Politechnika Śląska w Gliwicach

Wydział Automatyki, Elektroniki i Informatyki

Kierunek Automatyka i Robotyka

**Laboratorium Metod Sztucznej Inteligencji**

# Ćwiczenie 1: Sieci neuronowe jednokierunkowe

Autorzy:

Szymon Golemiec

Jakob Szwajnoch

Grupa Ti 1

Data wykonania ćwiczenia

25/10/2017

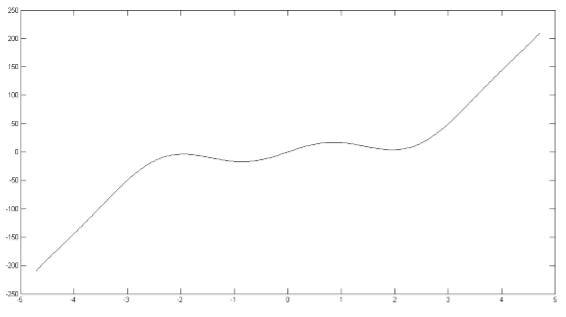
1. Cel ćwiczenia

Celem ćwiczenia była analiza zdolności aproksymacji jednokierunkowych sieci neuronowych. Zbadany został wpływ parametrów sieci takich jak liczba warstw sieci, liczby neuronów w warstwach, postaci funkcji aktywacji oraz wyboru metod uczenia oraz ilości epok na zachowanie sieci. Jakość aproksymacji została oceniona na postawie średnich wartości błędów zbioru uczącego i weryfikującego.

1. Funkcja badana w ćwiczeniu

Do badań przyjęliśmy następującą funkcję:

*y = 16sin(+*

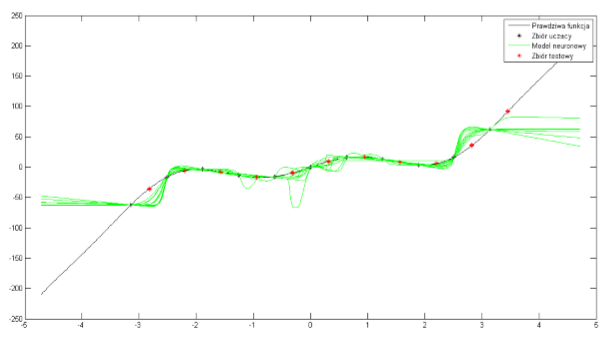


1. Liczba warstw sieci

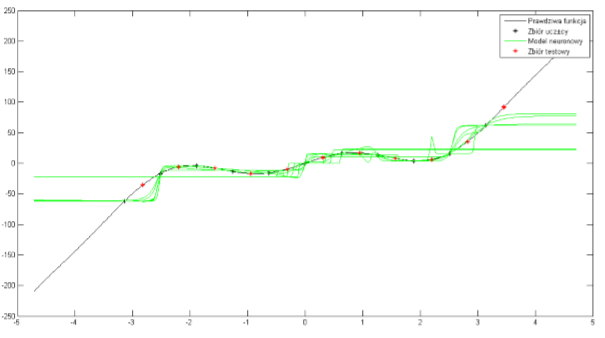
Pierwszą prowadzoną przez nas analizą działania sieci neuronowych był wpływ liczby warstw. Liczba neuronów w poszczególnych warstwach jest stała i została ustalona na 5 (za wyjątkiem ostatniej warstwy posiadającej 1 neuron). Proces uczenia został powtórzony 10 razy i następnie obliczona została średnia wartość błędów dla zbioru uczącego i weryfikującego.

|  |  |  |
| --- | --- | --- |
| **Liczba warstw** | **Średnia wartość błędu dla zbioru uczącego** | **Średnia wartości błędu dla zbioru weryfikującego** |
| **2** | 1.15 | 228 |
| **3** | 116.36 | 303.01 |
| **4** | 106.35 | 343.70 |

