Projekt 3 Algorytmy Ewolucyjne

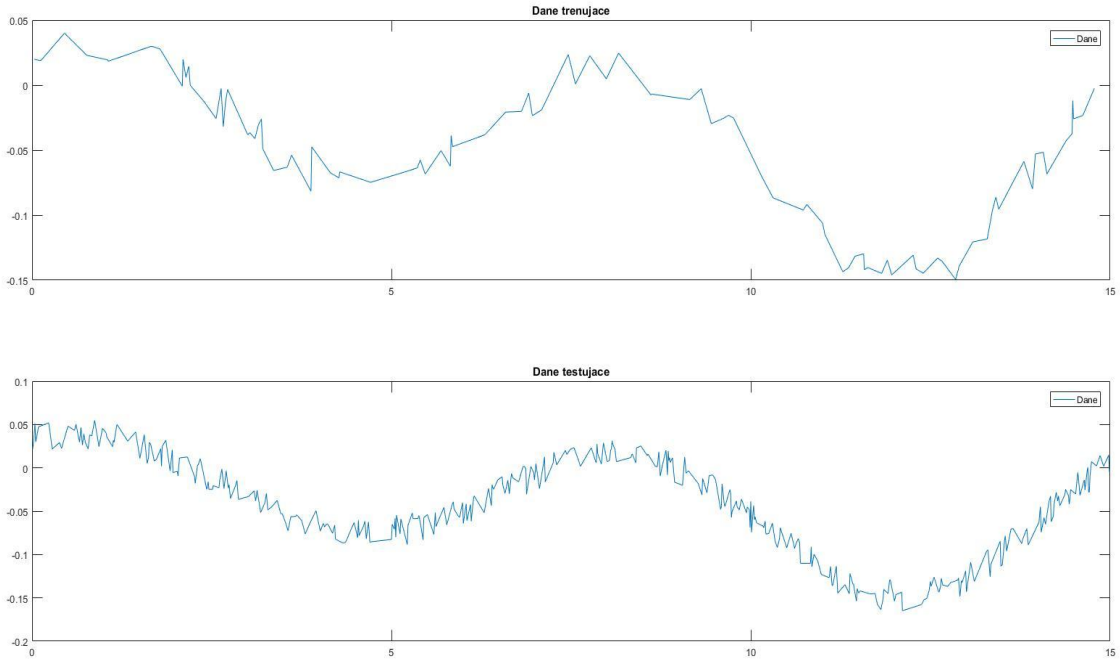
Autor: Bartłomiej Boczek

1. Cel projektu

Celem projektu jest zaprojektowanie sieci neuronowej o architekturze perceptronu wielowarstwowego do aproksymacji funkcji y = f(x) na podstawie zbioru uczącego.

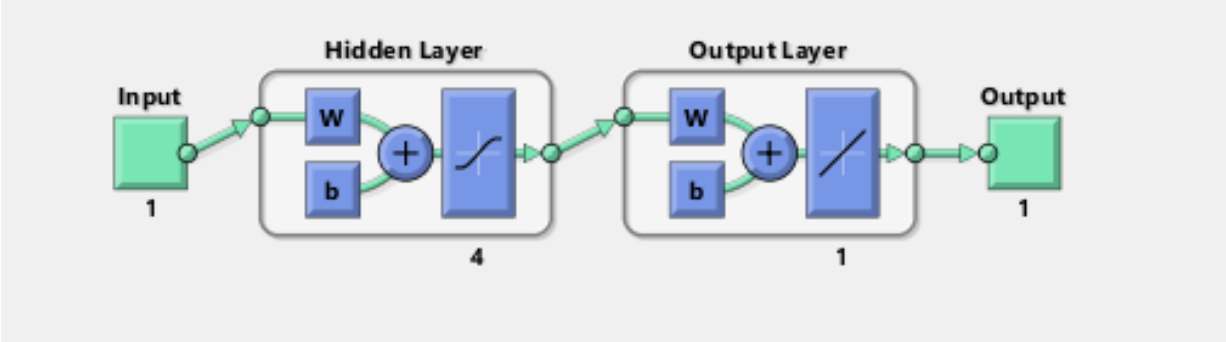
Do uczenia sieci zostanie użyty algorytm Levenberga-Marquardta 200 iteracji.

2. Wizualizacja danych



Jak widzimy na powyższych przebiegach, dane oscylują, są zaszumione. Może to generować problem w postaci nadmiernego dopasowania (overfittingu) modelu do danych przy zastosowaniu zbyt dużej liczby neuronów ukrytych.

3. Schemat sieci neuronowej – najlepsza struktura

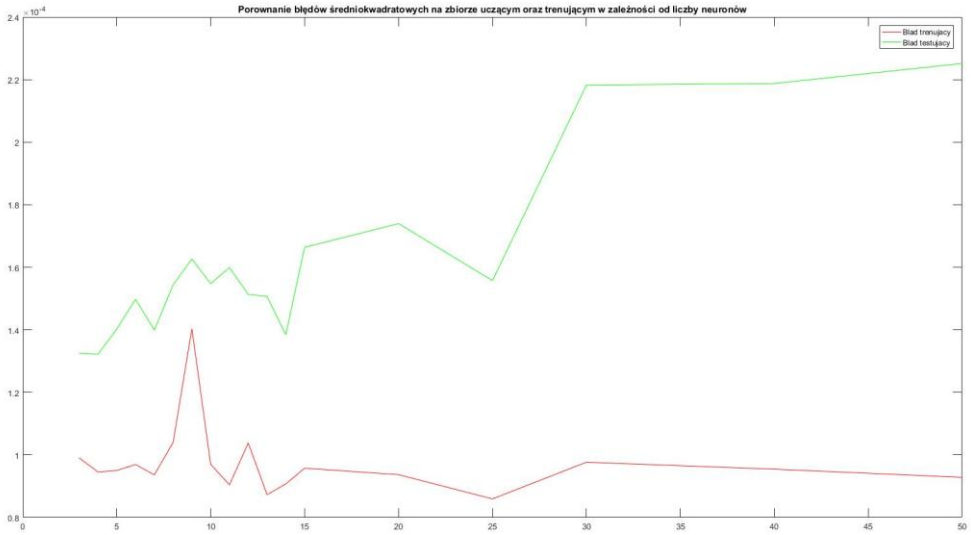


4. Wyniki obliczeń

Tabela błędów średniokwadratowych dla zbioru uczącego i testowego w zależności od liczby neuronów ukrytych.

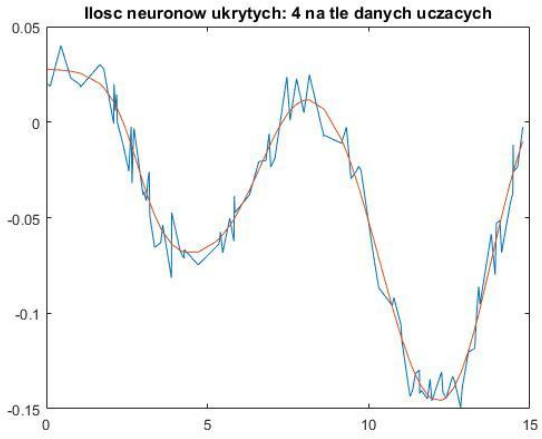
Liczba neuronów ukrytych	Błąd średniokwadratowy – zbiór uczący	Błąd średniokwadratowy – zbiór testowy
1	0.001763935026740	0.002148182626101
2	0.001058987731118	0.001334854378060
3	0.000099034606615	0.000132460324044
4	0.000094527069816	0.000132209418823
5	0.000095028104340	0.000140200183331
6	0.000096949390745	0.000149782880635
7	0.000093611197867	0.000139907141188
8	0.000103975405217	0.000154379490947
9	0.000140319432218	0.000162643698553
10	0.000097006862411	0.000154743427966
11	0.000090414193572	0.000159921462377
12	0.000103824206690	0.000151316735605
13	0.000087289474091	0.000150706466479
14	0.000090729167427	0.000138420034118
15	0.000095753809216	0.000166380811769
20	0.000093694393523	0.000173958696497
25	0.000085940593137	0.000155727699255
30	0.000097670564490	0.000218244634202
40	0.000095445010667	0.000218755239525
50	0.000092825825712	0.000225157586871

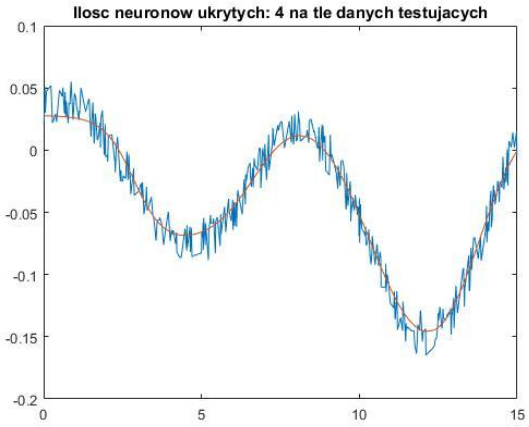
Powyższe dane na wykresie:

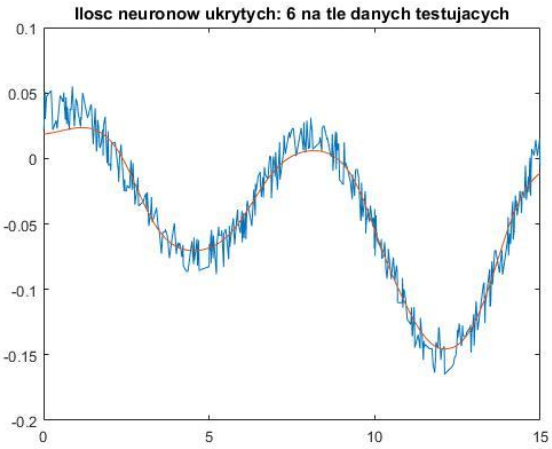


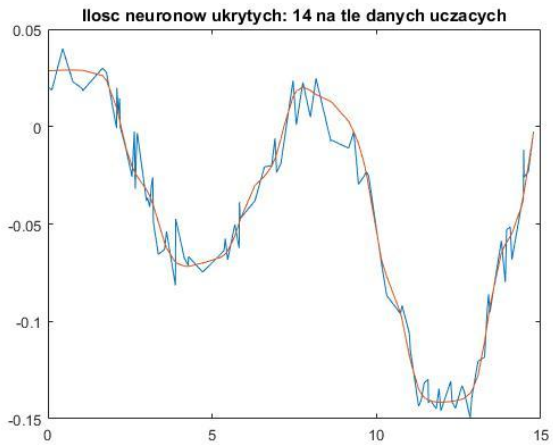
Na wykresie pominięte zostały wartości błędów dla 1 i 2 neuronów ukrytych, gdyż ich wartości są duże w porównaniu do reszty wyników.

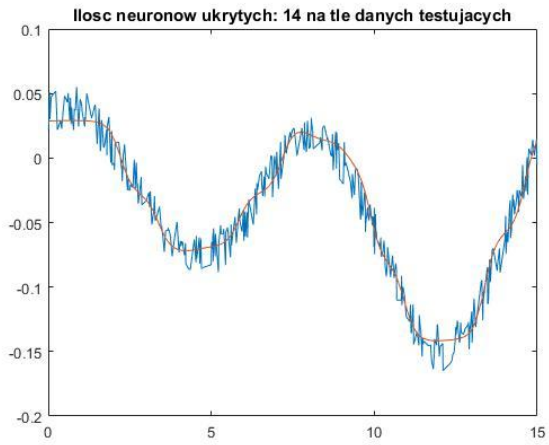
Na podstawie wyników możemy stwierdzić, że najlepszą siecią może być sieć dla 4, 6 lub 14 neuronów. Przeanalizujmy dla tych wartości wykresy funkcji aproksymujących:







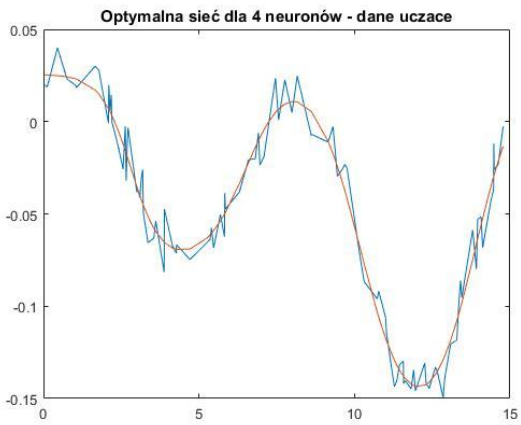


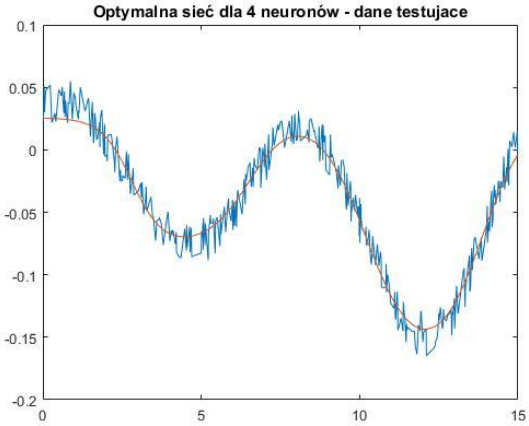


Jak możemy stwierdzić po wykresach aproksymacje dla 4 i 6 neuronów są bardzo podobne, 4 delikatnie lepsza, natomiast dla 14 występuje już overfitting, dlatego opcję tą odrzucam.

5. Wykres aproksymowanej funkcji.

Najlepsza sieć jaką udało się uzyskać dla 4 neuronów ukrytych po 20 iteracjach dla generowanych losowa wag początkowych z zakresu [-0.15 0.15]:



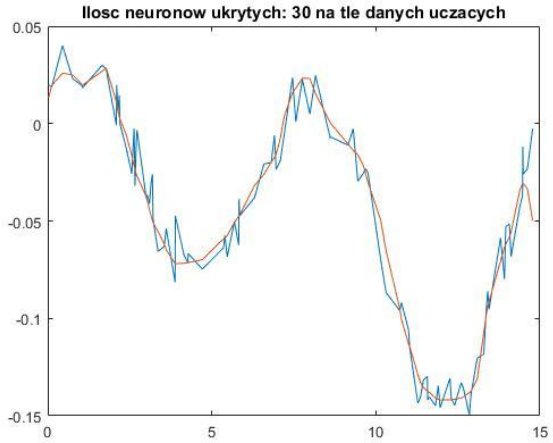


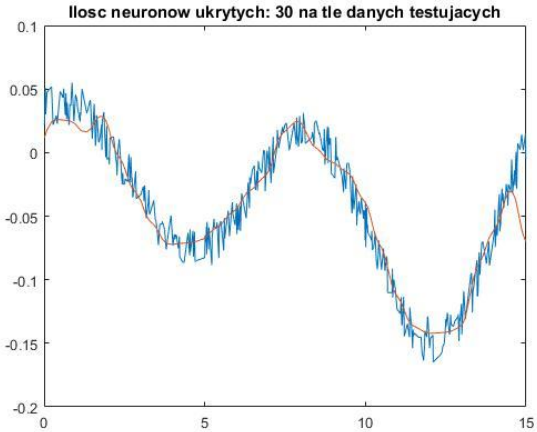
6. Komentarze

a) Do realizacji projektu zostało użyte środowisko Matlab oraz pakiet Neural Network Toolbox. Zawiera on funkcje pokrywające zarówno tworzenie modeli sieci neuronowych, jak i ich trenowanie wieloma znanymi metodami, a także metody pomocnicze takie jak liczenie błędów czy określanie parametrów eksperymentu tj. określenie liczby iteracji czy docelowy błąd. Dzięki temu możemy w prosty, wysoko-poziomowy sposób przeprowadzić eksperymenty i wytrenować kilka sieci testowych w krótkim czasie.

Podczas pracy bardzo pomagają możliwości środowiska Matlab, które umożliwia rysowanie wykresów czy operacje na macierzach.

b) Liczba neuronów ukrytych jaka okazała się optymalna przy modelowaniu danych trenujących to 4. Liczba ta została wybrana na podstawie obserwacji wielkości błędów na zbiorach trenującym i testującym. W momencie, gdy wzrasta błąd modelu na zbiorze testującym, pomimo malenia błędu na zbiorze trenującym możemy wnioskować, że mamy do czynienia ze zjawiskiem overfittingu, czyli zbytniego dopasowania modelu do danych, co możemy zobaczyć na poniższym wykresie (sieć o 30 neuronach ukrytych).





Widać, że model próbuje się dopasować do niektórych odstających od normy danych, co nie jest pożądanym zjawiskiem w kontekście modelowania.

8. Wnioski

Liczba neuronów ukrytych jaką należy wybrać to liczba dla której jest minimalny błąd na zbiorze testującym.

Oczywiście, można by skorzystać z bardziej wysublimowanych metod weryfikacji takich jak cross-validation lub leave-one-out, by uzyskać dokładniejsze rezultaty, jednak nie są one przedmiotem tego projektu.