

Sztuczne sieci neuronowe

Zofia Zdonek, Rafał Zwierzyński

January 2026

1 Implementacja perceptrona wielowarstwowego wraz z metodami uczenia

1.1 Struktura

Perceptron ma trzy warstwy: wejściową, ukrytą i wyjściową. W każdej warstwie jest n neuronów (poza warstwą wejściową ona tylko tworzy wejście dla następnej warstwy na podstawie wektora X). W warstwie wejściowej tak naprawdę dodajemy do wektora wejściowego X_i obciążenie neuronu $b_i = 1$ oraz określamy wstępne wartości wag ϕ które trafiają do wejścia warstwy ukrytej są one losowymi liczbami z rozkładu normalnego $N(0, 1)$. Następnie w warstwie ukrytej mamy n neuronów sigmoidalnych z funkcją aktywacji $\psi(s) = \tanh(s)$ których wyjście staje się następnie wejściem ostatniej warstwy wyjściowej. W niej mamy tylko neurony liniowe czyli takie że $\psi(s) = s$.

Celem perceptrona jest aproksymacja zadanej funkcji $f(x) : R^n \rightarrow R$. Przez to że wektor wejściowy ma postać $X_m = (x_1, x_2, \dots, x_m)$ to dla funkcji takiej że $n < m$ sieć może być uczena na wielu punktach naraz. Wyjście perceptrona zawiera przybliżenie wartości funkcji dla każdego punktu wprowadzonego do sieci.

1.2 Uczenie

Uczone są jednocześnie warstwa ukryta i wyjściowa co oznacza że są razem optymalizowane. Funkcja celu

$$q(\phi) = \frac{1}{N} * \sum_{i=1}^N ||f(x_i, \phi) - y_i||^2 \quad (1)$$

określa dopasowanie wag ϕ oraz obciążen neuronów b sieci przez ocenę błędu kwadratowego na danych treningowych. Funkcja ta jest minimalizowana.

2 Eksperyment

Badana została złożona oscylacyjna funkcja $f(x) : [-10, 10] \rightarrow R$:

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

Perceptron miał posłużyć do aproksymacji powyższej funkcji. Sieć uczyła się na 200 elementowym zbiorze trenującym. Zbiór testowy był dwukrotnie większy. Badane były dwa aspekty sieci neuronowej:

1. Wpływ liczby neuronów na jakość aproksymacji funkcji

Badaliśmy jakość aproksymacji funkcji przez sieć neuronową dla następujących ilości neuronów w warstwie $n \in \{5, 10, 20, 30, 40, 50\}$.

2. Różnice w aproksymacji w zależności od zastosowania metody gradientowej lub ewolucyjnej do znajdowania wag sieci

Zaimplementowane zostały dwie metody optymalizacji: gradientowa i ewolucyjna. Gradientowa opiera się na gradiencie prostym z propagacją wsteczną gdzie współczynnik uczenia jest równy 0.01 a liczba iteracji to 3000. Metoda ewolucyjna bazuje na losowym przeszukiwaniu gdzie liczba kandydatów to 30000 a skala mutacji jest równa 3.

3 Wyniki eksperymentu

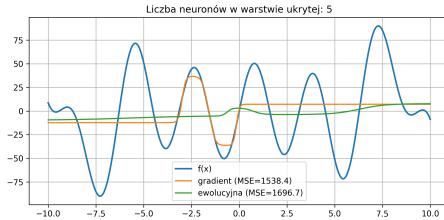


Figure 1: Porównanie metod dla 5 neuronów w warstwie

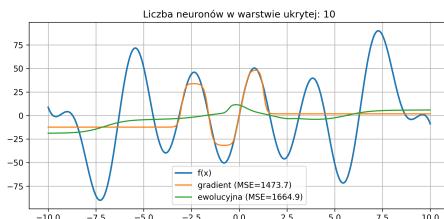


Figure 2: Porównanie metod dla 10 neuronów w warstwie

Obie metody dla małych $n \in \{5, 10\}$ nie są w stanie poprawnie odzworować struktury funkcji jednakże gradient radzi sobie znaczco lepiej ponieważ niektóre zachowania funkcji przybliżanej pokrywają się z tym co wyprodukowała sieć. Jednak aproksymacje są zbytnio wygładzone i nie oddają realnego zachowania funkcji badanej. Występuje silne niedouczenie.

