

WSI

Ćwiczenie nr 5 — Sieci Neuronowe

Antoni Grajek
Konrad Karpiuk
18.05.2024

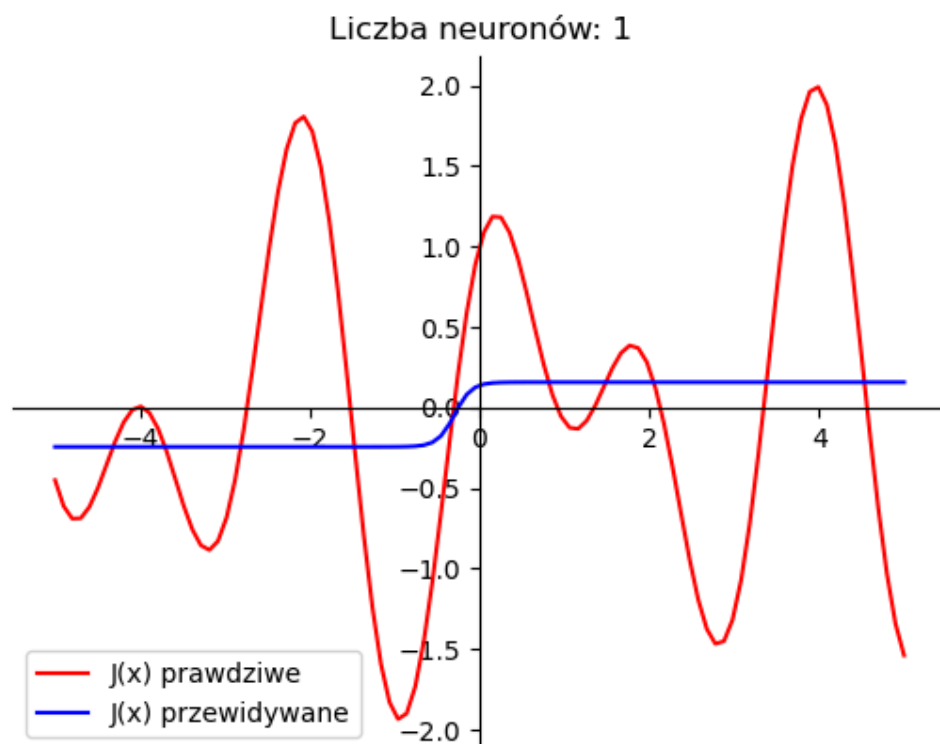
Cel zadania

Do zaimplementowania był Perceptron z jedną warstwą ukrytą, Aproksymujący funkcję:
 $J(x) = \sin(x \cdot \sqrt{4}) + \cos(x \cdot \sqrt{10})$

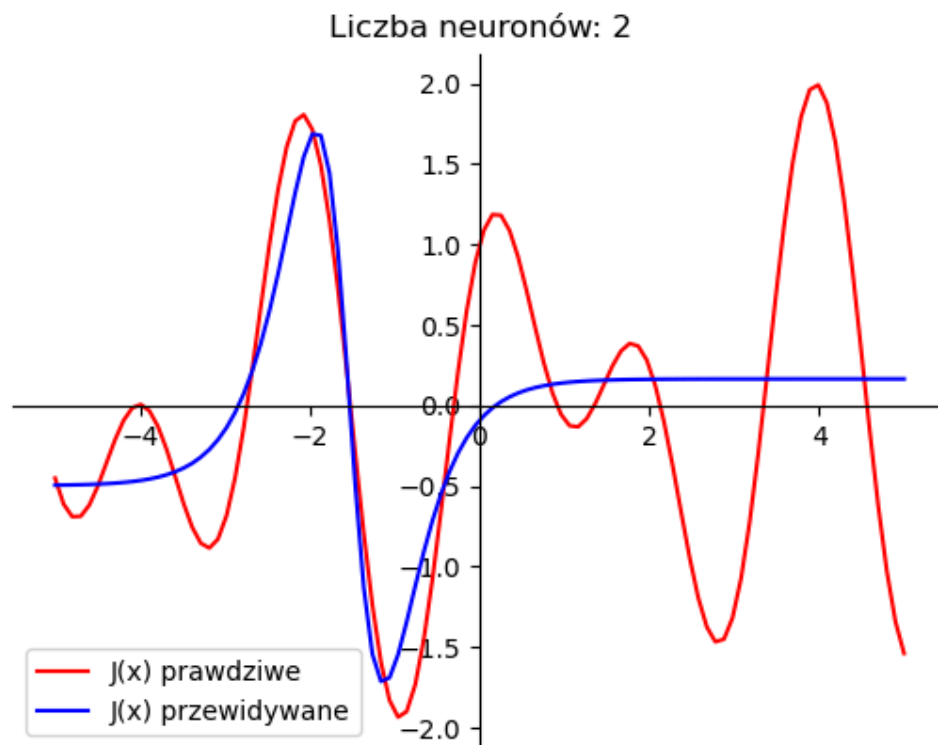
Następnie zbadać zależności parametrów sieci neuronowej z jakością jej aproksymacji.

Eksperymenty:

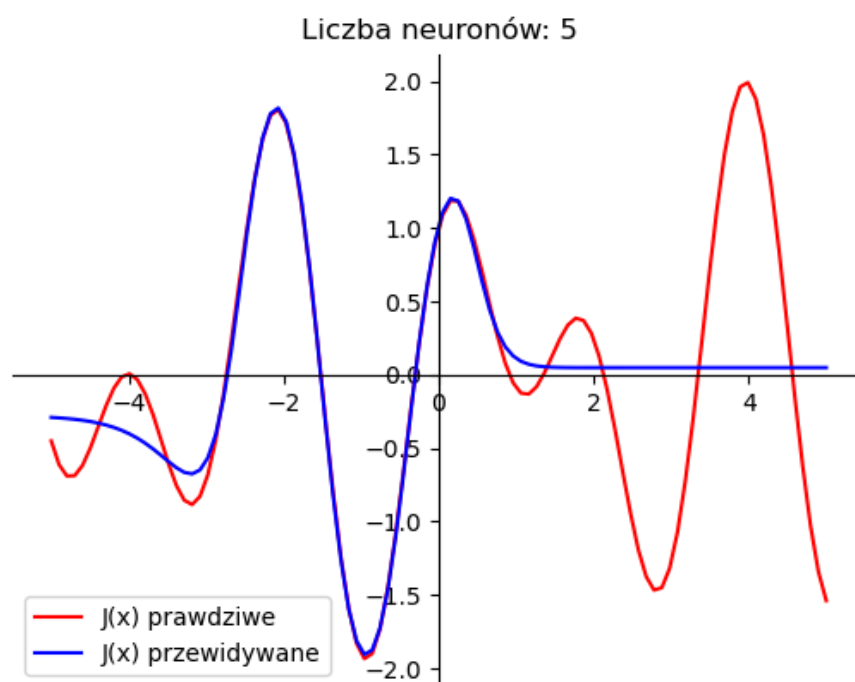
Po zaimplementowaniu sieci neuronowej przetestowano ją pod względem wpływu liczby neuronów tworzących warstwę ukrytą na jakość aproksymacji. Za każdym razem liczba iteracji wynosiła 15 000, a współczynnik learning rate był ustawiony na 0,003.



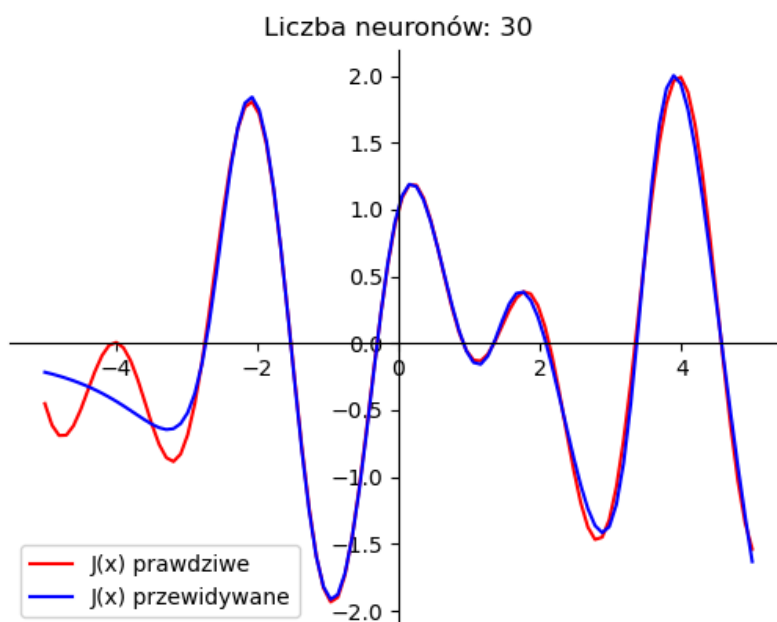
Warstwa ukryta złożona z jednego neuronu pozwala na zastosowanie tylko jednego współczynnika pomnożonego przez dane wejściowe oraz jednego współczynnika polaryzującego, zupełnie tak samo jak funkcja liniowa. Nie dziwi zatem kształt wyjścia modelu: jest to po prostu funkcja aktywacji neuronu, która podlega przekształceniu liniowemu. Jeden neuron nie będzie dobrze aproksymował funkcji o innym kształcie niż funkcja aktywacji.



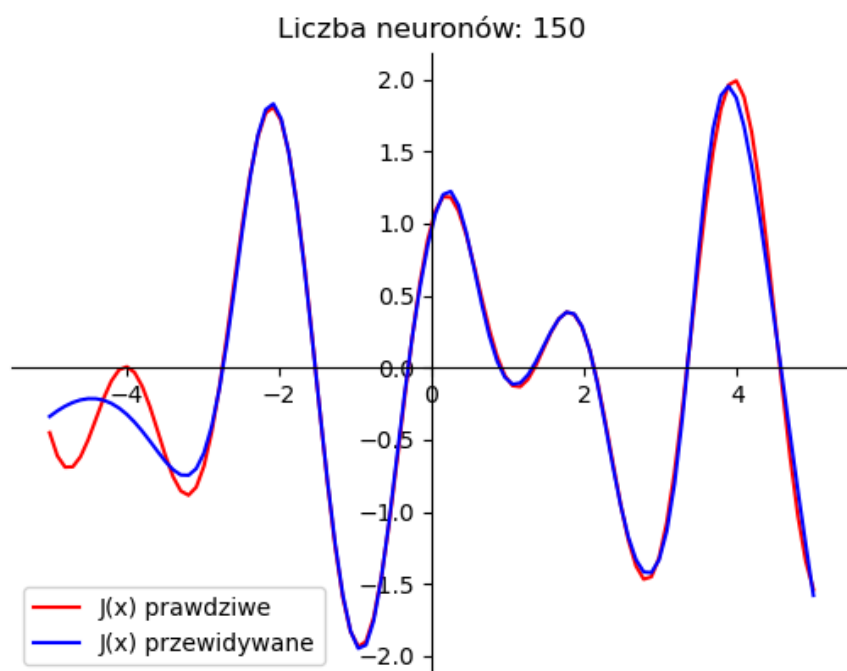
Zastosowanie dwóch neuronów pozwoliło oddać kształt wykresu dla argumentów z przedziału $[-2; 1]$, natomiast dla reszty argumentów nie można powiedzieć o jakiegokolwiek aproksymacji.



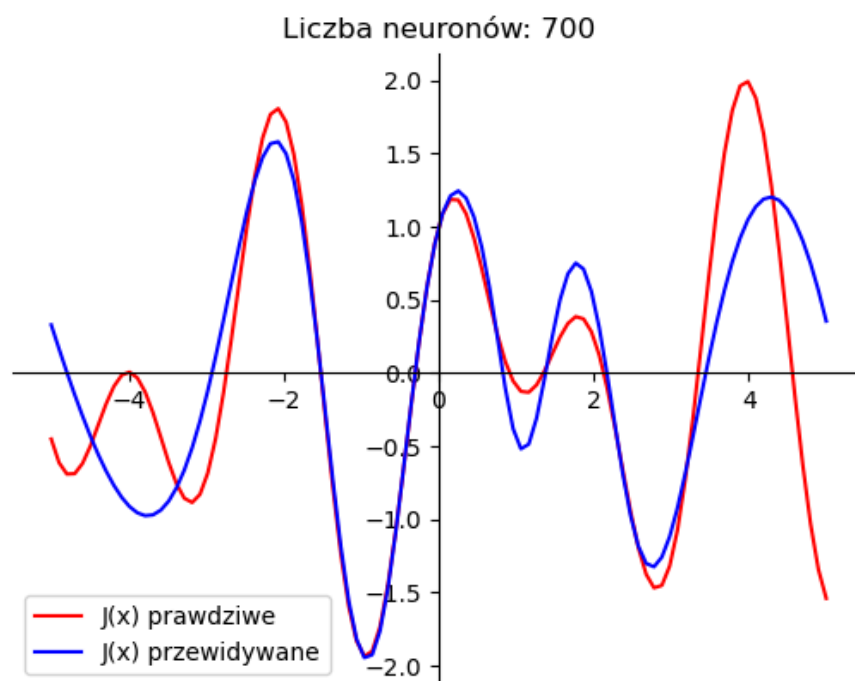
Poprzez zastosowanie 5 neuronów osiągnięto bardzo dobre przybliżenie dla argumentów z przedziału $[-3; 0]$, dla pozostałych argumentów aproksymacja dalej wygląda słabo.



Zastosowanie 30 neuronów pozwala na bardzo dobrą aproksymację pierwotnej funkcji. Niestety model dalej ma problem z oddaniem kształtu wykresu dla argumentów z przedziału $[-5; -3]$.



Mimo zastosowania pięciokrotnie większej liczby neuronów niż w poprzednim eksperymencie, wynik tylko nieznacznie się polepszył.



Zwiększając liczbę neuronów do 700 uzyskano pogorszenie działania sieci neuronowej. Zachowanie to może wynikać z faktu, że przy tak dużej liczbie neuronów optymalizacja sieci staje się zadaniem potrzebującym zdecydowanie większej liczby iteracji.

Dla każdego eksperymentu obliczono błąd średniokwadratowy jako wskaźnik jakości aproksymacji. Wyniki przedstawiono w poniższej tabeli:

Liczba neuronów	1	2	5	30	150	700
Błąd średniokwadratowy	0,9426	0,5716	0,4394	0,0217	0,0167	0,3022

Spośród modeli z przeprowadzonych eksperymentów najmniejszym błędem średniokwadratowym cechowała się sieć złożona ze 150 neuronów.

Komentarz do eksperymentów:

Liczba neuronów:

Liczba neuronów w warstwie ukrytej sieci neuronowej ma istotny wpływ na jakość aproksymacji funkcji, którą sieć modeluje. Większa liczba neuronów zwiększa zdolność sieci do aproksymacji skomplikowanych funkcji, ponieważ każdy neuron dodaje nowe możliwości kombinacji wag i aktywacji, co pozwala lepiej odwzorowywać nieliniowe zależności w danych. Zbyt mała liczba neuronów może prowadzić do niedopasowania, gdzie sieć nie ma wystarczającej mocy obliczeniowej, aby dokładnie modelować zależności w danych, co skutkuje dużymi błędami aproksymacji. Natomiast zbyt duża liczba neuronów może prowadzić do przewymiarowania, gdzie sieć staje się zbyt dopasowana do danych treningowych, co ogranicza jej zdolność do generalizacji na nowe dane. Kluczowe jest znalezienie odpowiedniego balansu między złożonością a generalizacją. Warto również brać pod uwagę koszt obliczeniowy, ponieważ większa liczba neuronów zwiększa wymagania dotyczące pamięci i czasu obliczeń.

Iteracje nauki:

Iteracje, czyli liczba kroków, które algorytm podejmuje podczas uczenia, mają istotny wpływ na proces trenowania sieci neuronowej. Wraz z kolejnymi iteracjami model staje się coraz lepiej dopasowany do danych treningowych, stopniowo minimalizując błąd. Jednak zbyt duża liczba iteracji może prowadzić do przeuczenia, gdzie model staje się nadmiernie dopasowany do danych treningowych i traci zdolność do generalizacji na nowe dane. Z drugiej strony, zbyt mała liczba iteracji może skutkować niedostatecznym dopasowaniem modelu.

Współczynnik uczenia:

Tu sytuacja jest analogiczna jak przy doborze współczynnika kroku w algorytmie najszybszego wzrostu implementowanego na laboratorium nr 1. Zbyt wysoki learning rate sprawia, że kroki są duże, co może prowadzić do przeskakiwania przez optymalne

rozwiązanie i niestabilności w procesie uczenia. Z kolei zbyt niski learning rate powoduje, że kroki są małe, co prowadzi do wolniejszej konwergencji i może sprawić, że proces uczenia będzie nieefektywny.