

# Ćwiczenie WP

## Wielowarstwowy perceptron

### Część teoretyczna

Wykład na temat sieci neuronowych w zadaniach regresji i klasyfikacji.

### Zadania pomocnicze

Zapoznaj się z modelem MLPRegressor zaimplementowanym w bibliotece Scikit-learn lub alternatywnymi rozwiązaniami w bibliotekach PyTorch i TensorFlow + Keras.

### Zadania do wykonania

Beton jest najważniejszym materiałem w inżynierii lądowej i budownictwie. Wytrzymałość betonu na ściskanie jest funkcją jego wieku oraz zawartości składników: cementu, żużla wielkopieczowego, popiołu lotnego, wody, superplastyfikatora, kruszywa grubego i drobnego.

Zadanie polega na aproksymacji zależności pomiędzy wytrzymałością betonu na ściskanie a jego składem i wiekiem.

Zbiór danych: [https://www.kaggle.com/datasets/maajdl/yeh-concret-data?select=Concrete\\_Data\\_Yeh.csv](https://www.kaggle.com/datasets/maajdl/yeh-concret-data?select=Concrete_Data_Yeh.csv)

1. Zaimportuj niezbędne moduły.
2. Wczytaj dane, dokonaj selekcji danych – każdy student tworzy inny zbiór danych, który składa się z 1000 wylosowanych wierszy danych oryginalnych (skorzystaj z kodu zamieszczonego w ćwiczeniu dot. drzew decyzyjnych).
3. Sprawdź czy w danych są zduplikowane wiersze i brakujące wartości. Jeśli tak, usuń te wiersze (`data = data.drop_duplicates()`).
4. Scharakteryzuj dane, (przydatne funkcje: `data.describe()`, `data.info()`), zobrazuj powiązania pomiędzy zmiennymi i ich rozkłady (`sns.pairplot()`) oraz korelacje pomiędzy zmiennymi.
5. Przygotuj dane do uczenia modelu. Utwórz tabele:
  - `x` z przykładami (zawierającą wszystkie kolumny z wyjątkiem `'csMPa'`) i
  - `y` z etykietami (kolumna `'csMPa'`).Podziel dane na część treningową (80% danych) i testową (20%).  
Dokonaj standaryzacji atrybutów w zbiorach treningowym i testowym.
6. Przeprowadź próbne uczenie modelu. Skorzystaj z implementacji sieci neuronowej w Scikit-learn, PyTorch lub TensorFlow + Keras (patrz ostatnia część wykładu). Pokaż krzywą zbieżności (`loss_curve`). Pokaż następujące metryki/wykresy dla danych testowych: MSE, histogram błędów, wykres *Actual value vs. Predicted value*.
7. Zoptymalizuj hiperparametry w procedurze *Grid Search*. Zobrazuj wyniki optymalnego modelu podobnie jak w p. 6.
8. Wybierz jeden z atrybutów, który Twoim zdaniem pozwoli najlepiej przewidzieć wartość zmiennej wyjściowej. Utwórz model sieci neuronowej aproksymującej zmienną wyjściową na podstawie tego atrybutu. Porównaj wyniki p. 8 i 7.

## Co powinno znaleźć się w sprawozdaniu

- A) Cel ćwiczenia.
- B) Treść zadania.
- C) Opis używanej w ćwiczeniu sieci neuronowej (nie kopiuj treści wykładu, poszukaj w literaturze i Internecie).
- D) Metodyka rozwiązania – poszczególne instrukcje z wynikami i komentarzem (zachowaj numerację zadań).
- E) Wnioski końcowe.

## Zadania dodatkowe dla ambitnych

1. Zbadaj wrażliwość sieci neuronowej na poszczególne hiperparametry.
2. Rozwiąż postawiony problem za pomocą modeli opartych na drzewach decyzyjnych. Porównaj wyniki z wynikami sieci neuronowych.
3. Wykonaj to ćwiczenie w innym środowisku, np. R, C#, Matlab, ...

## Przykładowe zagadnienia i pytania zaliczeniowe

1. Cel i plan ćwiczenia.
2. Materiał ze sprawozdania.
3. Sieci neuronowe do regresji i klasyfikacji.
4. Uczenie sieci neuronowych, wsteczna propagacja błędów.
5. Hiperparametry sieci neuronowej i ich optymalizacja.
6. Problem przeuczenia sieci i metody zapobiegania.

## Do przygotowania na następne zajęcia

1. Zapoznać się z instrukcją do kolejnego ćwiczenia.
2. Zapoznać się z częścią teoretyczną do kolejnego ćwiczenia.
3. Wykonać zadania pomocnicze do kolejnego ćwiczenia.