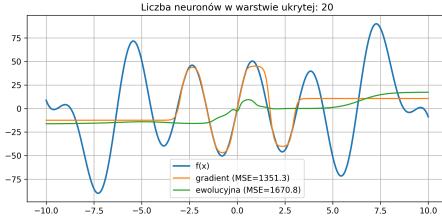


Figure 3: Porównanie metod dla 20 neuronów w warstwie

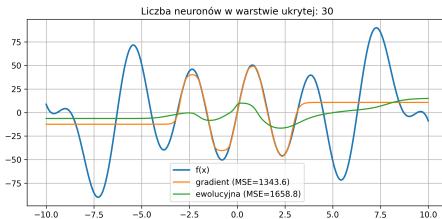


Figure 4: Porównanie metod dla 30 neuronów w warstwie

Dla średnich ilości neuronów $n \in \{20, 30\}$ widać znaczącą poprawę jakości aproksymacji metodą gradientową. Sieć zaczyna poprawnie odwzorywać najważniejsze maksima i minima funkcji badanej. Kształt i amplituda przybliżenia są również coraz bardziej dokładne. Metoda ewolucyjna też ulega niewielkiej poprawie chociaż nadal tworzy mocno wygładzoną aproksymację.

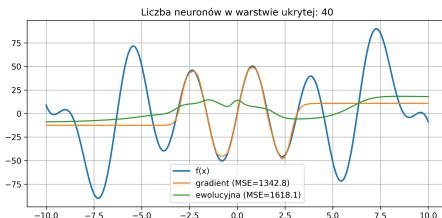


Figure 5: Porównanie metod dla 40 neuronów w warstwie

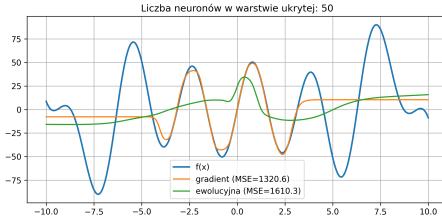


Figure 6: Porównanie metod dla 50 neuronów w warstwie

Dla większych ($n \in \{40, 50\}$) sieci metoda gradientowa osiąga najlepsze rezultaty. Poprawnie oddaje lokalne oscylacje i strukturę funkcji. Natomiast metoda ewolucyjna dalej poprawia się bardzo powoli i raczej nieskutecznie. Coraz bliżej jest jednego z maksimów jednak dalej jest nadmiernie wygładzona.

4 Wnioski

Mocno zauważalny jest w tym badaniu kolosalny wpływ liczby neuronów w warstwie na jakość aproksymacji funkcji badanej. Zbyt mała ilość prowadzi do niedouczenia sieci i braku zdolności do odwzorowania złożonej struktury funkcji. Widać też dużą przewagę metody gradientowej nad metodą ewolucyjną w zadaniu aproksymacji funkcji ciągłej i oscylującej. Gradient o wiele szybciej i lepiej wykorzystuje wzrost liczby neuronów w sieci, osiąga niższe wartości błędów oraz dokładniej odwzorowuje lokalne własności funkcji. Metoda ewolucyjna charakteryzuje się słabą skalowalnością. Wzrost liczby neuronów nie ma przełożenia proporcjonalnego na poprawę przybliżenia.

Dla analizowanej funkcji najlepszy kompromis między złożonością sieci a poprawnością aproksymacji zapewnia gradientowa metoda uczenia z liczbą neuronów między 20 a 50. Wyniki pokazują że perceptron wielowarstwowy wykorzystujący gradient do szukania odpowiednich wag jest dostatecznie dobry w aproksymacji funkcji złożonych.