Jan Jędrzejewski
Jakub Niezabitowski
22.01.2022r

WSI - zadanie 5

Sztuczne sieci neuronowe

Perceptrony wielowarstwowe

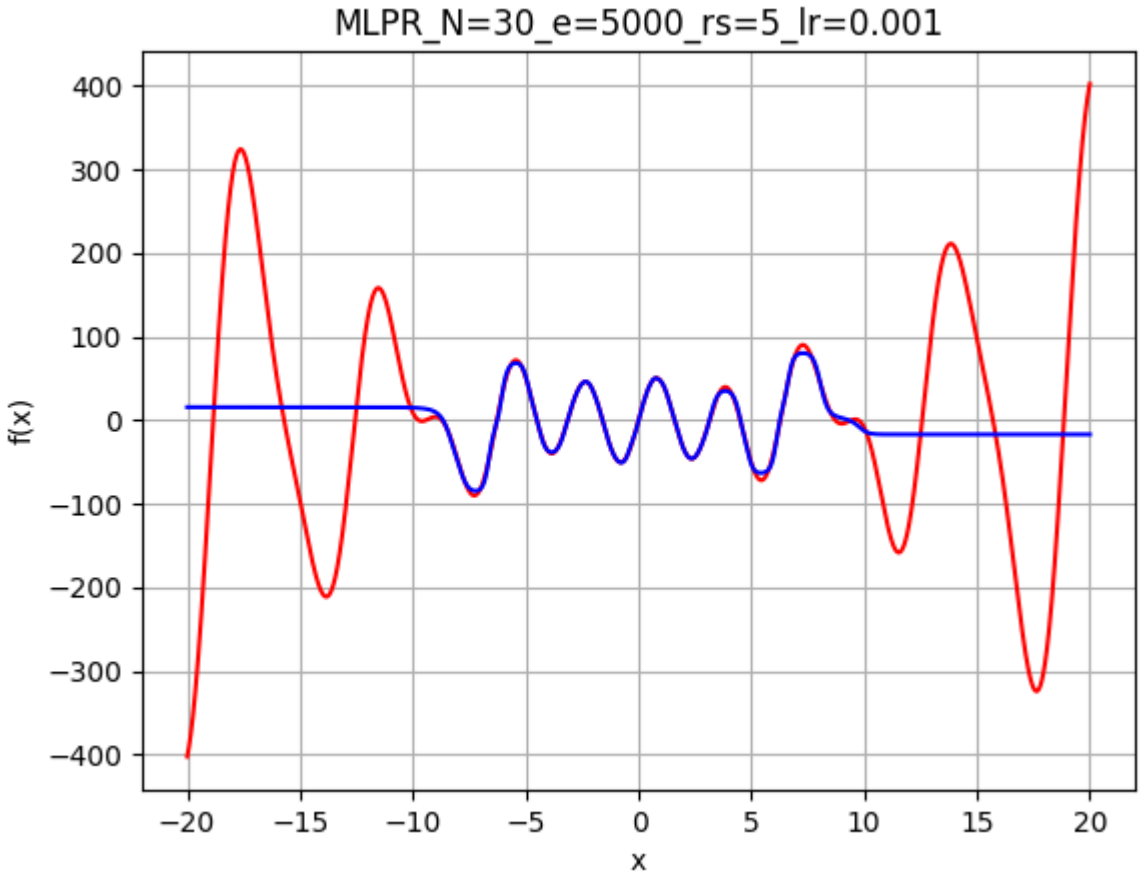
Treść zadania:
Zaimplementuj perceptron wielowarstwowy, który posłuży do aproksymacji zadanej funkcji f(x). Zbadaj wpływ liczby neuronów w warstwie na jakość uzyskanej aproksymacji.

