

WSI Sprawozdanie

Laboratorium 5

Sieci neuronowe

Filip Misztal 310276

Stanisław Moska 310284

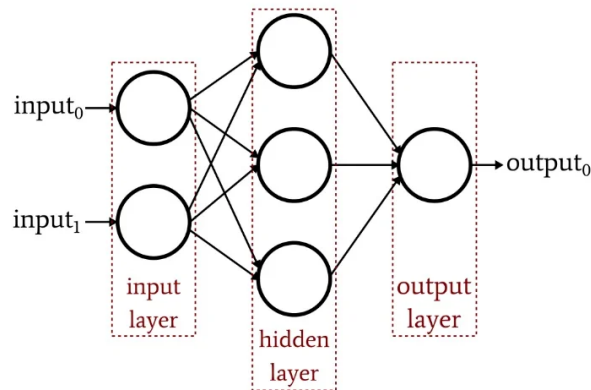
1 marca 2024

1 Wprowadzenie

Perceptron wielowarstwowy, znany również jako sieć neuronowa wielowarstwowa (MLP - Multilayer Perceptron), to rodzaj sztucznej sieci neuronowej, który składa się z co najmniej trzech warstw: warstwy wejściowej, jednej lub więcej warstw ukrytych oraz warstwy wyjściowej. Jest to rozwinięcie koncepcji perceptronu, który jest jednostką podstawową w neuronach sztucznych sieciach neuronowych.

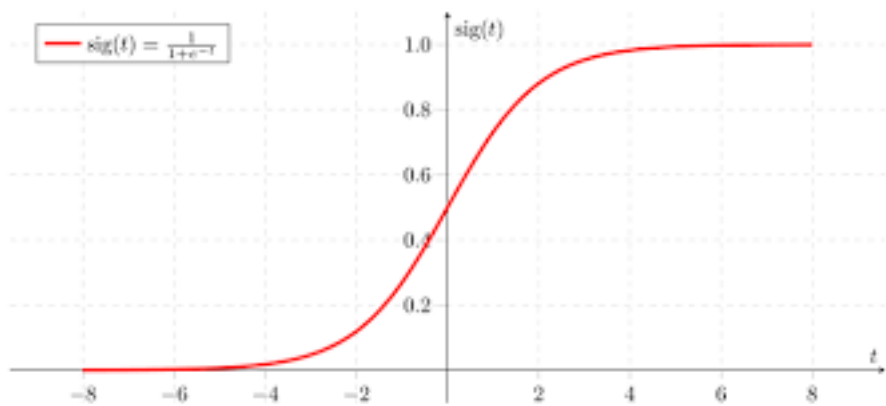
Wprowadzenie warstw ukrytych do perceptronu wielowarstwowego jest kluczowym elementem umożliwiającym modelowi uczenie się reprezentacji hierarchicznych cech danych. Proces ten obejmuje przekształcanie danych wejściowych przez warstwy ukryte, które uczą się hierarchicznych abstrakcji i reprezentacji, zanim ostatecznie trafiają do warstwy wyjściowej, gdzie generowane są końcowe predykcje.

Kluczowym elementem treningu perceptronu wielowarstwowego jest algorytm wstecznej propagacji błędów (backpropagation), który umożliwia dostosowywanie wag połączeń pomiędzy neuronami na podstawie błędów predykcji. To iteracyjne uczenie pozwala na stopniową poprawę zdolności modelu do generalizacji na nowe dane.

