Opis problemu

Rozważany problem jest zadaniem uczenia nadzorowanego polegającym na klasyfikacji wieloklasowej. Każda próbka opisuje jedno wino za pomocą zestawu cech numerycznych, natomiast celem jest przypisanie go do odpowiedniej klasy jakości. Celem uczenia sieci neuronowej jest znalezienie takiego odwzorowania pomiędzy przestrzenią cech a etykietami klas, które umożliwi poprawne uogólnienie na dane niewidziane w trakcie uczenia.

## 3 Architektura perceptronu wielowarstwowego

Zaimplementowany perceptron wielowarstwowy składa się z warstwy wejściowej, dwóch warstw ukrytych oraz warstwy wyjściowej. Warstwa wejściowa przyjmuje wektor cech wina. W warstwach ukrytych zastosowano funkcję aktywacji ReLU, natomiast w warstwie wyjściowej funkcję softmax, umożliwiającą interpretację wyników jako rozkładu prawdopodobieństwa klas.

Liczba neuronów w poszczególnych warstwach była parametryzowalna i w przeprowadzonych eksperymentach wynosiła odpowiednio 32 i 16 neuronów w warstwach ukrytych.

## 4 Proces uczenia

Uczenie sieci neuronowej odbywało się z wykorzystaniem algorytmu propagacji wstecznej błędu oraz klasycznego spadku gradientu. Jako funkcję straty zastosowano entropię krzyżową, która jest standardowym wyborem dla problemów klasyfikacji wieloklasowej. Minimalizacja funkcji straty polegała na iteracyjnej aktualizacji wag i biasów w kierunku przeciwnym do gradientu.

Uczenie przeprowadzono przez ustaloną liczbę epok, przy czym w każdej epoce wykonywane było pełne przejście przez zbiór treningowy.

## 5 Przygotowanie danych

Dane zostały wczytane z pliku `data.csv`. Zmienną docelową była kolumna `quality`, natomiast pozostałe kolumny stanowiły cechy wejściowe sieci. Zbiór danych został podzielony na część treningową, walidacyjną oraz testową w proporcjach odpowiednio 70%, 15% oraz 15%. Podział ten umożliwił ocenę zdolności uogólniania modelu.

## 6 Wyniki eksperymentów

Po zakończeniu procesu uczenia dokonano ewaluacji modelu na zbiorze testowym. Jako miarę jakości klasyfikacji wykorzystano dokładność, rozumianą jako stosunek liczby poprawnych predykcji do liczby wszystkich próbek testowych. Dla każdej liczby epok testy były przeprowadzone 5 razy i końcowa dokładność to średnia arytmetyczna, testy były przeprowadzone dla wartości  $\text{learning rate} = 0.02$  (dobrana ręcznie) oraz dla kilku zestawów liczb neuronów w warstwach ukrytych.

Tabela 1: Wpływ liczby epok na dokładność klasyfikacji, 32 i 16 neuronów

Liczba epok	Dokładność
100	0.409
200	0.420
400	0.510
800	0.579
1600	0.599
3200	0.601
6400	0.603
12800	0.550

Tabela 2: Wpływ liczby epok na dokładność klasyfikacji, 16 i 32 neuronów

Liczba epok	Dokładność
100	0.411
200	0.471
400	0.536
800	0.570
1600	0.592
3200	0.603
6400	0.601
12800	0.573

Tabela 3: Wpływ liczby epok na dokładność klasyfikacji, 96 i 48 neuronów

Liczba epok	Dokładność
100	0.516
200	0.541
400	0.563
800	0.586
1600	0.596
3200	0.599
6400	0.567
12800	0.548

Tabela 4: Wpływ liczby epok na dokładność klasyfikacji, 16 i 8 neuronów

Liczba epok	Dokładność
100	0.433
200	0.417
400	0.443
800	0.496
1600	0.578
3200	0.601
6400	0.621
12800	0.575

## 7 Analiza wyników

Uzyskane wyniki potwierdzają, że zastosowanie nieliniowych funkcji aktywacji w warstwach ukrytych umożliwiło modelowi uchwycenie złożonych zależności pomiędzy cechami wejściowymi a klasami wyjściowymi. Z danych w tabelach wynika, że warto ciągle sprawdzać dokładność modelu na zbiorze testowym, ponieważ po niektórej liczbie próbek następuje przeuczenie i dokładność się pogarsza. W przypadku 16 i 32 oraz 32 i 16 neuronów różnica jest mała, ale dla innych zbiorów mogła być znacznie większa. Proporcjonalna zmiana liczby neuronów dała niejednoznaczny efekt. Dla większej liczby neuronów model daje lepsze wyniki po małej liczbie epok, ale uczenie jest wolniejsze oraz przeuczenie następuje wcześniej. Natomiast dla mniejszej liczby neuronów została uzyskana najlepsza dokładność (0.621). Warto zauważyć że otrzymana dokładność nie jest bliska 100%, dlatego ten model nie może być używany do każdego zadania, gdzie np. trzeba ocenić 1 wino, ale dla kilku tysięcy beczek może być efektywnym rozwiązaniem.

## 8 Wnioski

W ramach ćwiczenia zaimplementowano perceptron wielowarstwowy uczony metodą propagacji wstecznej z wykorzystaniem algorytmu spadku gradientu. Przeprowadzone eksperymenty pokazały, że model ten może być skutecznie wykorzystany do klasyfikacji jakości wina na podstawie jego parametrów. Odpowiedni dobór architektury sieci oraz parametrów uczenia ma kluczowe znaczenie dla jakości uzyskiwanych wyników.