

# Algorytmy genetyczne i Sztuczne sieci neuronowe

## Lista 7. - Uczenie jednowarstwowej sieci neuronowej

W ramach niniejszej listy zadań należy przeprowadzić szczegółową analizę procesu uczenia jednowarstwowej sieci neuronowej uczącej się operatora XOR. Wykonane zadania pozwolą na ocenę jakości modelu, monitorowanie wyników podczas procesu uczenia oraz przedstawienie ich w formie raportu z odpowiednimi wizualizacjami.

### Zadania do wykonania:

1. **Zaimplementować propagację w przód dla danych treningowych po zakończeniu uczenia:**
  - Obliczyć przewidywane wyniki ( $\hat{y}$ ) dla każdego przykładu w zbiorze treningowym, używając finalnych wag i biasów.
  - Porównać uzyskane wyniki z oczekiwanymi wartościami ( $y$ ).
2. **Obliczyć jakość nauczonego modelu:**
  - Wyznaczyć wskaźnik dokładności klasyfikacji (ang. *accuracy*) jako procent poprawnych przewidywań w zbiorze treningowym.
  - Obliczyć błąd średniokwadratowy (ang. *mean squared error*, *MSE*) dla finalnego modelu.
3. **Monitorować zmiany błędu podczas uczenia:**
  - Zaimplementować mechanizm zapisujący wartość MSE w każdej epoce procesu uczenia.
  - Przechowywać wartości MSE w strukturze danych umożliwiającej późniejszą analizę (np. lista lub tablica NumPy).
4. **Przygotować wizualizacje wyników:**
  - Utworzyć wykres zmian MSE w trakcie epok:
    - Oś X: numer epoki.
    - Oś Y: wartość MSE.
  - Opcjonalnie przygotować inne wykresy, takie jak:
    - Porównanie oczekiwanych i uzyskanych wyników w postaci wykresu słupkowego.
    - Wizualizacja struktury sieci z finalnymi wagami i biasami.

**5. Sformułować wnioski na podstawie wyników:**

- Opisać, w jaki sposób zmieniał się błąd w trakcie uczenia i jak szybko model osiągnął zadowalającą jakość.
- Określić, czy finalny model poprawnie nauczył się operatora XOR dla całego zbioru danych treningowych.
- Wskazać ewentualne problemy napotkane w procesie uczenia i sposoby ich rozwiązania.
- Eksperymentalnie zmienić wartości współczynnika uczenia (ang. *learning rate*) i sprawdzić, jak wpływa to na proces uczenia oraz finalne wyniki.
- Przeanalizować wpływ liczby epok na jakość nauczonego modelu.
- Wyznaczyć średni czas jednej epoki uczenia.

**6. Przygotować raport z wyników:**

- Raport powinien zawierać:
  - Opis zaimplementowanego modelu i procesu uczenia.
  - Uzyskane wyniki (accuracy, MSE) oraz ich interpretację.
  - Wykresy przedstawiające zmiany błędu oraz inne wizualizacje.
  - Wnioski i refleksje na temat skuteczności algorytmu oraz ewentualne rekomendacje dalszych usprawnień.

**7. Warunki opracowywanych rozwiązań:**

- Wszystkie wykresy powinny być czytelne i estetycznie wykonane, z odpowiednimi opisami osi i legendą, jeśli jest to konieczne.
- Opracowane rozwiązania powinny być implementowane w języku Python. Dopuszczalne jest wykorzystanie innego języka programowania pod warunkiem uzyskania zgody prowadzącego.
- Podczas implementacji wolno posługiwać się bibliotekami do obliczeń numerycznych ogólnego przeznaczenia (np. NumPy, SciPy) oraz przetwarzania danych (np. Pandas).
- Podczas implementacji nie wolno posługiwać się dedykowanymi bibliotekami do tworzenia algorytmów genetycznych (np. PyGAD, DEAP itp.) oraz architektur sieci neuronowych (np. scikit-learn, PyTorch, Tensorflow, JAX itp.).