Rozwiązanie biblioteki sklearn - model sieci MLPRegressor

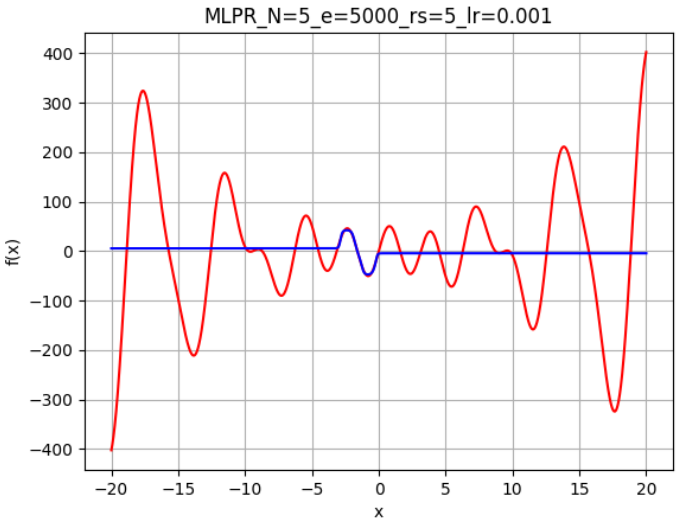
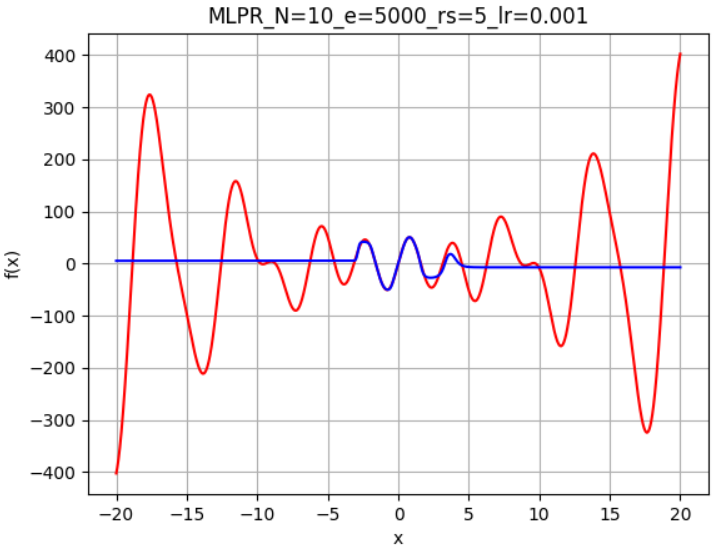
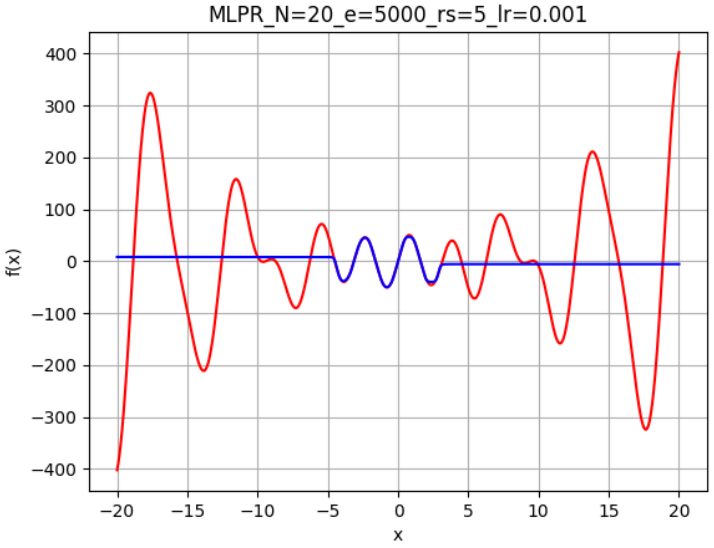
W pliku plot.py przygotowaliśmy funkcję rysującą wykresy i dodatkowo zbadaliśmy wpływ parametrów na przybliżenia funkcji przez model sieci neuronowej z biblioteki sklearn.

