Lidia Sobońska 310903
Arkadiusz Kojtek 310746

RAPORT

WSI - SIECI NEURONOWE

Uruchamianie programu

Przed uruchomieniem programu konieczne jest zainstalowanie trzech modułów:
Numpy do wykonywania obliczeń matematycznych:
pip install numpy
tqdm umożliwiający wizualizację przewidywanego czasu działania programu poprzez tworzenie paska ładowania
pip install tqdm
MinMaxScaler z **sklearn.preprocessing** - Sieć powinna otrzymywać dane znormalizowane. MinMaxScaler umożliwia więc łatwe przeskalowanie zestawu treningowego, testowego oraz wyników aby były z zakresu (0,1).
pip install -U scikit-learn
pyplot z **matplotlib** - umożliwia tworzenie wykresów na podstawie otrzymanych danych.
pip install matplotlib

Implementacja algorytmu

Implementacja algorytmu składa się z dwóch klas:

Layer - klasa reprezentująca poszczególne warstwy sieci neuronowej. Jako argumenty konstruktora przyjmuje wielkość swojej warstwy oraz wielkość warstwy jej poprzedzającej. W konstruktorze, ponadto informacje te są wykorzystywane do tworzenia tablic **weights** oraz **biases**.

Składa się z trzech metod:

set_funs - metoda ta ustala funkcje aktywacji (w testach używamy funkcji sigmoidalnej)

forward - metoda do propagacji - wartości z poprzedniej warstwy są mnożone przez odpowiednie wagi, następnie sumowane; suma ta jest przepuszczana przez funkcję aktywacji i podawana do kolejnej warstwy

backward - metoda do propagacji wstecznej - na podstawie informacji o błędach w następnej warstwie korygowane są weights i biases oraz wyliczana jest wielkość błędów do przekazania do poprzedniej warstwy

Network - klasa ta reprezentuje całą sieć. Jako argument konstruktora przyjmuje wielkość warstw ukrytych.

Składa się trzech metod:

set_funs - metoda ta ustala funkcje aktywacji (w testach używamy funkcji sigmoidalnej)

train - metoda służy za przeprowadzenie procesu trenowania sieci neuronowej.

predict - metoda służy do obliczania przewidywanej przez sieć wartości funkcji dla podanego argumentu