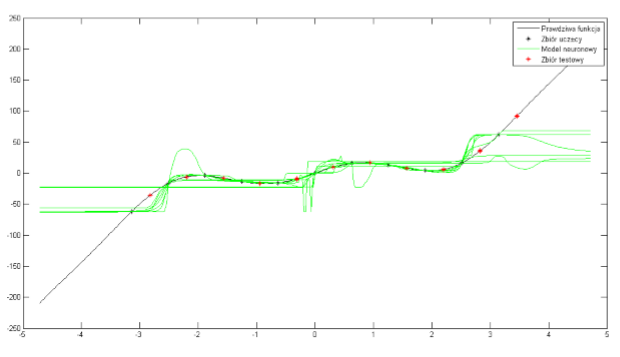
**2 warstwy**



**3 warstwy**



**4 warstwy**



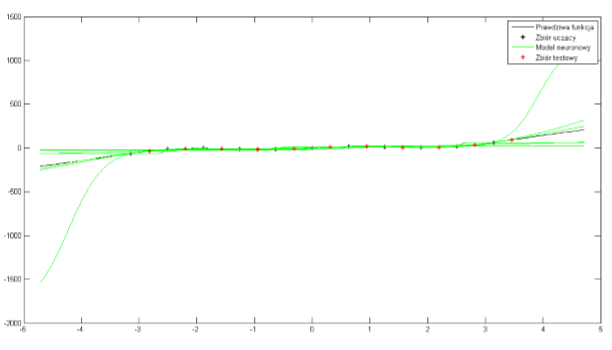
Wraz ze wzrostem liczby warstw nie ma poprawy aproksymacji. Najmniejszą wartość obu błędów (uczącego i weryfikującego) otrzymano dla 2 warstw.

1. **Liczba neuronów w warstwach**

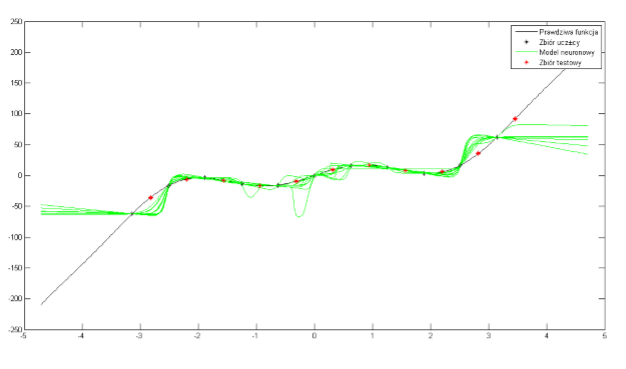
Przeprowadziliśmy analizę zachowania sieci w zależności od liczby neuronów w warstwach. W testach przyjęto model dwuwarstwowych i zmieniano liczbę neuronów w warstwie pierwszej, druga warstwa zawierała zawsze jeden neuron.

|  |  |  |
| --- | --- | --- |
| Liczba neuronów w warstwie pierwszej | Średnia wartość błędu dla zbioru uczącego | Średnia wartość błędu dla zbioru weryfikującego |
| 3 | 84.48 | 270.18 |
| 5 | 1.15 | 228 |
| 10 | 0.69 | 197.47 |
| 50 | 2.12 \* | 88.59 |
| 100 | 9.46 \* | 83.66 |
| 500 | 3.97 \* | 108.71 |

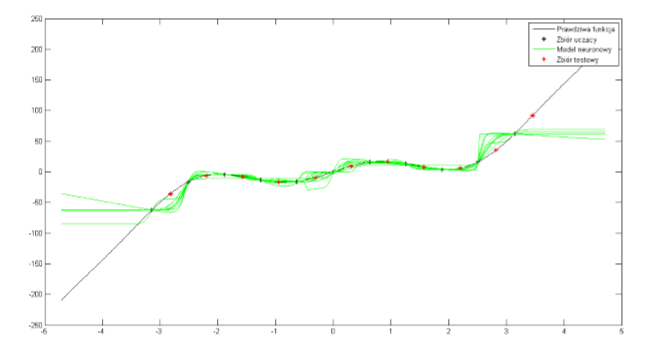
**3 neurony**

****

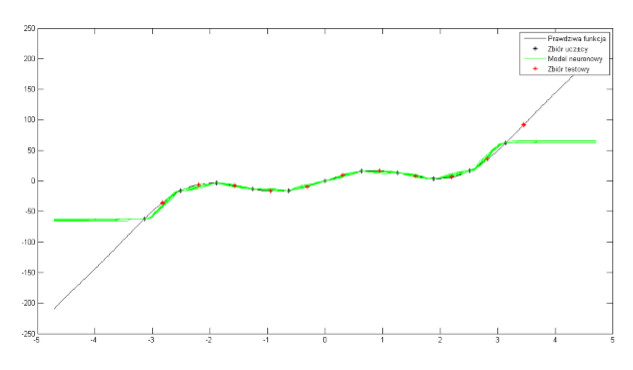
**5 neuronów**

****

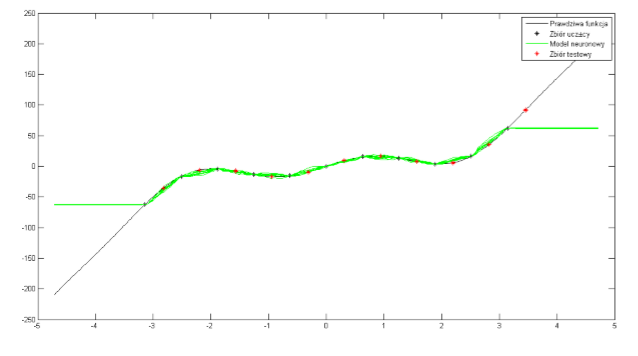
**10 neuronów**

****

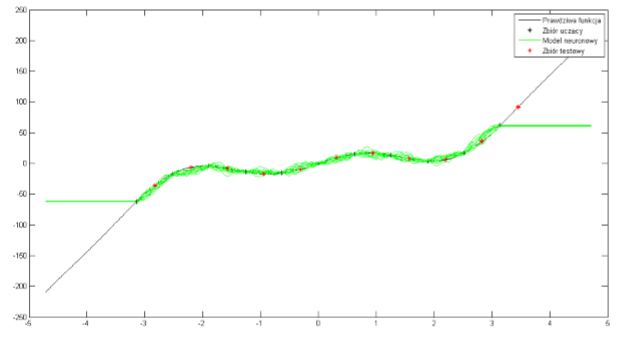
**50 neuronów**

****

**100 neuronów**

****

**500 neuronów**

****

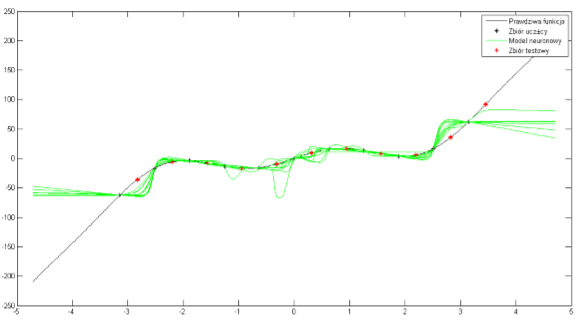
Najlepszy wynik udało się osiągnąć dla liczby 50 neutronów. Zbyt mała ilość neuronów skutkuje duże niedokładności odwzorowania zbiorów testowego i uczącego. Natomiast zbyt duża ilość neuronów skutkuje przetrenowaniem sieci i mimo dobrego dopasowania dla zbioru uczącego, otrzymujemy gorsze dopasowanie dla zbioru weryfikującego.

1. **Postać funkcji aktywacji**

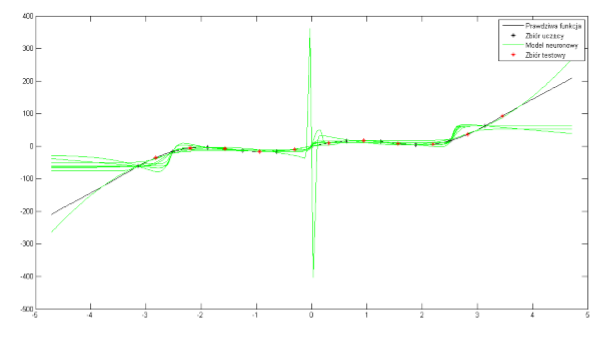
Kolejnym krokiem było przeprowadzenie analizy działania sieci w zależności od jej aktywatora. Testy zostały przeprowadzone dla sieci posiadającej dwie warstwy, w której funkcja aktywacyjna w warstwie numer dwa była zawsze stałego typu (purelin), natomiast pierwsza była zmienna.

|  |  |  |
| --- | --- | --- |
| Funkcja aktywacyjna | Średnia wartość błędu dla zbioru uczącego | Średnia wartość błędu dla zbioru weryfikującego |
| Tansig | 1.15 | 228 |
| Logsig | 14.25 | 207.98 |
| Purelin | 217.29 | 336.22 |

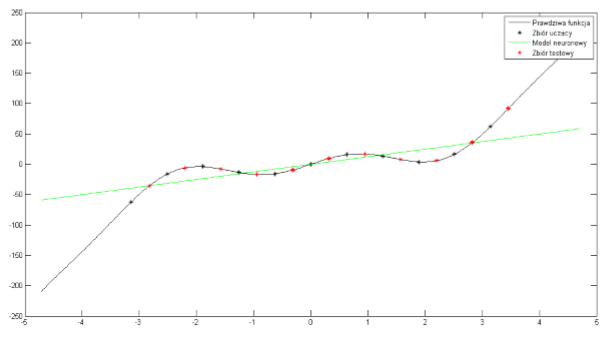
**Tansig**

****

**Logsig**

****

**Purelin**

****

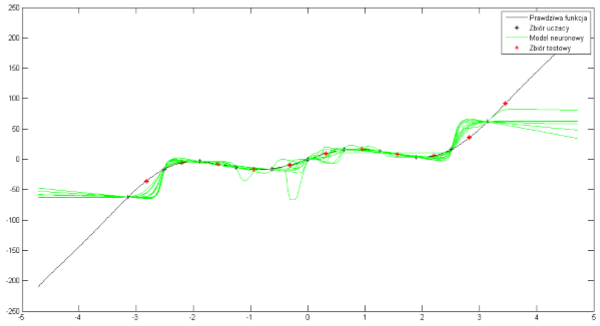
Funkcja liniowa zdecydowanie nie nadaje się do aproskymacji wybranej funkcji co bardzo dobrze widać na ostatnim załączonym wykresie. Natomiast najlepsze efekty uzyskaliśmy dla funkcji aktywacyjnej będącej w postaci tangensa hiperbolicznego. W miarę dobry rezultat poza anomaliami w okolicy 0 otrzymaliśmy dla funkcji logarytmicznej.

1. **Wybór metody uczenia**

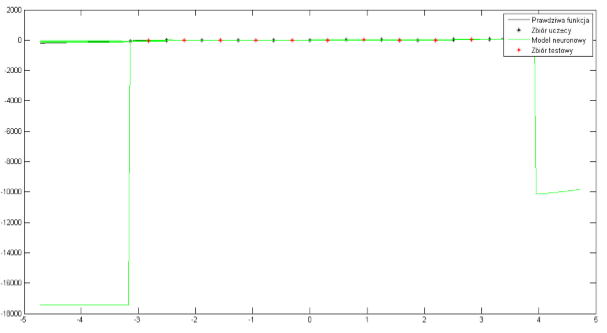
Sprawdzimy metody nauczania na działanie sieci.

|  |  |  |
| --- | --- | --- |
| Metody nauczania | Średnia wartość błędu dla zbioru uczącego | Średnia wartość błędu dla zbioru weryfikującego |
| Trainlm | 1.15 | 228 |
| Trainbfg | 15.80 | 205.25 |
| Trainrp | 14.11 | 31.08 |

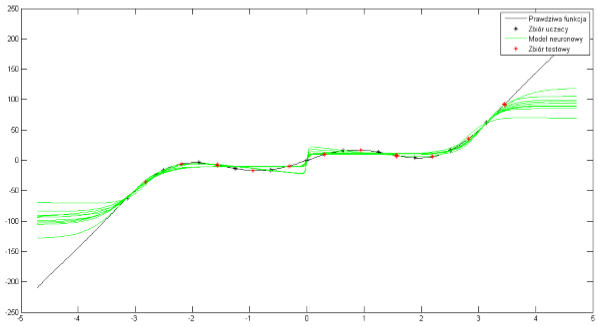
**Trainlm**

****

**Trainbfg**

****

**Trainrp**



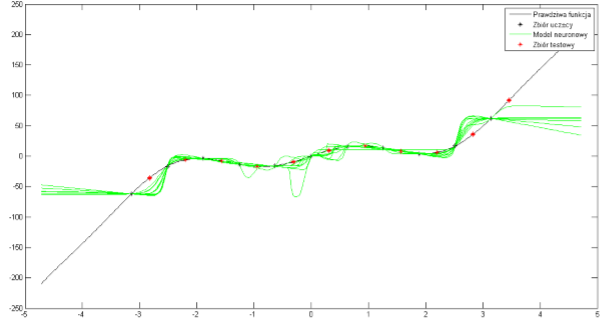
Dobre wyniki uzyskano dla metod trainlm i trainrp. W przypadku trainlm ogólne dopasowanie jest dobre, jednak w niektórych miejscach występują znaczne rozbieżności od badanej funkcji. Dla trainrp sytuacja jest podobna - błędy pojawiają się tym razem w okolicy 0. Metodą trainbfg uzyskany rezultat jest najgorszy, a wyznaczona funkcja bardzo odbiega od pierwowzoru w punktach skrajnych.

1. **Liczba epok**

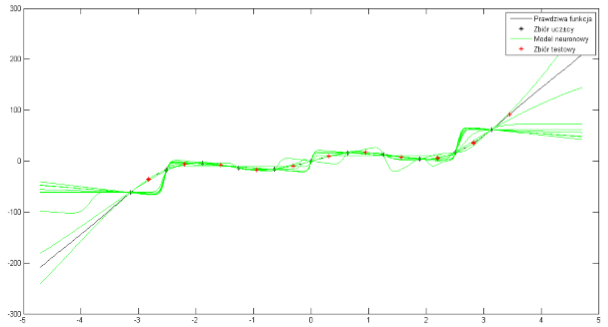
Wpływ liczby epok na wyniki aproksymacji

|  |  |  |
| --- | --- | --- |
| Liczba epok | Średnia wartość błędu dla zbioru uczącego | Średnia wartość błędu dla zbioru weryfikującego |
| 200 | 1.15 | 228 |
| 300 | 2.24 | 199.59 |
| 400 | 14.63 | 235.79 |

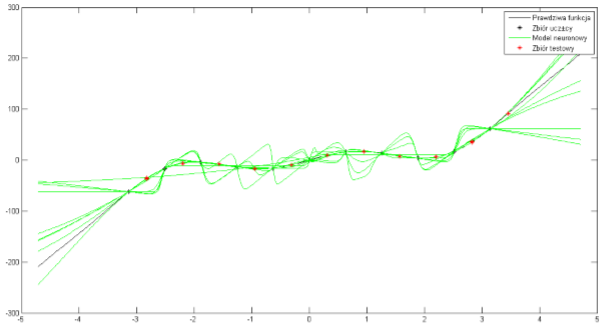
**200 epok**

****

**300 epok**

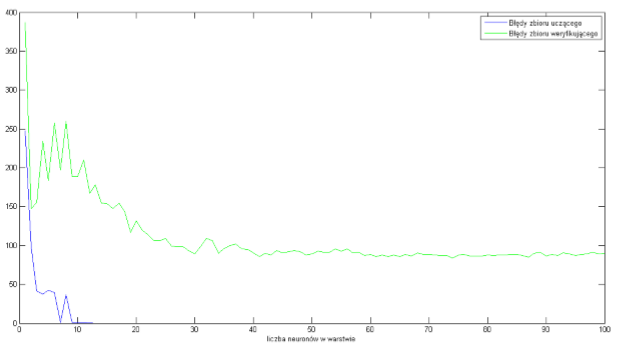
****

**400 epok**



Najlepsze rezultaty otrzymano dla 200 epok zbioru uczącego natomiast jeśli chodzi o zbiór weryfikujący najlepsze okazał się 300 epok. Wraz z wzrostem ilości tętnień można wyraźnie zauważyć wzrost tętnień na wykresach.

1. **Zależność jakości aproksymacji od liczby neuronów**



Błędy zbioru uczącego maleją wraz ze wzrostem liczby neuronów, aż osiągają wartości bliskie zeru. Błędy zbioru weryfikującego maleją do pewnego momentu, aby następnie osiągnąć ustalony poziom lub nawet się zwiększać.

1. **Wnioski i podsumowanie**

* Dla każdego programu należy prowadzić odrębną analizę, by dostosować .odpowiednie parametry sieci
* Wzrost ilości neutronów nie jest równoznaczny z poprawą zachowania sieci – jeśli liczba neuronów przekroczy wartość krytyczną, następuje przeuczenie sieci i błędy zaczynają się zwiększać. Liczba ta jest zależna od postawionego sieci zadania. Tak samo liczba warstw jak i epok nie koniecznie prowadzi do poprawy wyników.
* Należy także zwrócić uwagę na wybranie odpowiedniej funkcji aktywacji do rozwiązania problemu. Jej zły dobór uniemożliwi uzyskanie zadowalających nas wyników – w przypadku naszej funkcji takim wyborem byłą funkcja liniowa.