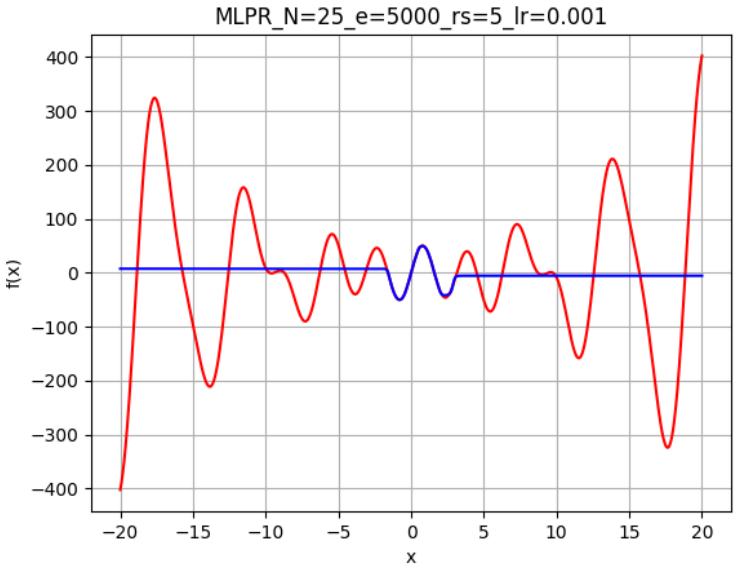
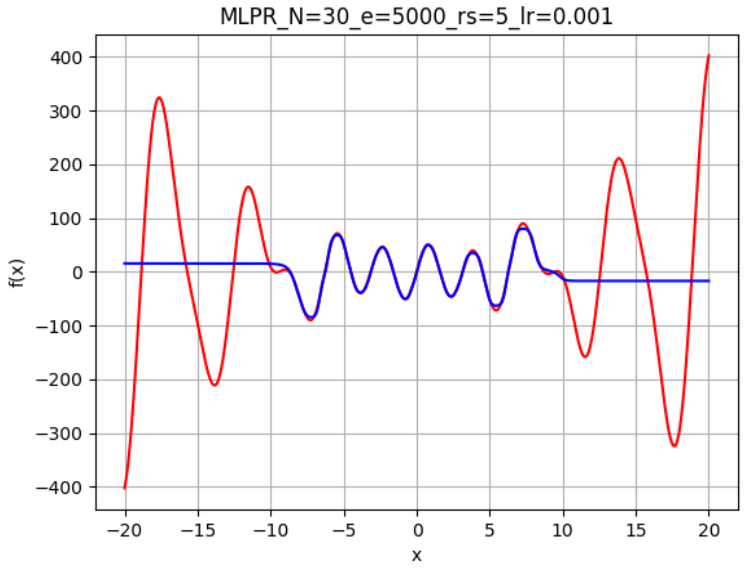
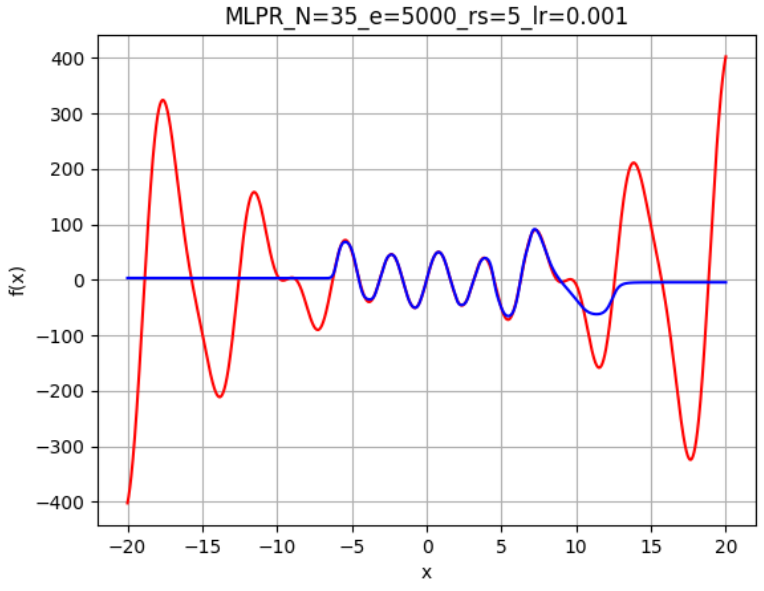
Za pomocą tego modelu najlepsze wyniki otrzymano dla poniższych parametrów. Na następnej stronie znajduje się zestawienie aproksymacji funkcji dla różnej liczby perceptronów w warstwie ukrytej. I dla stałej wartości pozostałych parametrów.