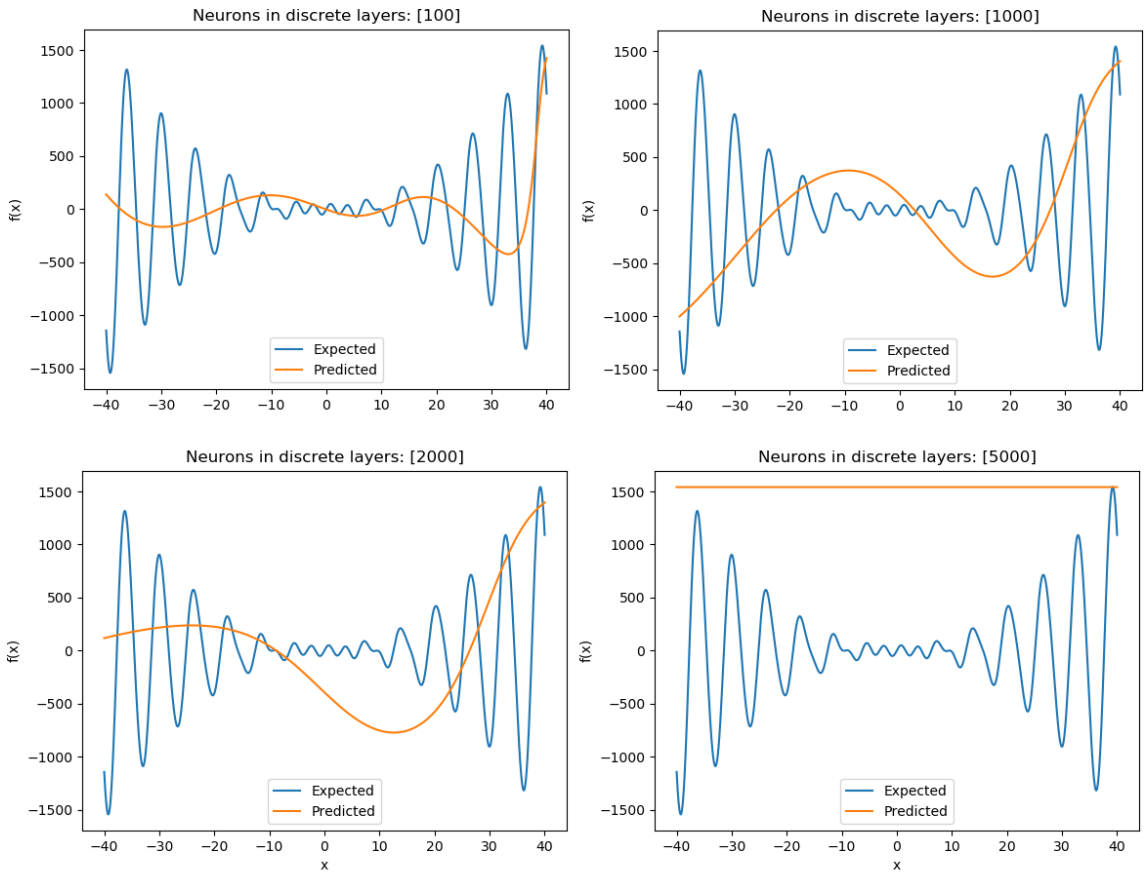
Testy

Założenia wstępne

Przed rozpoczęciem testów ustaliliśmy wstępne założenia do jakich będziemy się stosować w trakcie przeprowadzania eksperymentów. Ilość epok wynosić będzie 10 000 natomiast zestaw treningowy będzie o wielkości 801.

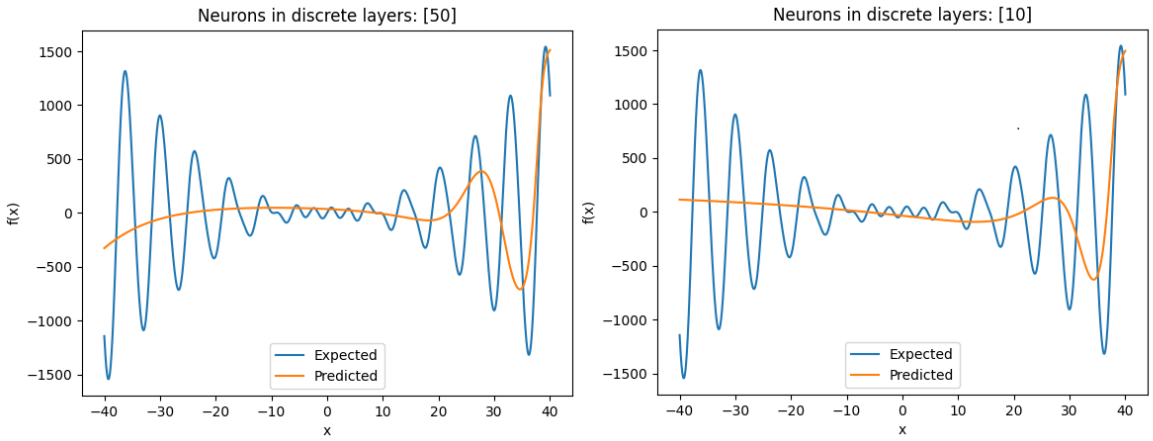
Jedna warstwa dyskretna

Na początek algorytm został przetestowany dla zadanej funkcji dla jednej warstwy dyskretnej. Testy przeprowadzone zostały zaczynając od niewielkiej ilości neuronów w warstwie po czym w kolejnych testach ta ilość była powiększana.Badania zostały przeprowadzone na 100, 1000, 2000 oraz 5000 neuronów w warstwie



Na podstawie tych testów można zauważyć że wyniki dla tylko jednej warstwy wyraźnie nie pokrywają się z oczekiwanymi wynikami funkcji.
Ponadto zauważalna jest zależność - zwiększona ilość neuronów w warstwie wpływa negatywnie na wynik działania algorytmu. Najbardziej zbliżony wynik algorytm uzyskał dla najmniejszej założonej liczby neuronów czyli 100 a najgorszy dla 5000 gdzie wynik przyjął wartość stałą.

