

Wydział Elektroniki i Technik Informacyjnych  
Sieci Neuronowe i Neurokomputery

Projekt A2

Aproksymacja funkcji przy użyciu sieci neuronowej

Wykonał:

Paweł Gajewski

Warszawa 2019

## Spis treści

|  |   |
|--|---|
| 1) Cel projektu .....  | 3 |
| 2) Opis i wizualizacja danych .....  | 3 |
| 3) Algorytm uczenia .....  | 4 |
| 4) Wybór optymalnego modelu .....  | 4 |
| a) Dobór liczby neuronów .....   | 4 |
| b) Dobór liczby neuronów ukrytych metodą porównania błędu średniokwadratowego<br>aproxymacji na zbiorze uczącym i na zbiorze testowym po zakończeniu uczenia ..... | 5 |
| c) Dobór liczby neuronów ukrytych metodą virtual leave-one-out. ....   | 6 |
| d) Optymalna ilość neuronów ukrytych. ....   | 7 |
| 5) Parametry, wskaźniki i wyniki wybranej sieci neuronowej .....   | 8 |
| 6) Wnioski .....   | 9 |
| 7) Załączniki .....  | 9 |

## 1) Cel projektu

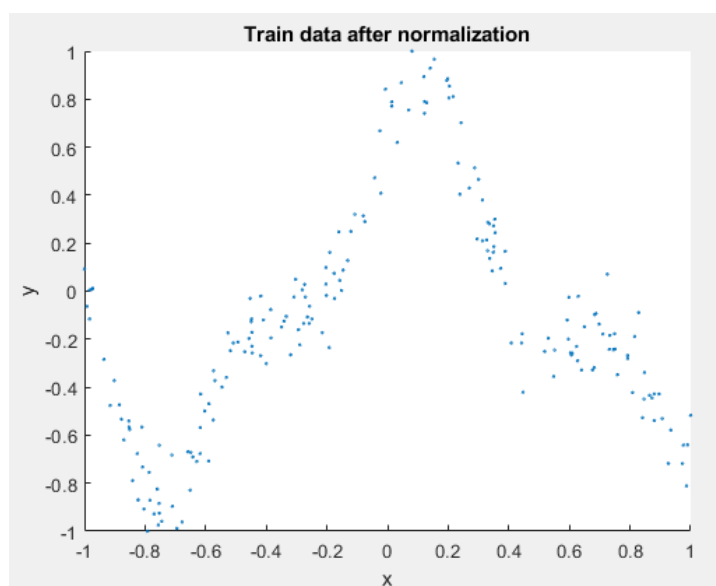
Celem projektu jest aproksymacja jednowymiarowej funkcji przy użyciu sieci neuronowej zoptymalizowanej pod kątem dostarczonych danych. Jako optymalizację rozumie się

## 2) Opis i wizualizacja danych

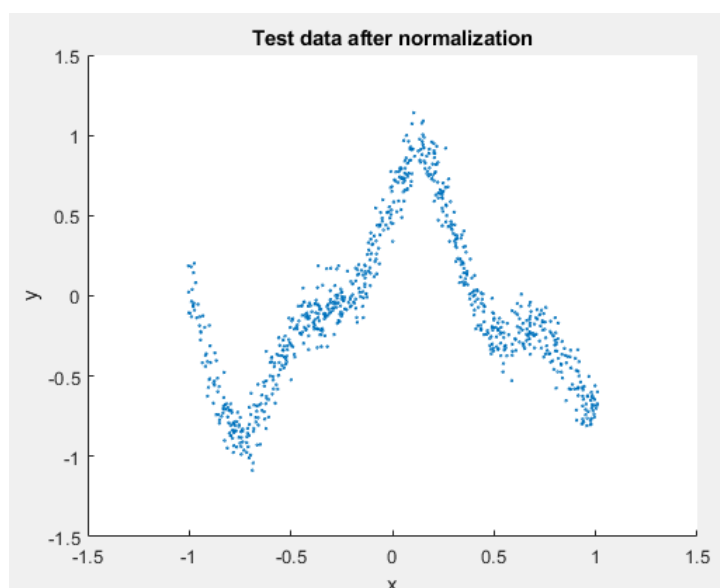
Otrzymane dane przedstawiające przebieg funkcji podzielono na dwa zbiory: uczący oraz testowy. Każdy zbiór składa się z dwóch wartości –  $x$  (wartość wejściowa) oraz  $y$  (wartość wyjściowa).

- Zbiór uczący: 200 punktów
- Zbiór testowy: 800 punktów

Dokonano normalizacji oraz centrowania danych na podstawie wartości zbioru uczącego. Wyniki operacji graficznie przedstawiono na wykresach poniżej.