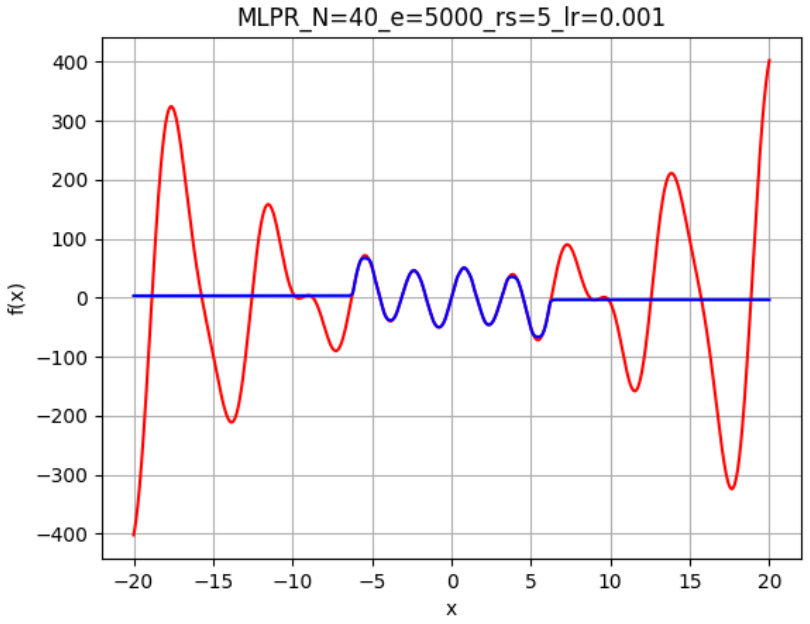
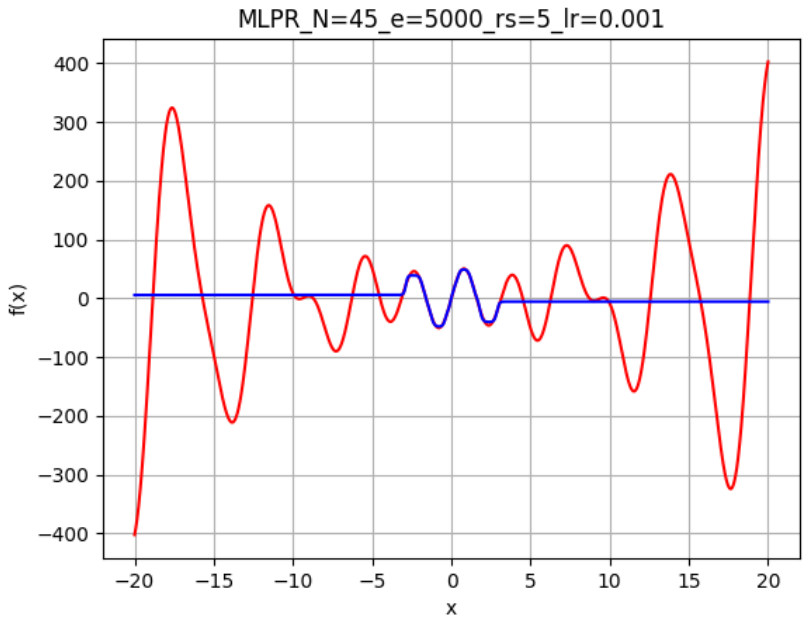
Neurony=30, epoki=5000, mini_bath=100, learning_rate=0.001, random_state=5,
funkcja aktywacji=sigmoidalna, solver=SGD



Na Następnych stronach aproksymacja funkcji względem liczby neuronów:

Liczba neuronów: 5	
Liczba neuronów: 10	
Liczba neuronów: 20	

Liczba neuronów: 25	
Liczba neuronów: 30	
Liczba neuronów: 35	

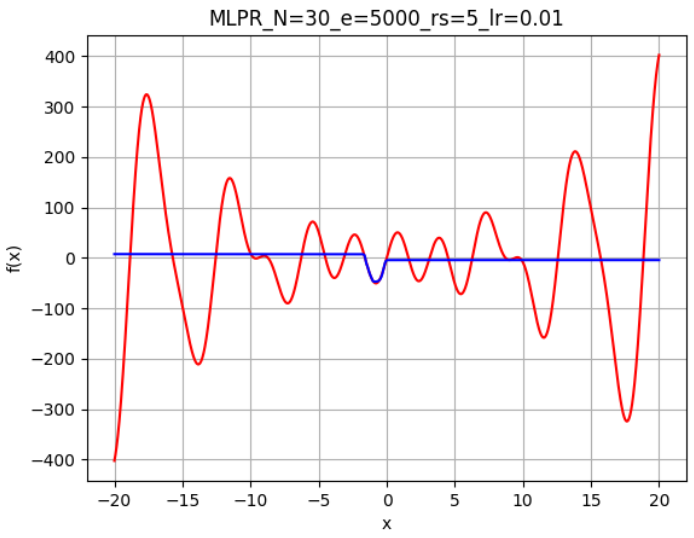
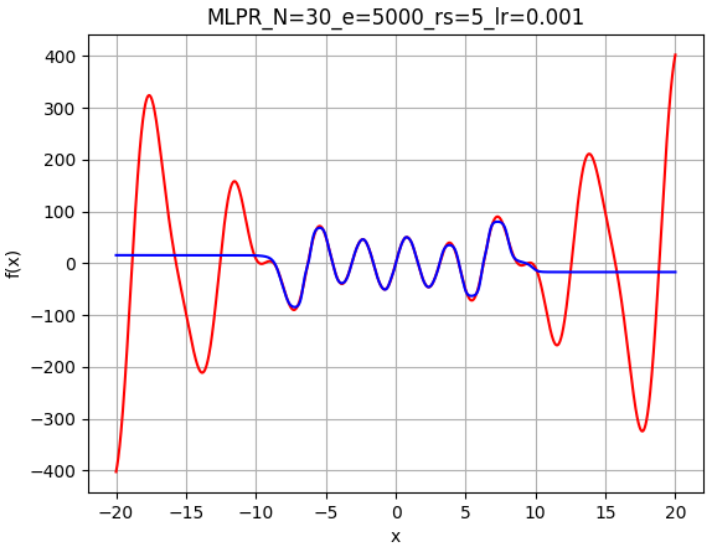
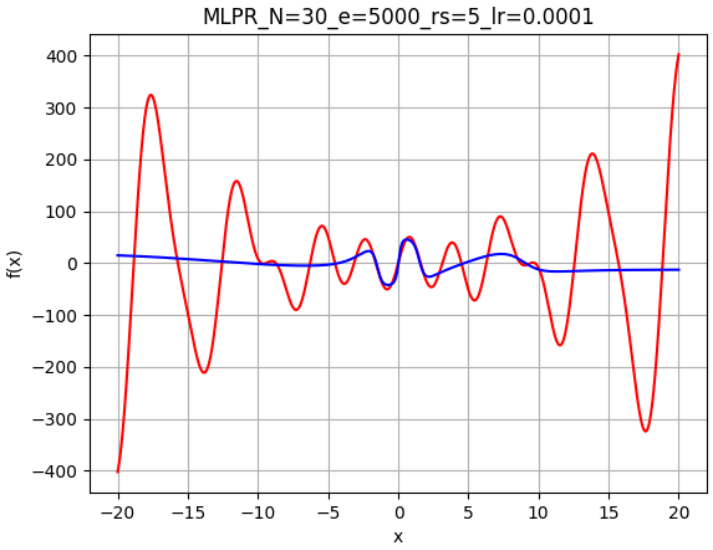
Liczba neuronów: 40	
Liczba neuronów: 45	