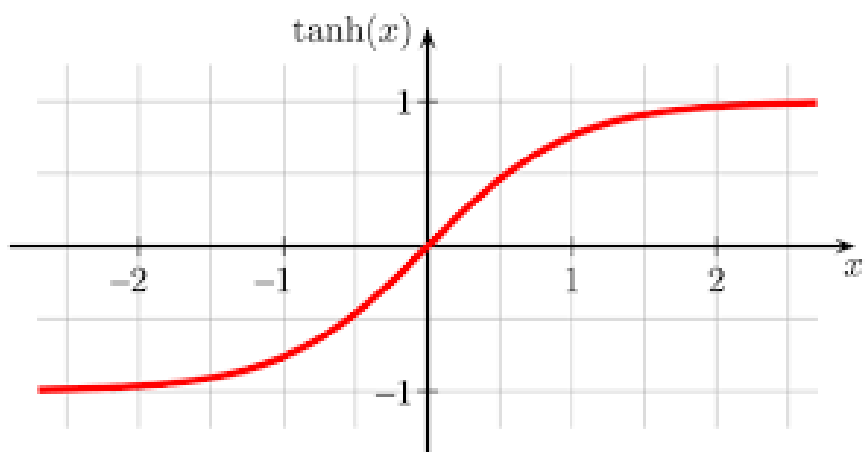


Rysunek 1: Model perceptronu wielowarstwowego

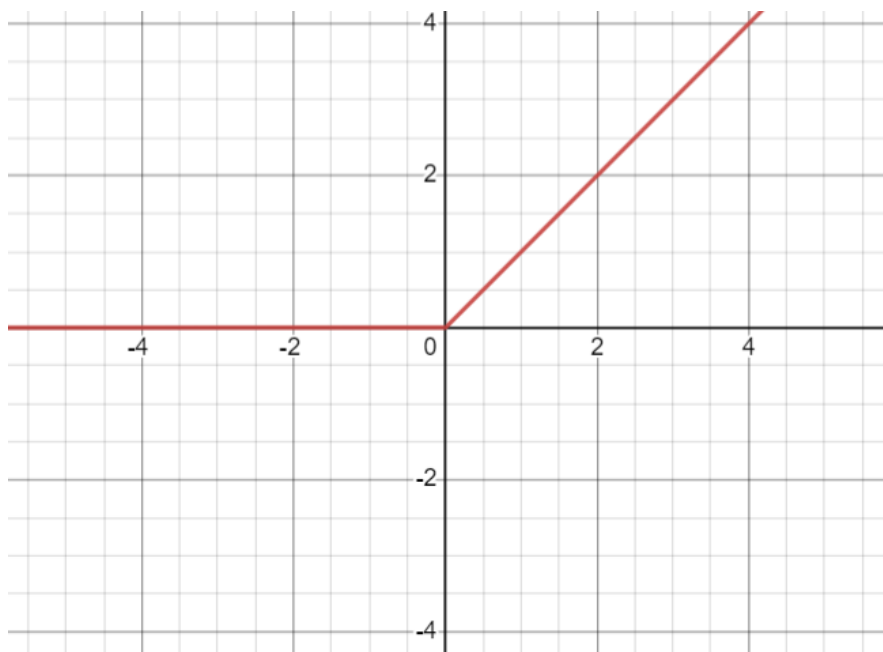
Różne funkcje wykorzystywane do aktywacji neuronów:



Rysunek 2: Funkcja sigmoidalna



Rysunek 3: Funkcja Tangensa Hiperbolicznego (\tanh)



Rysunek 4: Funkcja ReLU:

2 Przebieg ćwiczenia

2.1 Cel

Celem było stworzenie sieci neuronowej zdolnej do przewidywania zadanej funkcji w przedziale $[-10, 10]$

$$f(x) = x^2 \cdot \sin(x) + 100 \cdot \sin(x) \times \cos(x) \quad (1)$$

2.2 Projekt perceptronu

Przeprowadziliśmy badania dla jednej i dwóch warstw ukrytych, wyniki były bardzo zbliżone i dodatkowa warstwa nie przyniosła efektów

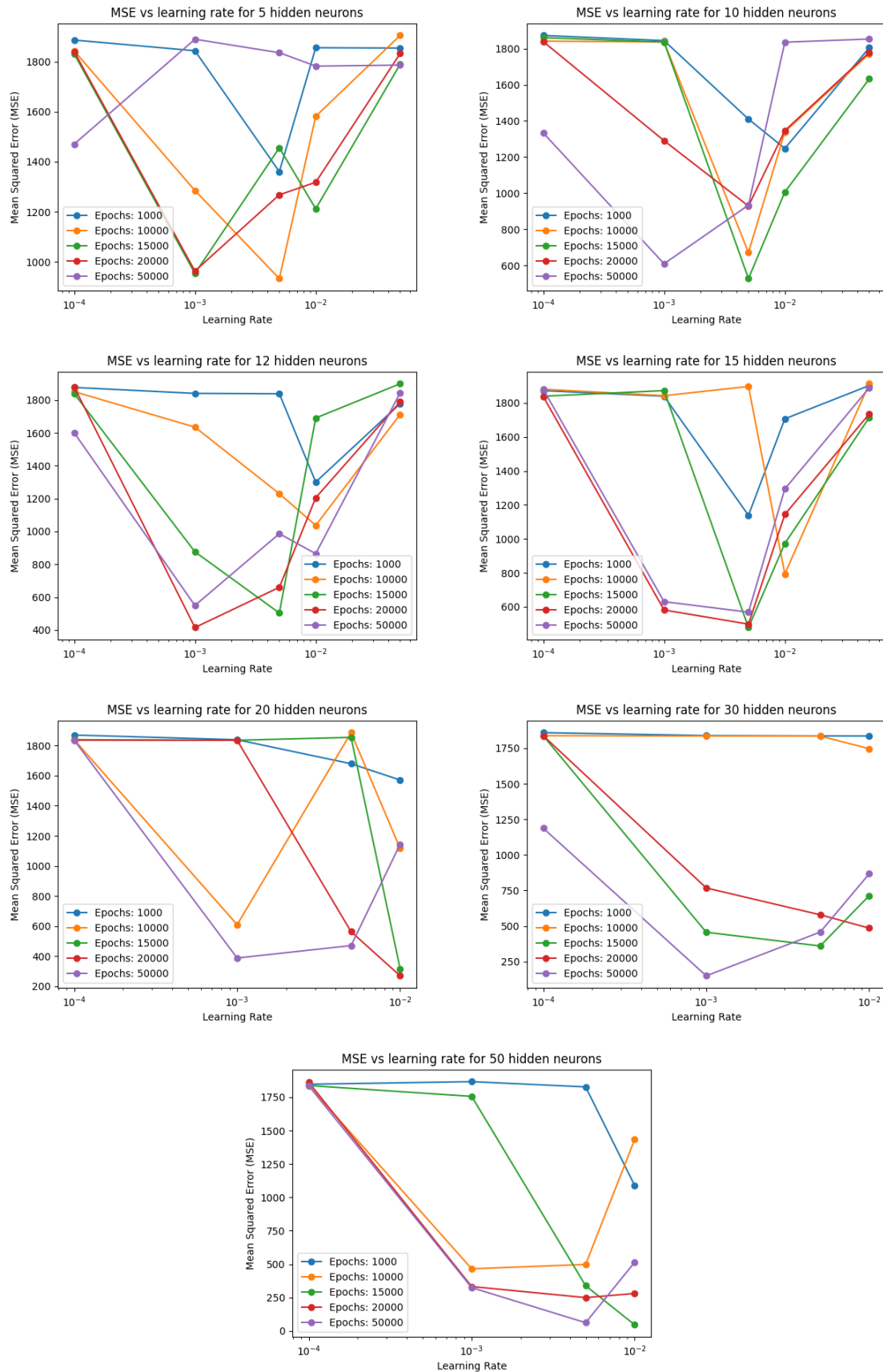
Badaliśmy też wpływ ilości epok, współczynnika learning rate oraz ilości neuronów w warstwie ukrytej

2.3 Skalowanie danych

Zauważyliśmy, że model dobrze przewiduje wartości funkcji z środkowego przedziału. Postawiliśmy więc skalować dane do przedziału $[-1, 1]$ za pomocą funkcji MinMaxScaler z pakietu sklearn

2.4 Wyniki

Badania wpływu hiperparametrów perceptronu na dokładność aproksymacji dokonywanej przez model mierzonej za pomocą błędu kwadratowego:

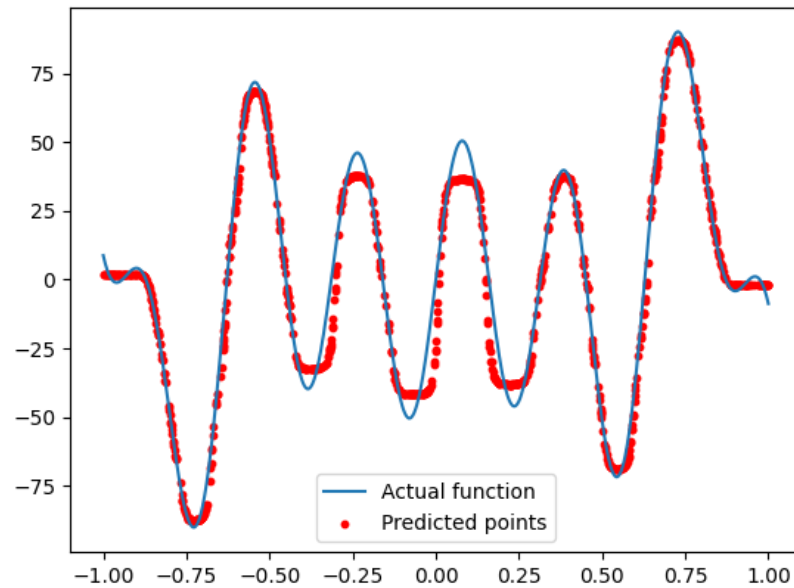


Rysunek 5: Wartość końcowego MSE aproksymacji w zależności od hiperparametrów użytych do trenowania modelu

Po przeanalizowaniu powyższych wykresów wybraliśmy optymalną konfigurację hiperparametrów:

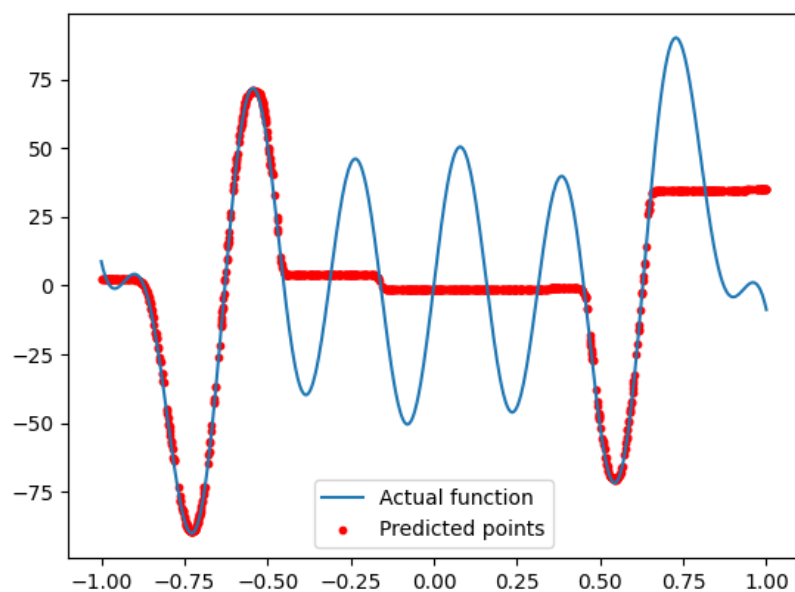
- Współczynnik uczenia: 0.01
- Liczba neuronów w warstwie ukrytej: 50
- Liczba epok: 50000
- Funkcja aktywacji: tangens hiperboliczny

Wykres przedstawiający efekt końcowy aproksymacji dla najlepszego modelu:



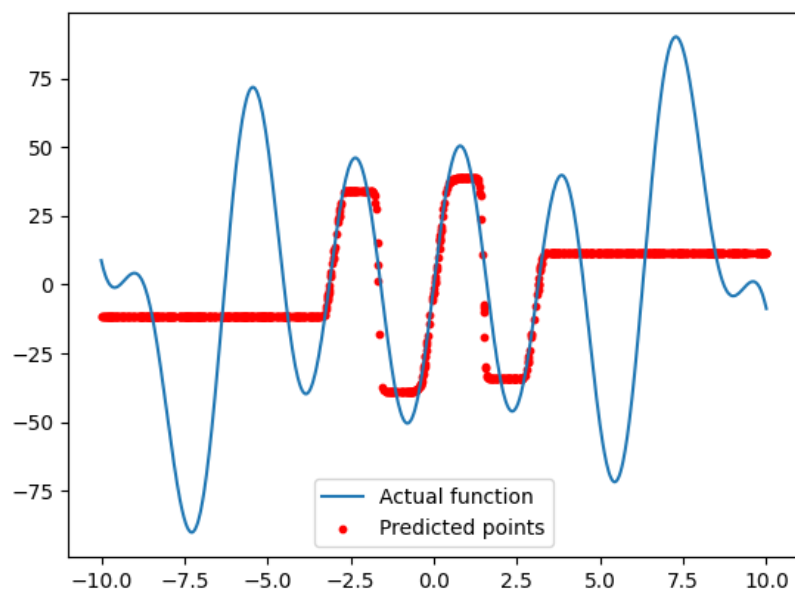
Rysunek 6: Aproksymacja funkcji przez optymalny model (funkcja aktywacji tangensa hiperbolicznego)

Dla porównania, tak wygląda wykres aproksymacji dla optymalnego modelu uczonego przy użyciu sigmoidalnej funkcji aktywacji:



Rysunek 7: Aproksymacja funkcji przez optymalny model (sigmoidalna funkcja aktywacji)

Wykres dla modelu uczonego przy użyciu sigmoidalnej funkcji aktywacji, bez przeprowadzenia normalizacji danych wejściowych:



Rysunek 8: Aproksymacja funkcji przez optymalny model (sigmoidalna funkcja aktywacji)

2.5 Wnioski

Po wypróbowaniu różnych konfiguracji i odmiennych architektur, uzyskaliśmy zadowalające wyniki, co widać na wykresie 6. Użycie normalizacji pozwoliło na pokrycie krańców przedziału. Najlepszą funkcją aktywacji okazał się tangens hiperboliczny, a dodatkowa warstwa nie przyniosła poprawy, wydłużając znacznie czas uczenia się modelu. Sensowne wyniki osiągnęliśmy powyżej 20 neuronów w warstwie ukrytej. Wyznaczona przez nas optymalna wartość współczynnik uczenia wyniosła 0.01. Jest to stosunkowo wysoka wartość, jednak zejście do niższych jego poziomów powodowało utykanie w minimach lokalnych funkcji straty. Rozwiązaniem tego problemu mogłoby być np. zaimplementowanie mechanizmu adaptacji learning rate'u.

Literatura

- [1] Muraszkiewicz, M., Nowak, R. (Red.). (2022). *Sztuczna inteligencja dla inżynierów. Metody ogólne*. Oficyna Wydawnicza Politechniki Warszawskiej.