Rysunek 1 Dane uczące po normalizacji



Rysunek 2 Dane testowe po normalizacji

Poprzez obserwację znormalizowanych zbiorów danych można zauważyć, że:

- Aproksymowana funkcja może być z zadowalającą dokładnością przybliżona przez funkcję prostoliniową łamaną składającą się z 5-6 odcinków. W związku z tym warstwa ukryta sieci neuronowej niezbędnej do realizacji podobnej klasy aproksymacji będzie zawierać minimum 6-7 neuronów.
- Większość punktów skupia się wokół „konturu” aproksymowanej funkcji. Oznacza to, że najprawdopodobniej punkty zostały wygenerowane/zmierzone poprzez dodanie do wartości wyjścia czynnika losowego, np. szumu pomiarowego. Brak jest widocznych tzw. błędów grubych, które trzeba by było odrzucić przed rozpoczęciem wybierania struktury sieci neuronowej.

### 3) Algorytm uczenia

Projekt zaimplementowano w środowisku Matlab 2018b. Utworzono sieć neuronową o jednej warstwie ukrytej, składającej się z neuronów o funkcji aktywacji tangens hiperboliczny (*tansig*) oraz o jednym neuronie wyjściowym o liniowej funkcji aktywacji (*purelin*). Każdy z neuronów posiada dodatkowe wejścia typu bias. Wagi wejść na każdego neuronu sieci zostały zainicjowane wartościami losowymi z przedziału  $(-0.15, 0.15)$  lub przy użyciu domyślnego algorytmu Nguyena-Widrowa. Każda sieć uczona była (zgodnie z wytycznymi zadania) dwuetapowo:

- i) Przez 100 epok przy użyciu algorytmu wstecznej propagacji błędu BP (*traingd*),
- ii) Przez 200 epok przy użyciu algorytmu Levenberga-Marquardta,

Stwierdzono, że założona liczba epok dla każdego etapu jest wystarczająca, ponieważ wydłużenie ich ilości dla kilku zestawów losowych rozmiarów sieci oraz wag początkowych nie prowadziło do znacznego zmniejszenia błędu średniokwadratowego.

Nałożono również dodatkowy warunek na zatrzymanie procesu uczenia w drugim etapie – wartość błędu średniokwadratowego dla poniżej 0.01.

Zbiór uczący, w trakcie uczenia przy użyciu funkcji programu Matlab, dzielony jest na trzy podzbiory. Elementy do każdego z nich przydzielane są przy użyciu określonej funkcji (domyślnie jest to losowy wybór elementów ze zbioru), ale liczność zbiorów jest ściśle określona. Są to:

- Zbiór wykorzystywany do uczenia – wyliczania gradientu oraz modyfikacji wag oraz biasów (70%).
- Zbiór wykorzystywany do walidacji – wyliczania wskaźnika jakości w trakcie procesu uczenia oraz zapobiegają przeuczeniu sieci (15%).
- Zbiór wykorzystywany do testowania – elementy niewykorzystywane bezpośrednio w trakcie uczenia sieci, ale pozwalający bezpośrednio porównywać poszczególne sieci pomiędzy sobą (15%).

Ponieważ w treści zadania wskazano zbiór danych mających służyć do uczenia sieci, zmieniono domyślne proporcje rozmiarów podzbiorów (80%, 20%, 0), aby w cyklu uczenia wykorzystywać więcej dostępnych punktów uczących.

### 4) Wybór optymalnego modelu

#### a) Dobór liczby neuronów

W celu dobrania liczby neuronów ukrytych posłużono się dwiema metodami i porównano ich wyniki. Oby dwa eksperymenty przeprowadzane były jednak w podobny sposób.

Na podstawie pierwszego etapu eksperymentu (dobór poprzez porównanie błędu średniokwadratowego) stwierdzono, że inicjalizacja przy użyciu wartości losowych z przedziału  $(-0.15, 0.15)$  daje lepsze wyniki niż przy domyślnej inicjalizacji, więc drugą część eksperymentu przeprowadzono już wyłącznie dla sieci neuronowych inicjalizowanych przy użyciu losowych wag.