Badanie pozostałych parametrów

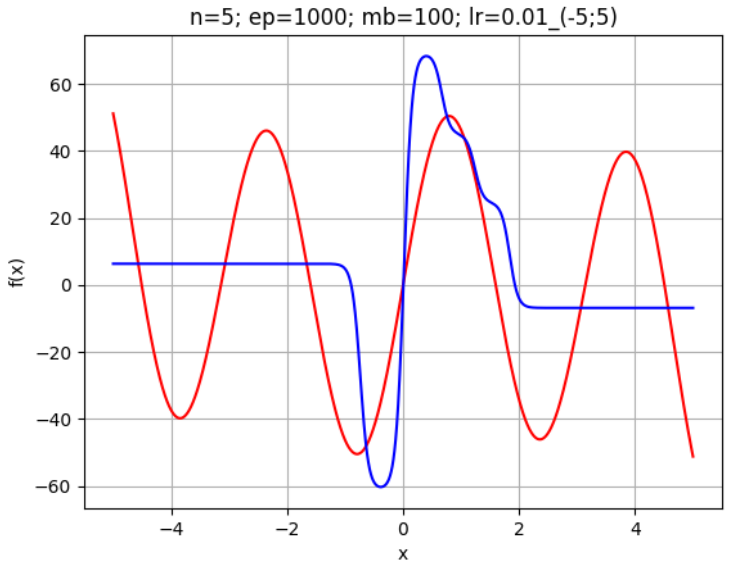
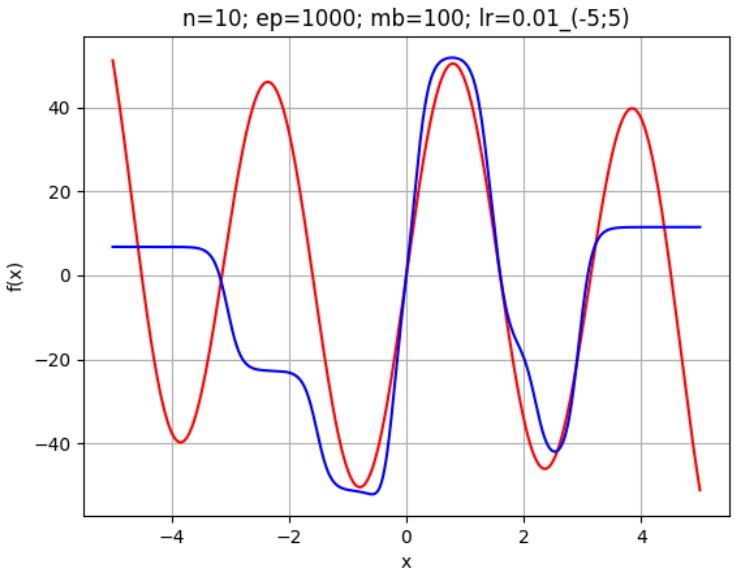
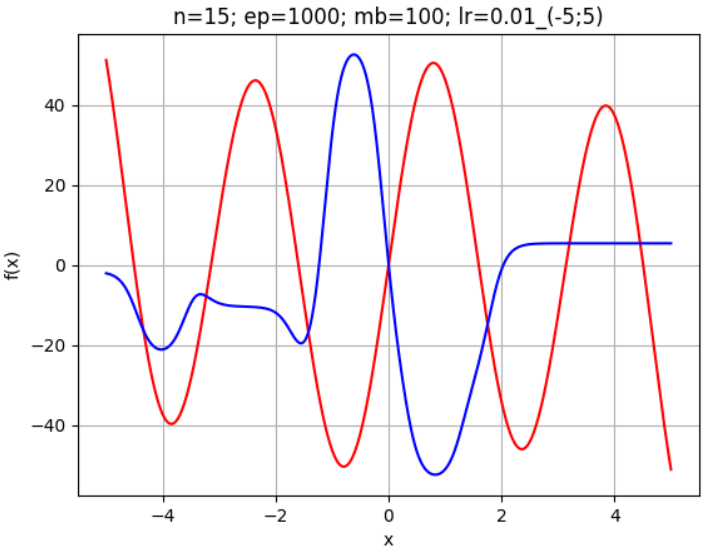
Stworzyliśmy kilka modeli dla różnych wartości parametru random_state odpowiedzialnego za początkowe wylosowane wartości wag i biasów dla warstwy ukrytej. Nie miało to większego wpływu na wyniki, ale najlepsze przybliżenie dostaliśmy dla wartości 5, dlatego jej używamy w pozostałych doświadczeniach.