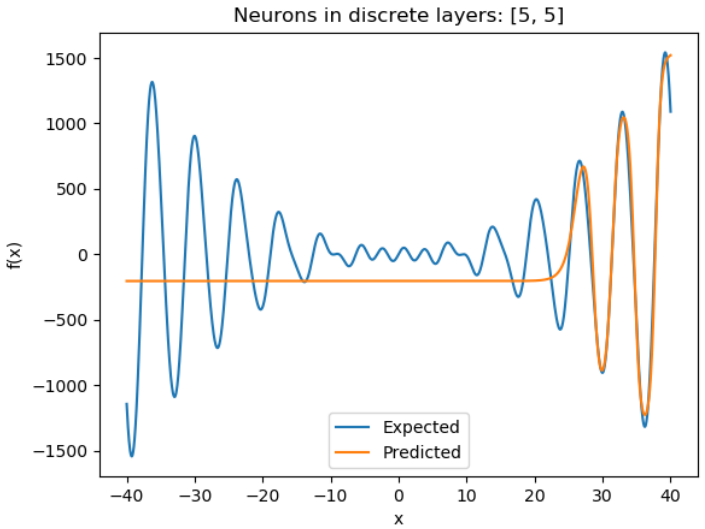
Mając to na uwadze przeprowadzone zostały dodatkowe testy dla mniejszej liczby neuronów - 10 oraz 50.



Zauważyć można wyraźnie bardziej zbliżone wyniki dla argumentów dodatnich. Jednak działanie naszego programu przyjęło zauważalnie bardziej liniowe rezultaty dla argumentów ujemnych.

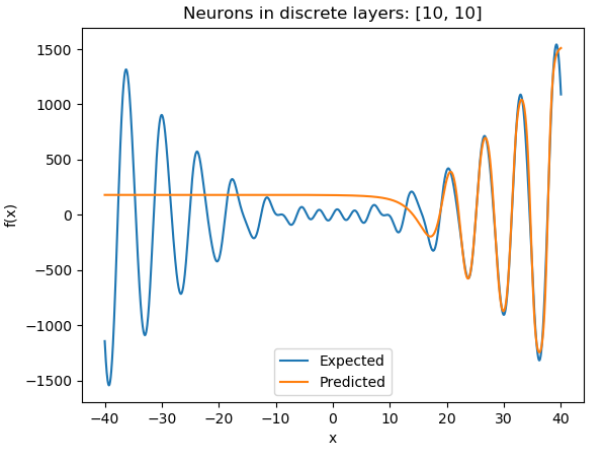
Dwie warstwy dyskretne

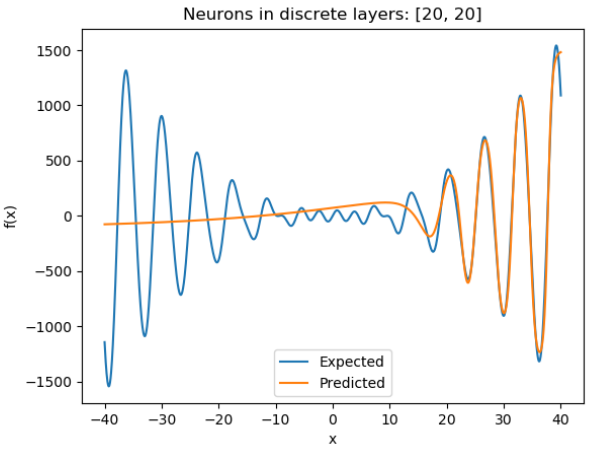
W następnej kolejności zostało sprawdzone działanie programu dla dwóch warstw dyskretnych sieci neuronowej. Jednak mając na uwadze wyniki działania programu z poprzednich testów tym razem program zaczęliśmy testować dla znacznie mniejszej ilości neuronów w warstwie. Testy zostały przeprowadzone dla 5, 10, 20, 50 oraz 100 neuronów na warstwę sieci.



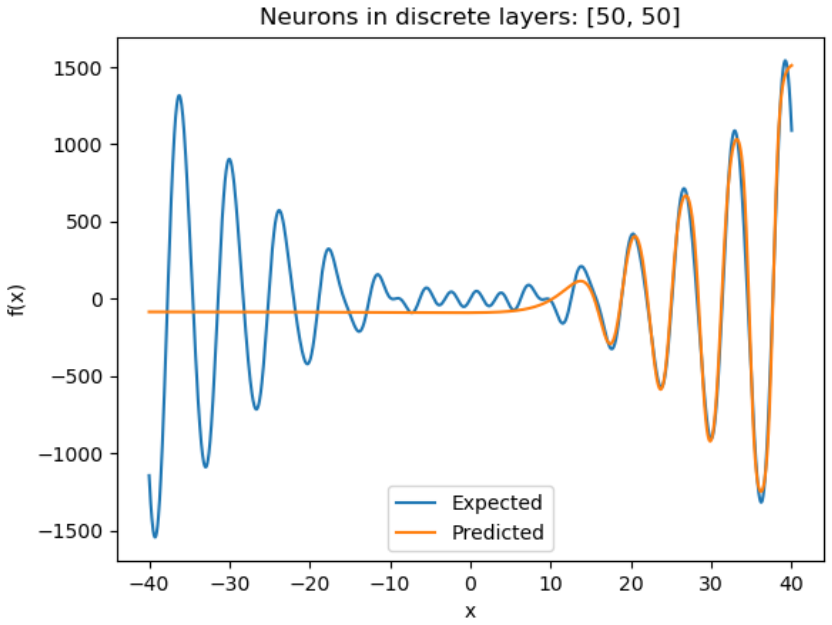
Już wynik pierwszego testu pokazał lepszej jakości rezultaty działania sieci. Jednak działanie naszego programu dla argumentów ujemnych wciąż pozostawiały wiele do życzenia.

Kolejne testy były jednak dużo bardziej obiecujące.

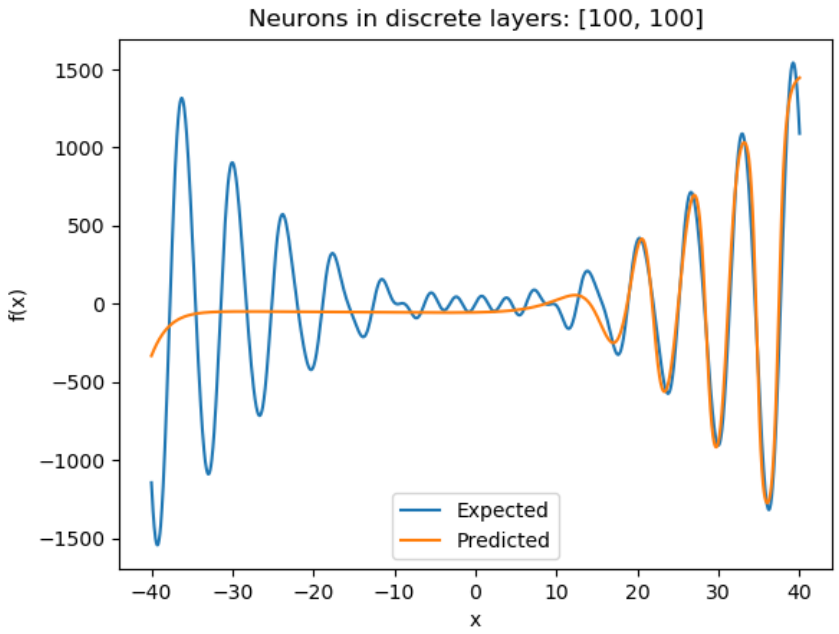




Przewidywane wyniki dla argumentów dodatnich coraz bardziej zbliżają się do oczekiwanych rezultatów dla zadanej funkcji. Jednak wyniki dla argumentów ujemnych wciąż przyjmują prawie postać liniową.

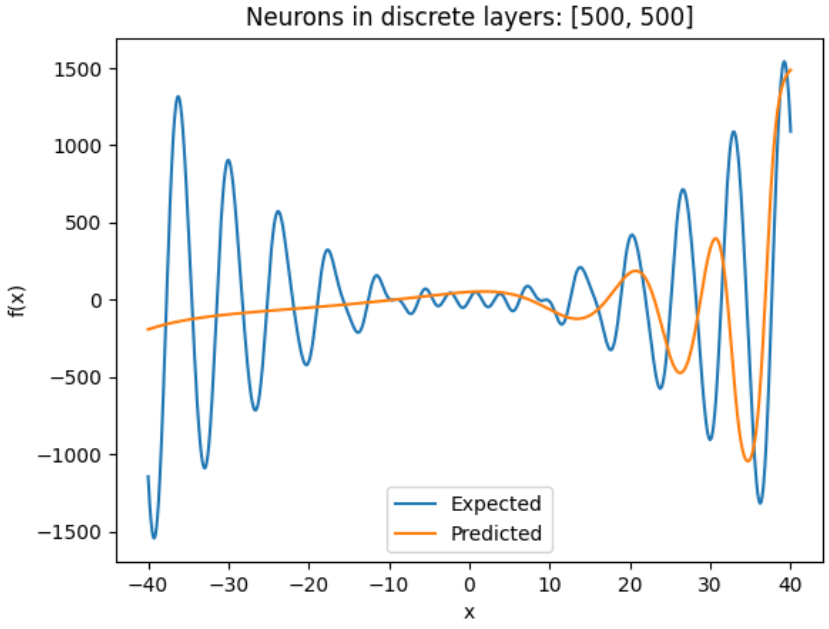


Kolejny test pokazał, że dla dodatnich argumentów kształt funkcji jeszcze bardziej zbliżył się do oczekiwanej postaci tracą jednak na dokładności.



Przy kolejnym teście rezultaty dla wartości dodatnich uległy pogorszeniu. Interesująca okazała się zmiana dla wartości ujemnych. Zwiększona ilość neuronów spowodowała nieznaczną ale jednak zmianę kształtu dla tej części funkcji co może sugerować że większa ilość neuronów wpłynąć może na działanie naszego programu dla argumentów ujemnych.

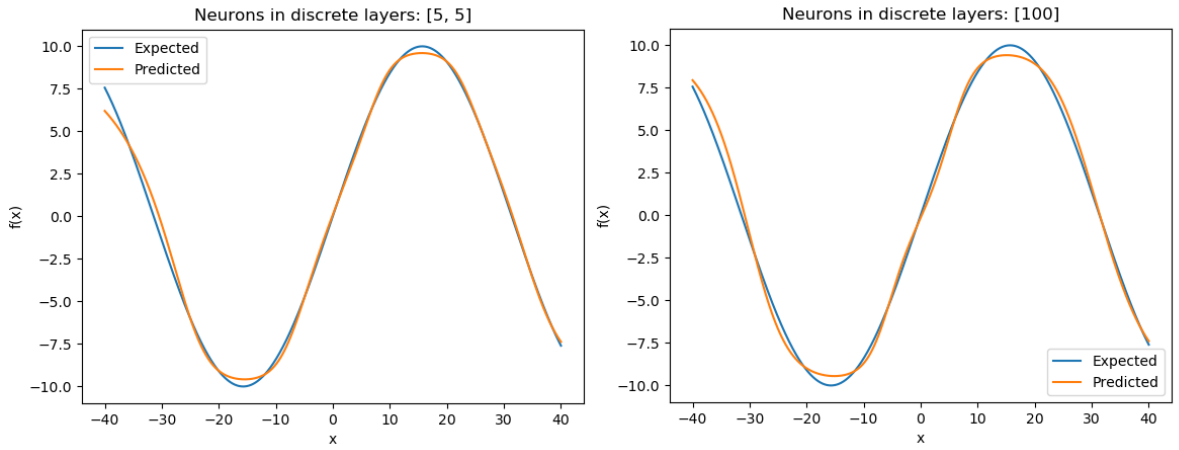
Niestety dodatkowe testy jednoznacznie pokazał że większa ilość neuronów wpłynęła wyłącznie na pogorszenie ogólnej jakości przewidywania



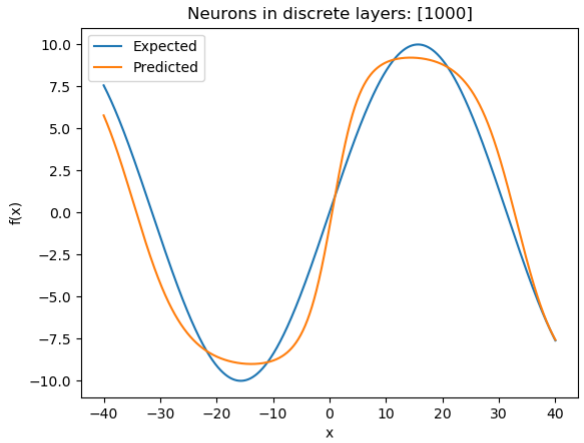
Zweryfikowanie działania programu

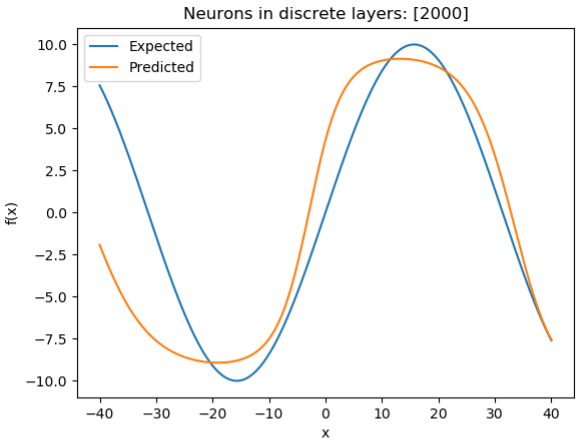
Rezultaty działania programu skłoniły nas do przetestowania go dla innej funkcji niż ta podana na zajęciach. Zdecydowaliśmy się na funkcję **10*sin(x/10)**.

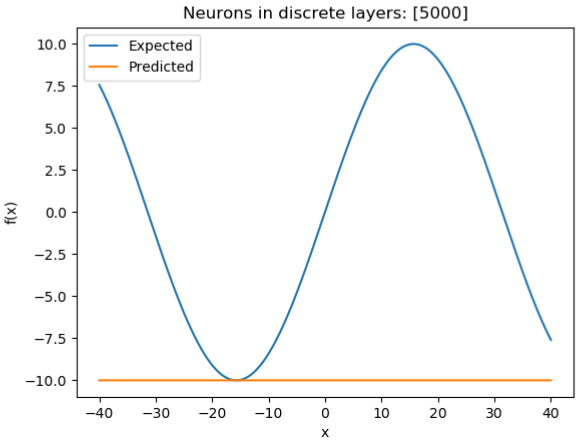
Podobnie jak wcześniej dla niewielkiej ilości neuronów wyniki okazały się najbardziej zbliżone do oczekiwanych rezultatów.



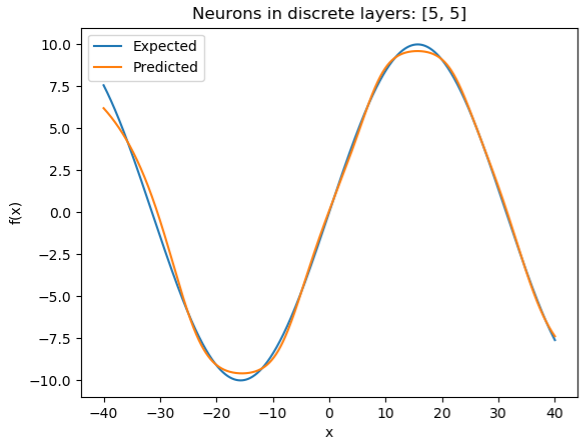
Zwiększenie ich ilości w warstwie doprowadziło do pogorszenia działania programu, a przy 5000 podobnie jak dla zadanej na zajęciach funkcji rezultaty przewidywane przyjmowały wartość stałą

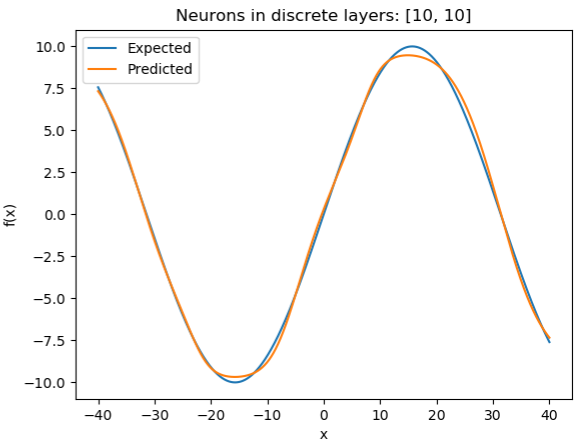






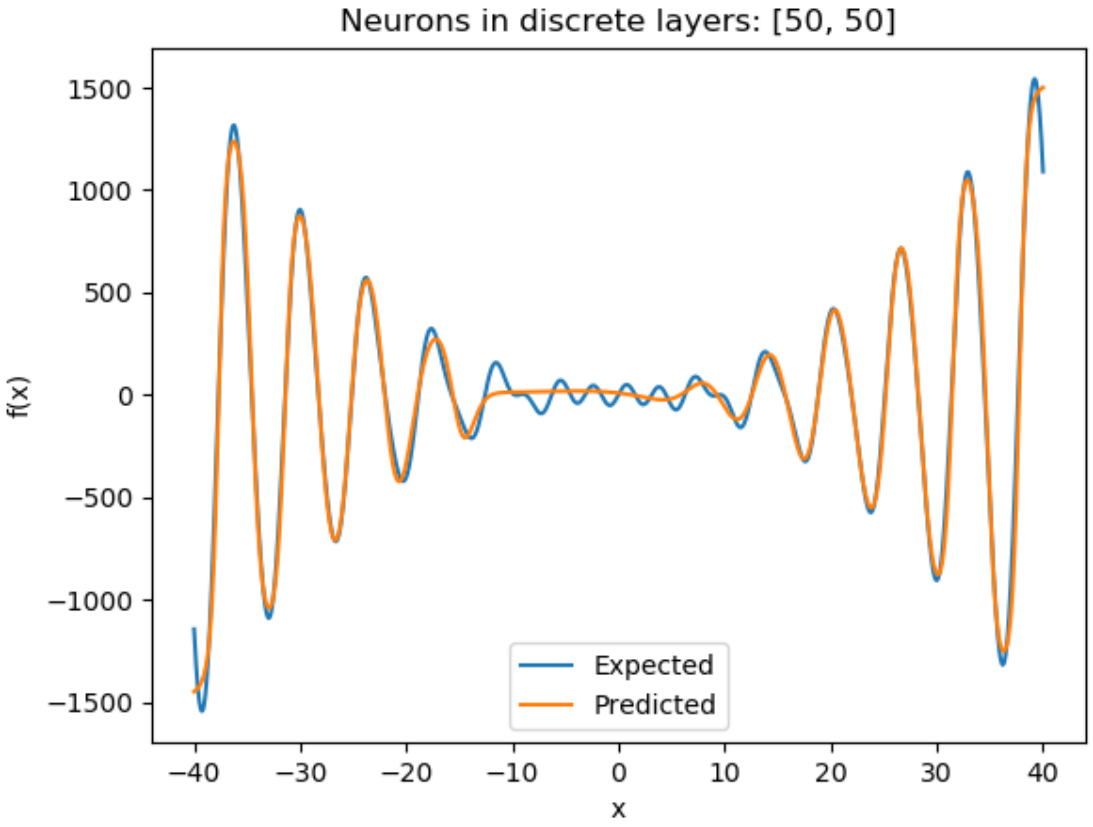
Podobnie dla dwóch warstw dyskretnych działanie naszego programu z dużą dokładnością przewidziało wartość określonej przez nas funkcji.





Test dla zwiększonej ilości epok

Dodatkowe testy pokazały nam, że problem przy przewidywaniu dla argumentów ujemnych może wynikać z czegoś innego niż błędna implementacja algorytmu. Jako dodatkowy test zbadaliśmy działanie naszego programu dla danych, które dały przy poprzednich testach najbardziej obiecujące rezultaty - dwie warstwy dyskretne po 50 neuronów tym razem jednak zwiększyliśmy liczbę epok dziesięciokrotnie do 100 000. Przewidywanie dla tych danych stało się wyjątkowo dokładne. Może to sugerować że pozostałe testy dla zwiększonej ilości epok również dałyby nam lepsze rezultaty. NIestety czas generowania dla tego testu wynosił aż cztery godziny, powtórzenie więc wcześniejszych testów dla nowej ilości epok mogłoby zająć zbyt dużą ilość czasu.



Wnioski

Ilość warstw

Nasze eksperymenty jednoznacznie pokazały że dwie warstwy dyskretne dają lepsze rezultaty. Eksperymenty dla jednej warstwy są niedokładne dla każdej testowanej przez nas ilości neuronów. Dla dwóch warstw dyskretnych przewidywane rezultaty dla argumentów dodatnich były bardzo zbliżone do rezultatów przewidywanych dla zadanej funkcji.

Ilość neuronów w warstwie

Przeprowadzone przez nas testy wyraźnie pokazały że ilość neuronów ma znaczny wpływ na przewidywanie wartości zadanej funkcji.
Dla jednej warstwy dyskretnej zauważalne jest, że dla niewielkiej ilości neuronów (50, 10) przewidywana wartość funkcji dla dodatnich argumentów jest bardziej zbliżona do oczekiwanych rezultatów niż w pozostałych testach. Dalsze zwiększanie ilości neuronów powoduje wyłącznie pogorszenie jakości przewidywania.

Wyniki na dwóch warstwach, które był dużo bardziej dokładne pozwoliły dodatkowo na ocenienie dla jakiej ilości neuronów otrzymujemy najdokładniejsze rezultaty. Wyniki testów jasno pokazały że zarówno małe ilości neuronów (10) jak ich większe ilości (100) osiągają gorsze wyniki niż ilość neuronów znajdujące się w tym zakresie (10;100). Wyniki testu dla 50 neuronów w warstwie dyskretnej osiągnęły najlepsze rezultaty w porównaniu z pozostałymi eksperymentami. Jednak i tu wynik naszego przewidywania dla argumentów ujemnych jest daleki od oczekiwanych rezultatów.

Dodatkowe testy pokazały jednak, że przy zwiększonej ilości epok efekty przewidywania stają się zbliżone do wartości oczekiwanych również dla argumentów ujemnych. Winnym temu faktowi pozostaje sposób implementacji naszego programu, który korekcje błędów zaczyna od argumentów z prawej strony osi. Dlatego też dla mniejszej ilości epok lewa strona wykresu nie zdąża się skorygować. Niestety długi czas oczekiwania dla testów przy zwiększonej ilości epok uniemożliwił nam efektywne przeprowadzenie dalszych testów dla tej ilości epok.