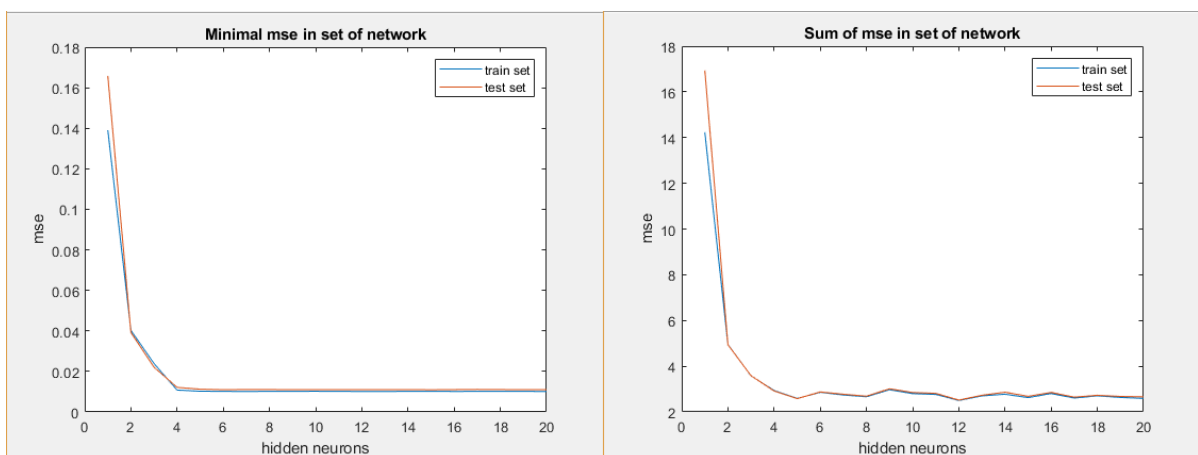
b) Dobór liczby neuronów ukrytych metodą porównania błędu średniokwadratowego aproksymacji na zbiorze uczącym i na zbiorze testowym po zakończeniu uczenia

Testy przeprowadzono dla dwudziestu zestawu stu sieci neuronowych o licznie neuronów ukrytych z zakresu  $[1:20]$ . W ramach jednego rozmiaru warstwy ukrytej tworzono 100 sieci o różnych wagach, dla których obliczono błąd średniokwadratowy. Zapamiętano sieć o najmniejszym błędzie średniokwadratowym z każdego zestawu oraz zsumowano błędy średniokwadratowe wszystkich sieci o danym rozmiarze, aby móc wywnioskować pewne reguły dotyczące ilości neuronów ukrytych.

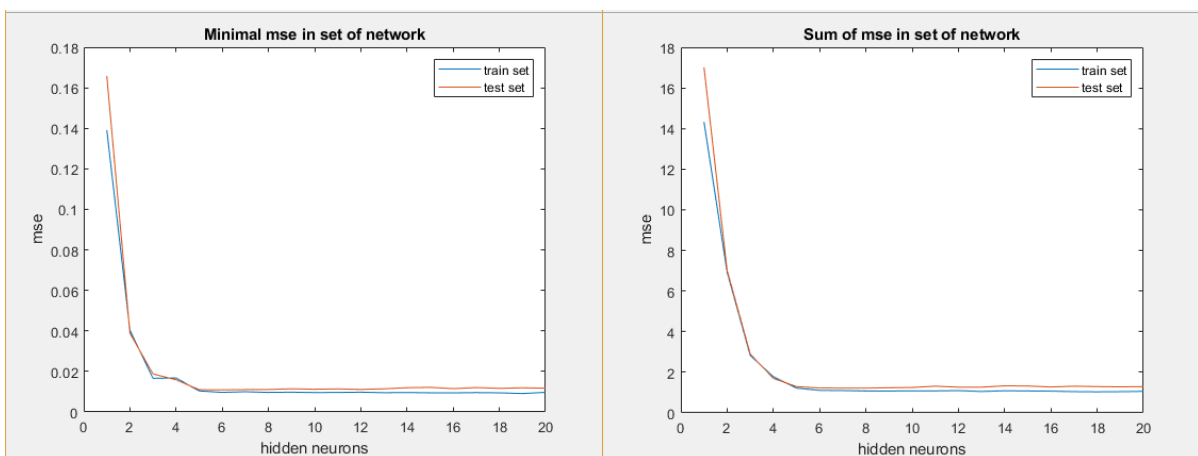
Eksperyment powtórzono dla inicjalizacji wartościami losowymi oraz przy użyciu algorytmu Nguyena-Widrowa.

Wyniki przedstawiono na wykresach poniżej.

Wagi losowe z przedziału  $(-0.15, 0.15)$



Inicjalizacja przy użyciu algorytmu Nguyena-Widrowa.



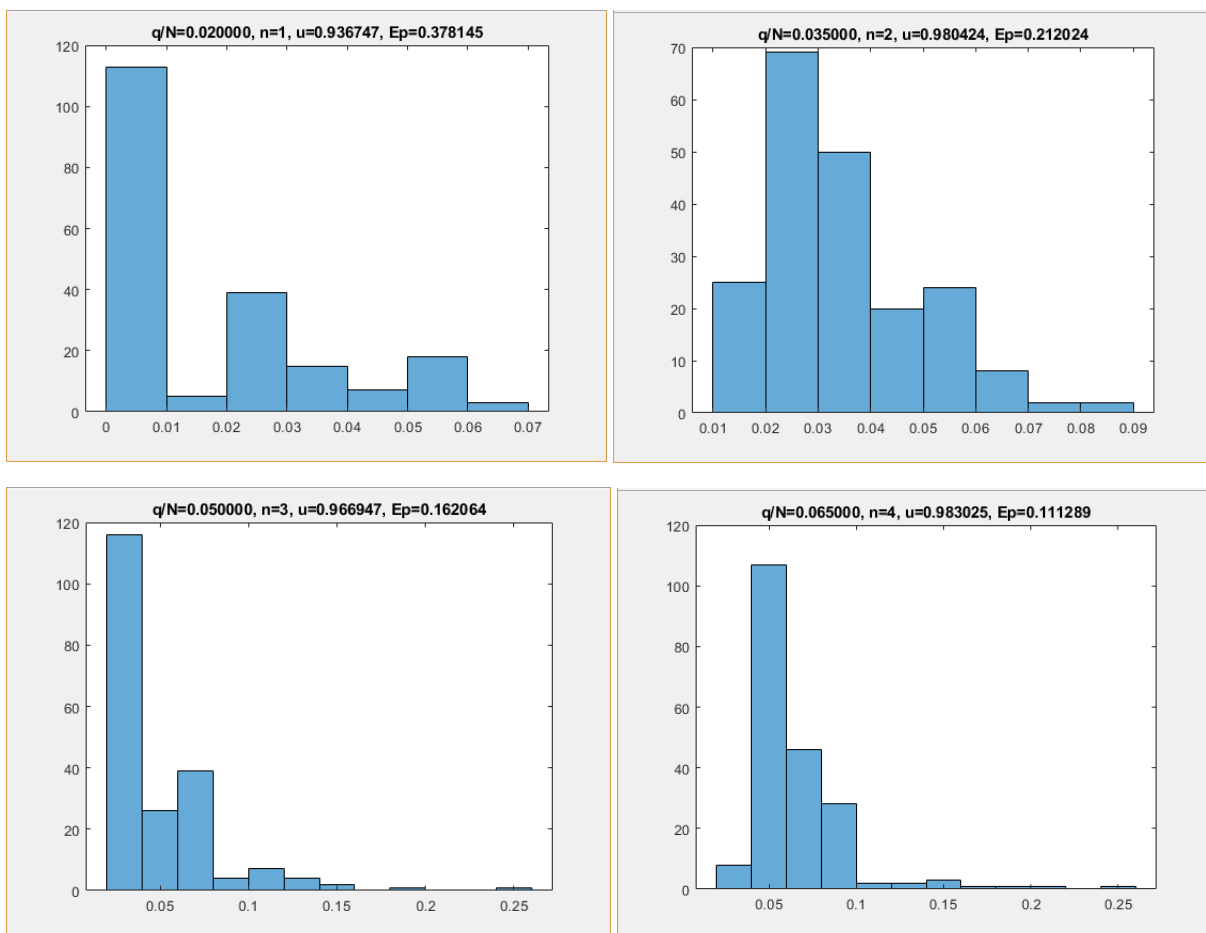
## Wnioski:

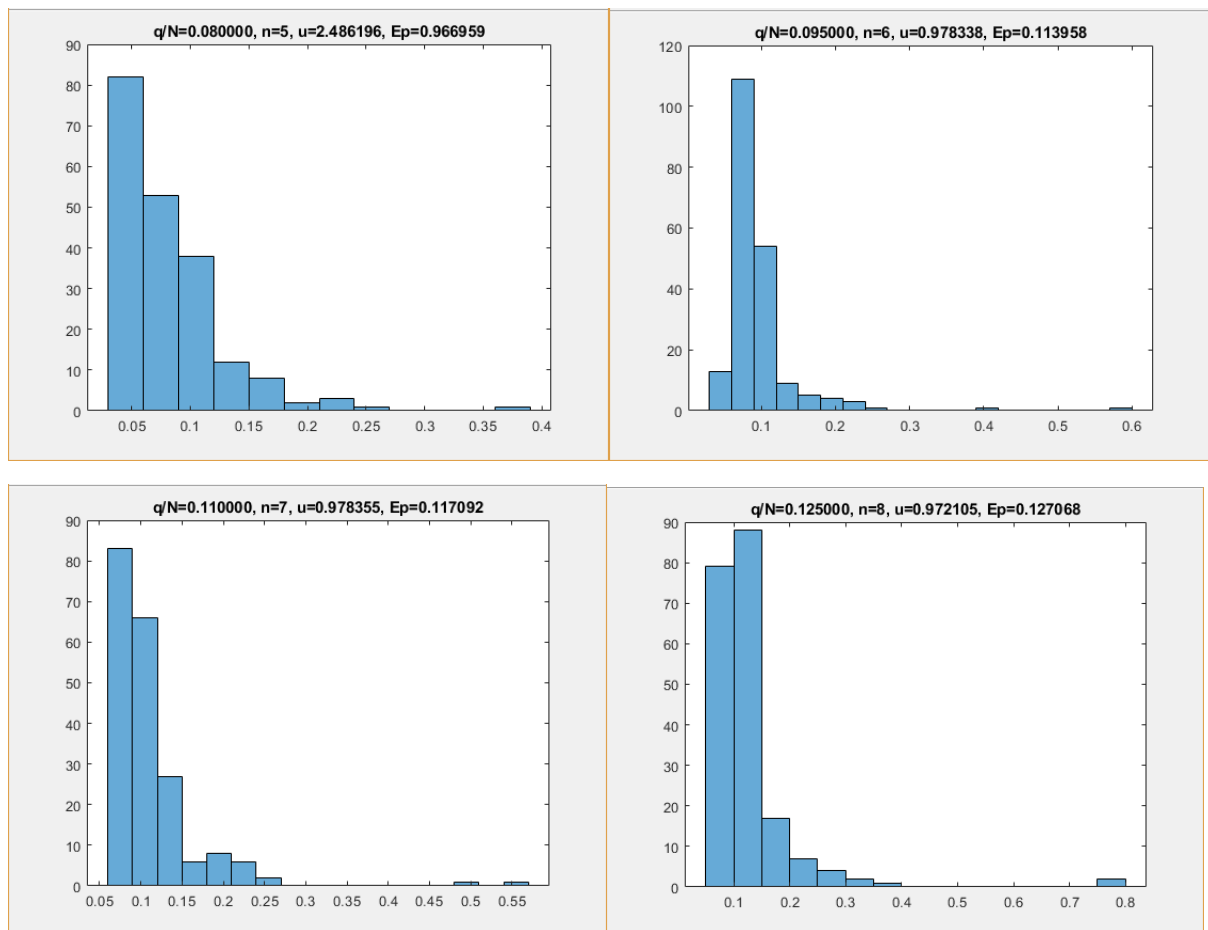
- Widać, że rozbieżność wartości błędów dla sieci zainicjalizowanej przy użyciu losowych wskaźników dla zbiorów uczącego oraz testowego jest mniejsza niż w przypadku użycia algorytmu Nguyena-Widrowa, zarówno w przypadku najlepszej sieci (o minimalnym błędzie) jak i ogólnym przypadku (zsumowanych błędach). Widać, że losowa inicjalizacja zwiększa tolerancję sieci na przeuczenie i zwiększa zdolność generalizacji. Dlatego ta metoda inicjalizacji będzie obowiązywała w kolejnym etapie eksperymentu.
- Sieci osiągnęły zadowalające poziomy błędów średniokwadratowych już przy 4 neuronach ukrytych. Dalsze zwiększanie ilości neuronów ukrytych albo zwiększało błąd albo nie zmniejszało go z znaczący sposób.

### c) Dobór liczby neuronów ukrytych metodą virtual leave-one-out.

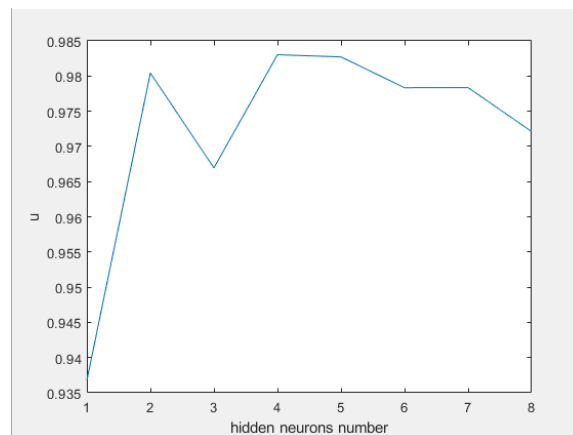
Dla sieci neuronowych o poszczególnych rozmiarach najlepiej minimalizujących błąd średniokwadratowy zbadano również wartości dźwigni (histogramy przedstawiono poniżej). Co ciekawe, dla wag o losowych wartościach, jacobian Z jest pełnego rzędu wyłącznie na rozmiarów warstwy ukrytej od 1 do 8 (testowane przy wielu różnych symulacjach), natomiast przy inicjalizacji algorytmem Nguyena-Widrowa – dla wszystkich testowanych rozmiarów.

Histogramy wag poszczególnych sieci neuronowych wraz z charakterystycznymi parametrami.





Histogramy w ogólności skupiają się wokół wartości  $q/N$  dla odpowiednich rozmiarów. Zbadano więc również zależność współczynnika  $\mu$  od ilości neuronów ukrytych.



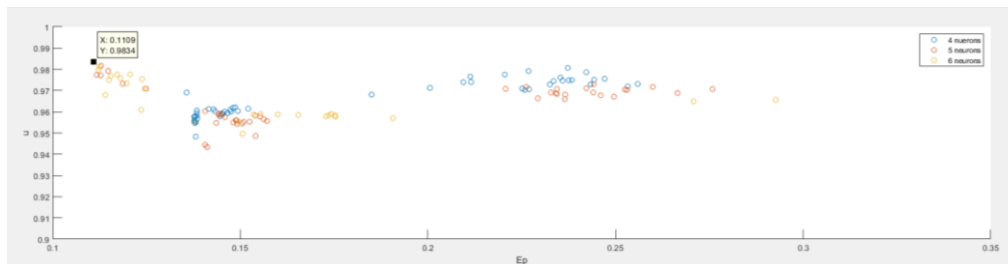
Rysunek 3 Przebieg wartości  $\mu$  w zależności od ilości neuronów ukrytych

Ponieważ do porównania wybrano sieci o najmniejszych wyznaczonych MSE, rozbieżność wartości współczynnika  $\mu$  wynosi ok. 0.06. Na wykresie widać jednak, że współczynnik  $\mu$  osiąga największe wartości na sieci o 4, 5 neuronach ukrytych, a następnie maleje wraz ze zwiększaniem ilości neuronów ukrytych.

#### d) Optymalna ilość neuronów ukrytych.

Analizując wyniki otrzymane w poprzednich etapach eksperymentu stwierdzono, że model ostateczny zostanie wyznaczony na podstawie dokładnego doboru wartości  $E_p$  spośród 150 sieci neuronowych z

4,5, oraz 6 neuronami ukrytymi (po 50 dla każdej konfiguracji). Spośród wyuczonych sieci, jacobiany aż 28 okazały się niepełnego rzędu, więc nie umieszczono ich na wykresie poniżej.

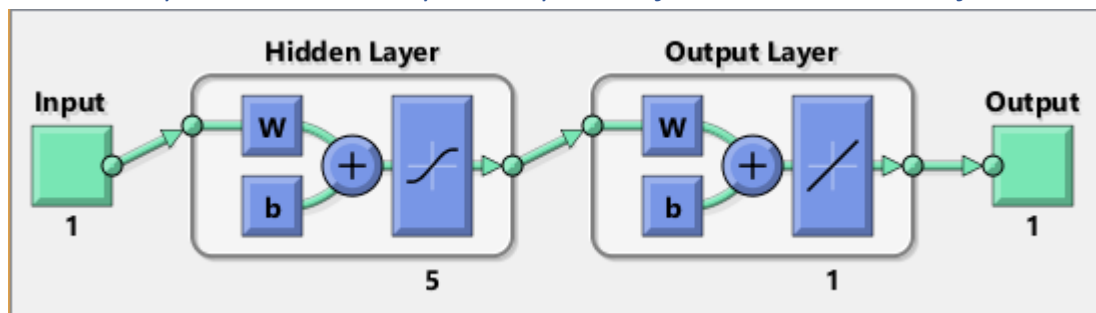


Rysunek 4 Wartości  $E_p$  oraz  $u$  dla wytrenowanych sieci neuronowych

Otrzymane wartości w trakcie testu umieszczono w tablicy 1.

Najlepsze wyniki osiągnęła grupa sieci 5 oraz 6 neuronowych skupiona w lewym górnym rogu wykresu. Spośród nich wybrano sieć z pięcioma neuronami ukrytymi zaznaczoną na wykresie.

## 5) Parametry, wskaźniki i wyniki wybranej sieci neuronowej

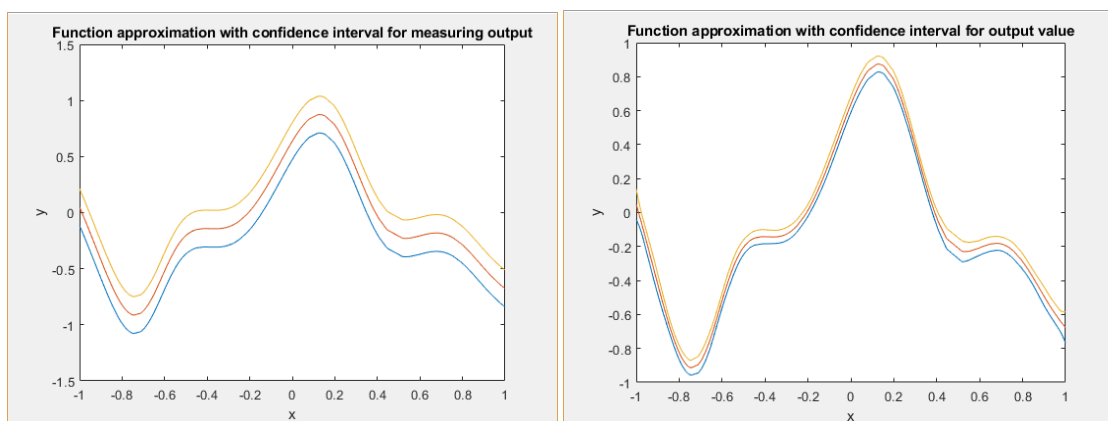


Rysunek 5 Widok wybranej sieci neuronowej

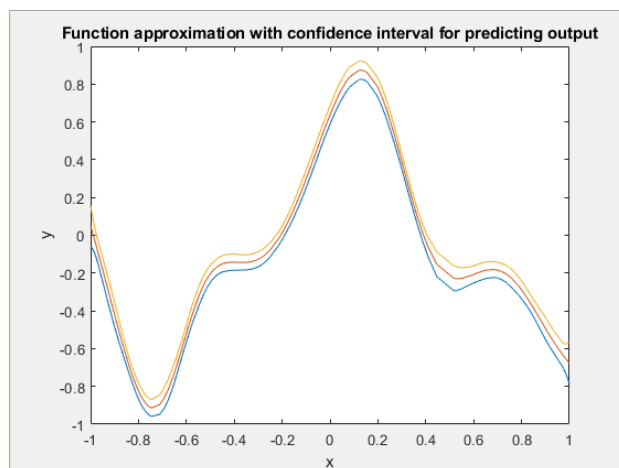
Wybrana sieć charakteryzuje się najlepszymi wartościami wskaźników spośród wszystkich wyników ( $E_p = 0,110851$  oraz  $\mu = 0,983442$ ).

Wybraną sieć przeanalizowano używając danych testowych, dla których uzyskano błąd średniokwadratowy równy 0.0116, co daje bardzo dobry wynik bliski wartości  $E_p$ . Poniżej przedstawiono wykres aproksymowanej funkcji wraz z przedziałem ufności dla  $(1-\alpha) = 99$ .

Przebieg aproksymowanej funkcji wraz z przedziałami ufności (kolejno) pomiaru wyjścia, wielkości wyjściowej oraz predykcji wyjścia.







Parametry wybranej sieci zawarto w tabeli 2.

## 6) Wnioski

- Dobór funkcji inicjalizacji funkcji ma większe znaczenie niż początkowo mogło się wydawać. Jakobiany sieci uczonych przy inicjalizacji algorytmem Nguyena-Widrowa były lepiej uwarunkowane, ale za to sieć inicjalizowana losowymi wartościami miała lepsze właściwości generalizacji.
- Sieć neuronowa wymagała mniej neuronów ukrytych niż początkowo zakładano na podstawie wstępnej analizy zbioru uczącego.
- Dzięki użyciu GPU komputera, nawet tak duża ilość sieci neuronowych uczyła się zdumiewająco szybko (opcja 'useGPU',='yes').
- Udało się dobrać sieć jednoznacznie najlepszą spośród otrzymanych sieci (nie trzeba było znajdować kompromisu pomiędzy  $E_p$  oraz  $\mu$ ).

## 7) Załączniki

| L.p | 4 neurony |          | 5 neuronów |          | 6 neuronów |          |
|-----|-----------|----------|------------|----------|------------|----------|
|     | u         | Ep       | u          | Ep       | u          | Ep       |
| 1   | 0,244327  | 0,975063 | 0,155307   | 0,957709 | 0,123876   | 0,975358 |
| 2   | 0,149367  | 0,960328 | 0,236616   | 0,96814  | 0,150608   | 0,949612 |
| 3   | 0,200587  | 0,971246 | 0,220723   | 0,970815 | 0,112184   | 0,979724 |
| 4   | 0,237294  | 0,980649 | Brak       |          | Brak       |          |
| 5   | 0,225156  | 0,970956 | 0,234122   | 0,968714 | Brak       |          |
| 6   | 0,211302  | 0,976582 | 0,111755   | 0,977349 | Brak       |          |
| 7   | 0,185017  | 0,968113 | 0,144795   | 0,958613 | Brak       |          |
| 8   | 0,247103  | 0,975523 | 0,229309   | 0,966336 | 0,17428    | 0,958957 |
| 9   | 0,238387  | 0,974926 | 0,144231   | 0,958586 | 0,292702   | 0,965613 |
| 10  | 0,145856  | 0,960027 | 0,234438   | 0,968409 | 0,120673   | 0,977584 |
| 11  | 0,235894  | 0,97468  | 0,14816    | 0,954969 | Brak       |          |
| 12  | 0,138025  | 0,955216 | 0,150947   | 0,9552   | 0,124561   | 0,970869 |
| 13  | 0,135727  | 0,96904  | Brak       |          | 0,115481   | 0,976757 |
| 14  | 0,137789  | 0,956004 | Brak       |          | 0,190707   | 0,956986 |
| 15  | 0,141595  | 0,961296 | 0,144365   | 0,959211 | 0,119738   | 0,973354 |

|    |          |          |          |          |          |          |
|----|----------|----------|----------|----------|----------|----------|
| 16 | 0,144632 | 0,957849 | 0,244088 | 0,969118 | Brak     |          |
| 17 | 0,148087 | 0,961865 | 0,242252 | 0,971081 | 0,270811 | 0,964893 |
| 18 | 0,146614 | 0,95937  | 0,226228 | 0,971678 | 0,114053 | 0,967901 |
| 19 | 0,232536 | 0,972829 | 0,112944 | 0,981716 | Brak     |          |
| 20 | 0,137738 | 0,95772  | 0,1525   | 0,9553   | Brak     |          |
| 21 | 0,147324 | 0,960942 | 0,141179 | 0,94335  | Brak     |          |
| 22 | 0,211521 | 0,974006 | 0,157149 | 0,955638 | 0,175256 | 0,958349 |
| 23 | 0,225857 | 0,970269 | 0,112824 | 0,977095 | 0,112473 | 0,981159 |
| 24 | 0,145404 | 0,958986 | 0,150506 | 0,954498 | Brak     |          |
| 25 | 0,137728 | 0,955344 | 0,253113 | 0,970286 | 0,117182 | 0,977586 |
| 26 | 0,138158 | 0,955041 | 0,148834 | 0,955931 | 0,155392 | 0,958926 |
| 27 | 0,145155 | 0,959174 | 0,249626 | 0,967115 | Brak     |          |
| 28 | 0,253265 | 0,971968 | Brak     |          | 0,17288  | 0,957892 |
| 29 | 0,138412 | 0,959308 | 0,246053 | 0,96772  | 0,123616 | 0,960889 |
| 30 | 0,152126 | 0,961441 | 0,234405 | 0,970694 | Brak     |          |
| 31 | 0,147586 | 0,960059 | 0,25996  | 0,971702 | Brak     |          |
| 32 | 0,138397 | 0,957956 | 0,140639 | 0,944398 | Brak     |          |
| 33 | 0,235333 | 0,976107 | 0,275845 | 0,970681 | Brak     |          |
| 34 | 0,220512 | 0,977519 | 0,149158 | 0,954262 | 0,165487 | 0,958452 |
| 35 | 0,226819 | 0,979167 | 0,244206 | 0,972946 | Brak     |          |
| 36 | 0,138141 | 0,948288 | 0,118675 | 0,973197 | 0,149631 | 0,955328 |
| 37 | 0,237567 | 0,974814 | 0,154121 | 0,948601 | Brak     |          |
| 38 | 0,14867  | 0,962076 | 0,140642 | 0,960245 | Brak     |          |
| 39 | 0,138592 | 0,956817 | 0,114792 | 0,979212 | Brak     |          |
| 40 | 0,137948 | 0,957659 | Brak     |          | 0,137929 | 0,955252 |
| 41 | 0,209473 | 0,973915 | 0,26654  | 0,968794 | Brak     |          |
| 42 | 0,138444 | 0,960555 | 0,156282 | 0,956494 | Brak     |          |
| 43 | 0,233499 | 0,974425 | 0,154029 | 0,958194 | 0,115038 | 0,974918 |
| 44 | 0,255942 | 0,973114 | 0,110851 | 0,983442 | 0,173532 | 0,958327 |
| 45 | 0,143267 | 0,960253 | 0,124941 | 0,970885 | 0,160006 | 0,958738 |
| 46 | 0,242227 | 0,978614 | 0,143611 | 0,954766 | Brak     |          |
| 47 | 0,226977 | 0,970616 | 0,236487 | 0,965947 | 0,175369 | 0,957687 |
| 48 | 0,137925 | 0,954693 | 0,2328   | 0,969055 | 0,11807  | 0,975762 |
| 49 | 0,243387 | 0,9729   | 0,252527 | 0,970477 | 0,153584 | 0,958764 |
| 50 | 0,142895 | 0,961202 | 0,149067 | 0,95591  | Brak     |          |

Tabela 1 Wartości  $E_p$  oraz  $u$  na poszczególnych sieci neuronowych

| Numer elementu | Waga wejściowe | Biasy warstwy ukrytej | Wagi wyjściowe | Bias wyjściowy |
|----------------|----------------|-----------------------|----------------|----------------|
| 1              | 4.1161         | -1.2180               | -3.4506        | -1.2859        |
| 2              | -2.5967        | -1.7491               | 4.1652         |                |
| 3              | 1.7991         | -0.4235               | 5.6701         |                |
| 4              | -5.5740        | -3.5117               | -2.3746        |                |
| 5              | 4.0042         | -3.3674               | -1.2859        |                |

Tabela 2 Wartości biasów oraz wag wybranej sieci neuronowej

Spis załączników:

- 'best\_network.mat' – plik binarny do programu Matlab z siecią neuronową wyznaczoną w trakcie eksperymentów.
- 'SNN\_Matlab.7z' – spakowane skrypty matlabowe wykorzystywane do wykonania eksperymentu.