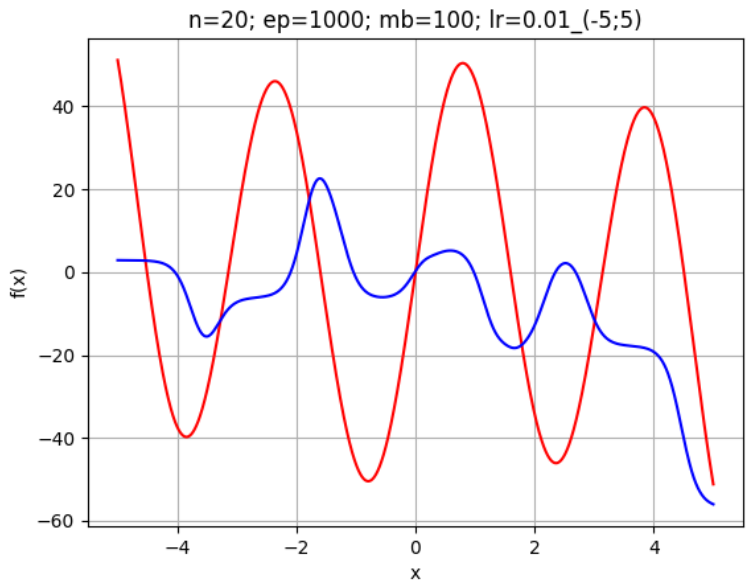
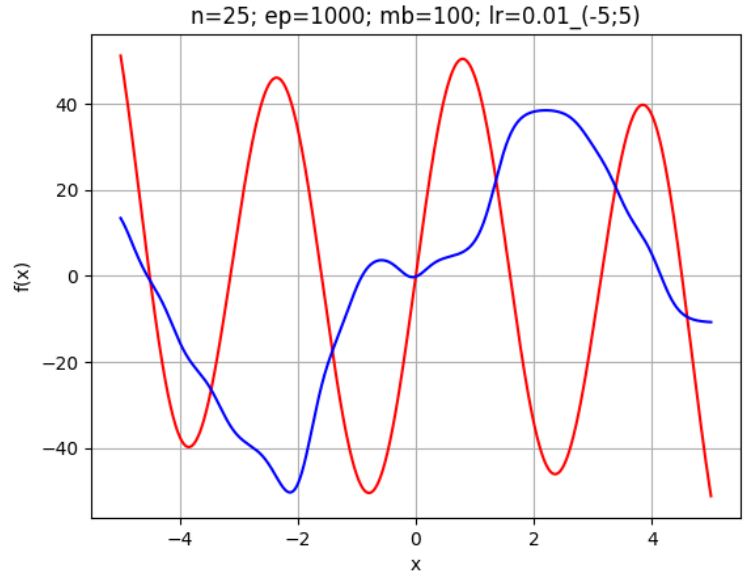
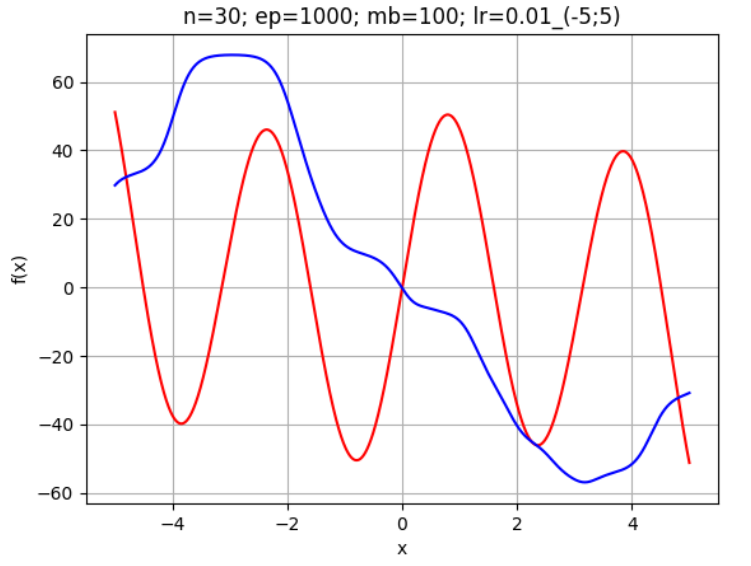
Parameter learning rate należało dostosować do rozmiaru zbioru testującego. Dla 40 000 próbek, najlepiej sprawdzała się wartość 0,001, co demonstrują poniższe wykresy.

Learning rate

Learning rate: 0.01	
Learning rate: 0.001	
Learning rate: 0.0001	

Nasze rozwiązanie

Liczba neuronów: 5	
Liczba neuronów: 10	
Liczba neuronów: 15	

Liczba neuronów: 20	
Liczba neuronów: 25	
Liczba neuronów: 30	

Analiza naszego rozwiązania

Zaimplementowany przez nas algorytm algorytm jest niestabilny, nie zawsze dobrze aproksymuje zadaną funkcję. Na wykresie z 15 neuronami niebieska linia przypomina funkcję o przeciwnych wartościach do danej. Dodatkowo większa liczba epok wcale nie wpływała na poprawę aproksymacji. Algorytm najlepiej zachowywał się dla 10-15 neuronów. Dlatego przeprowadziliśmy dodatkowe badanie w tym zakresie. Oto najepsze przybliżenie jakie otrzymaliśmy, dla 12 neuronów: