Sprawozdanie

Temat:

Budowa i działanie sieci wielowarstwowej typu feedforward

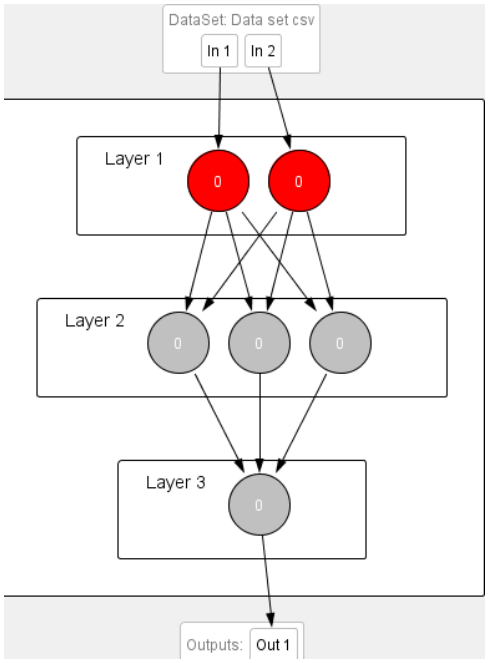
Cel ćwiczenia:

Celem ćwiczenia jest poznanie budowy i działania wielowarstwowych sieci neuronowych poprzez uczenie kształtu wykresu funkcji matematycznej z użyciem algorytmu wstecznej propagacji błędu.

1) **Syntetyczny opis budowy oraz wykorzystanego algorytmu uczenia**
   Scenariusz jest realizowany przez sieci wielowarstwowe zbudowane z perceptronów. Algorytm uczenia perceptronu przedstawiono poniżej:

$$w_1 \mathrel{+}= n * (d-y) * x_1$$
$$w_2 \mathrel{+}= n * (d-y) * x_2$$
$$b \mathrel{+}= n * (d-y)$$

   W jest wagą wejścia, n - współczynnikiem uczenia, d – wartością otrzymaną na wyjściu, y – wartością oczekiwaną, x – wartością na wejściu.
   Funkcję aktywację stanowi funkcja sigmoidalna bipolarna.
   Sieć zbudowana jest w następujący sposób:



   Layer 1 stanowi warstę wejściową. Jej neuronu aktywuje funkcja liniowa, nie wykonują więc żadnych operacji.

   Layer 2 to warstwa perceptronów zbuowanych zgodnie z powyższym schematem.

   Layer 3 składa się z jednego perceptronu, który agreguje dane z poprzedniej warstwy, tak aby wynikiem sieci była jedna liczba.

   Jest to sieć feedforward czyli jednokierunkowa. Oznacza to że sygnały przekazywane są w najbardziej intyicyjny sposób czyli tylko do kolejnej warstwy.
   Sieć wykorzystuje algorytm propagacji wstecznej do wyznaczenia błędu w poszczególnych neuronach. W tym algorytmie wartość błędu stanowi sumę iloczynu błędu w neuronach następujących i wag odpowiednich wejść tych neuronów.

2) **Zestawienie otrzymanych wyników**
   Dane uczące generowane są za pomocą autorskiego programu i zapisywane do pliku tekstowego w postaci:

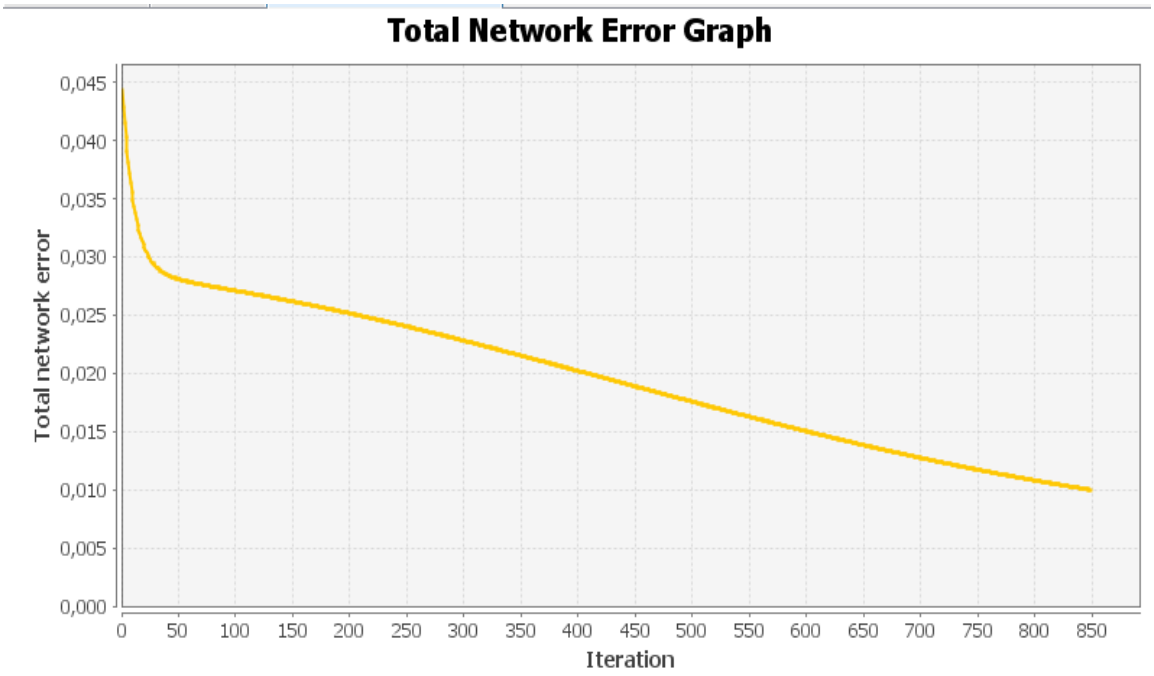
```
-2;-2;8
-2;-1.6;24.6501699437495
-2;-1.2;12.3498300562505
-2;-0.8;11.5498300562505
-2;-0.4;22.2501699437495
-2;0;4
-2;0.4;22.2501699437495
-2;0.8;11.5498300562505
-2;1.2;12.3498300562505
-2;1.6;24.6501699437495
-2;2;8
-1.6;-2;24.6501699437495
-1.6;-1.6;41.300339887499
-1.6;-1.2;29
-1.6;-0.8;28.2
-1.6;-0.4;38.9003398874989
-1.6;0;20.6501699437495
-1.6;0.4;38.900339887499
-1.6;0.8;28.2
-1.6;1.2;29
-1.6;1.6;41.300339887499
-1.6;2;24.6501699437495
-1.2;-2;12.3498300562505
```

Po znormalizowaniu ma postać:

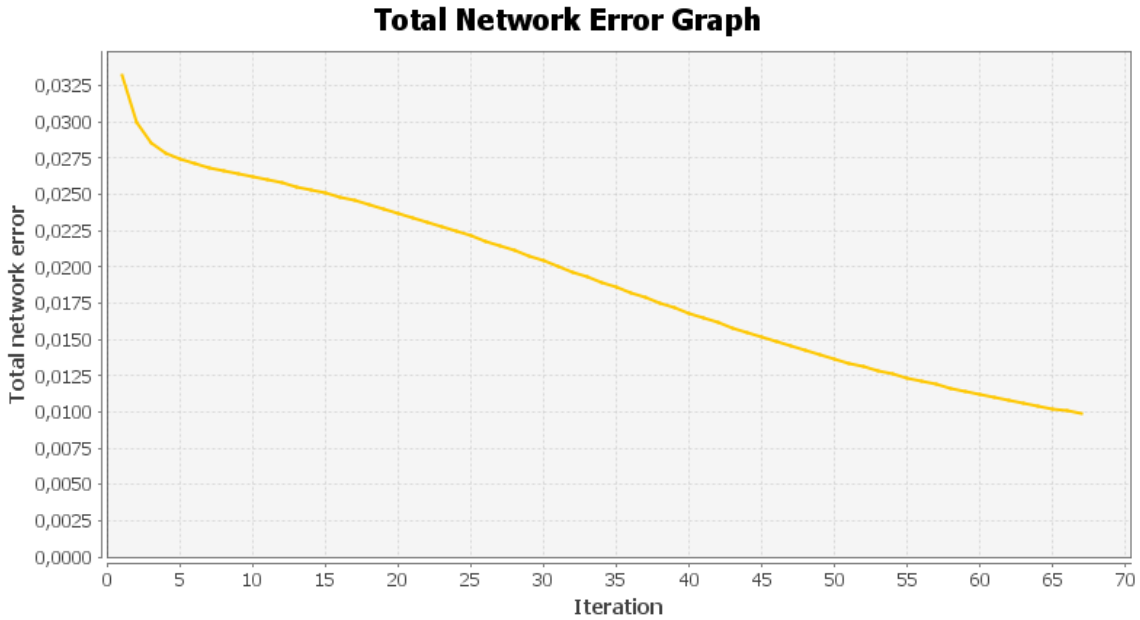
Input1	Input2	Output1
0.0	0.0	0.1937030063624605
0.0	0.09999999999999998	0.5968515031812303
0.0	0.2	0.2990249012451496
0.0	0.3	0.2796546006089035
0.0	0.4	0.5387406012724921
0.0	0.5	0.09685150318123024
0.0	0.6	0.5387406012724921
0.0	0.7	0.2796546006089035
0.0	0.8	0.2990249012451496
0.0	0.9	0.5968515031812303
0.0	1.0	0.1937030063624605
0.09999999999999998	0.0	0.5968515031812303
0.09999999999999998	0.09999999999999998	1.0
0.09999999999999998	0.2	0.7021733980639193
0.09999999999999998	0.3	0.6828030974276732
0.09999999999999998	0.4	0.9418890980912594
0.09999999999999998	0.5	0.5
0.09999999999999998	0.6	0.9418890980912619
0.09999999999999998	0.7	0.6828030974276732
0.09999999999999998	0.8	0.7021733980639193
0.09999999999999998	0.9	1.0
0.09999999999999998	1.0	0.5968515031812303
0.2	0.0	0.2990249012451496
0.2	0.09999999999999998	0.7021733980639193
0.2	0.2	0.40434679612783864
0.2	0.3	0.384976495491595
0.2	0.4	0.6440624961551812
0.2	0.5	0.2021733980639198
0.2	0.6	0.6440624961551812
0.2	0.7	0.3849764954915926
0.2	0.8	0.404346796127841
0.2	0.9	0.7021733980639193
0.2	1.0	0.2990249012451496

Poniżej przedstawiono wyniki uczenia dla sieci w konfiguracji: 2 neurony wejściowe, **3 neurony**, 1 neuron wyjściowy

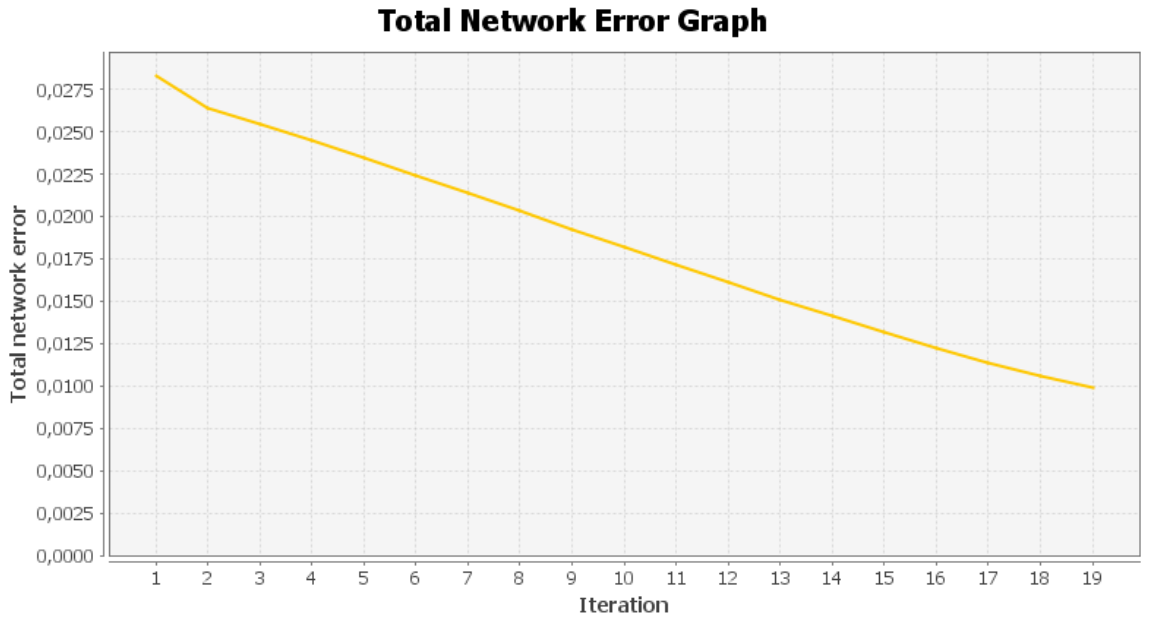
Współczynnik uczenia = 0,01
Ilość epok = 850
Błąd średniokwadratowy = 0,01



Współczynnik uczenia = 0,1
Ilość epok = 67
Błąd średniokwadratowy = 0,01

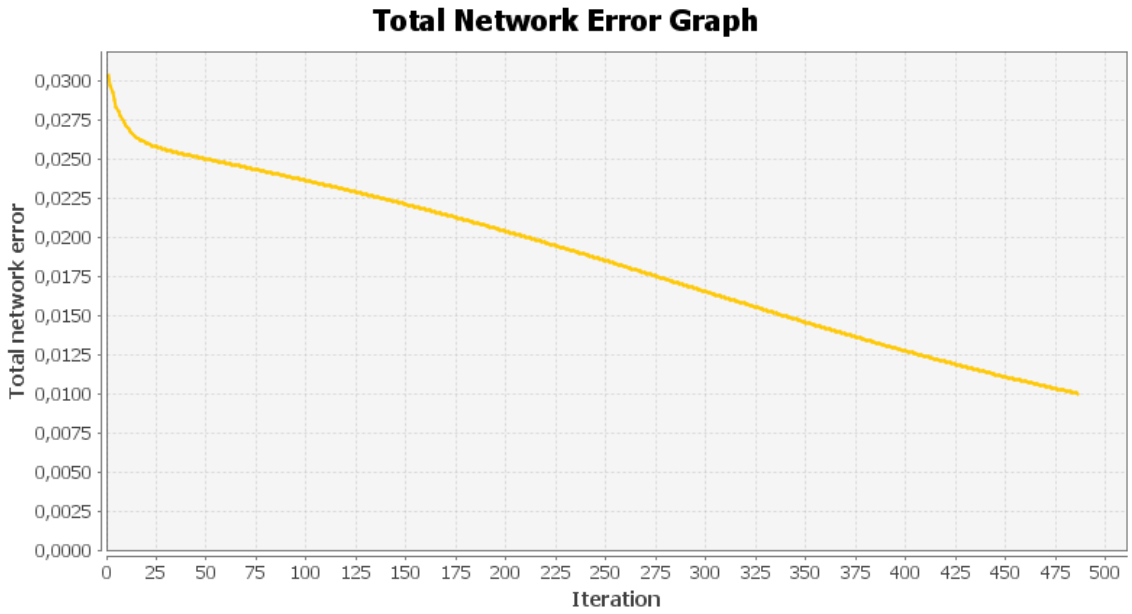


Współczynnik uczenia = 0,5
Ilość epok = 19
Błąd średniokwadratowy = 0,01

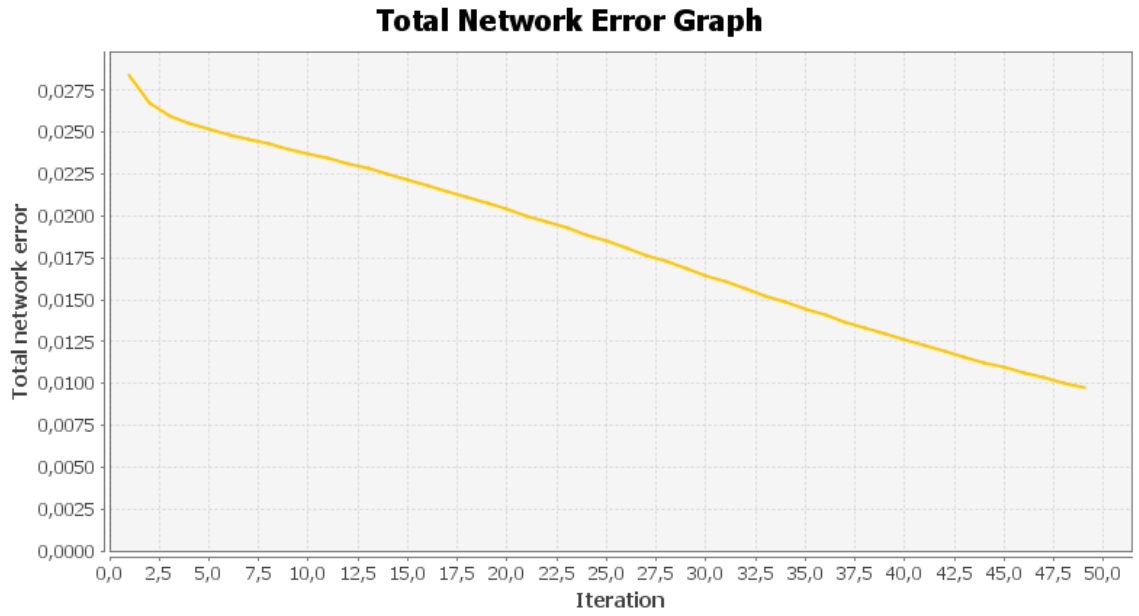


Poniżej przedstawiono wyniki uczenia dla sieci w konfiguracji: 2 neurony wejściowe, **3 neurony, 3 neurony**, 1 neuron wyjściowy

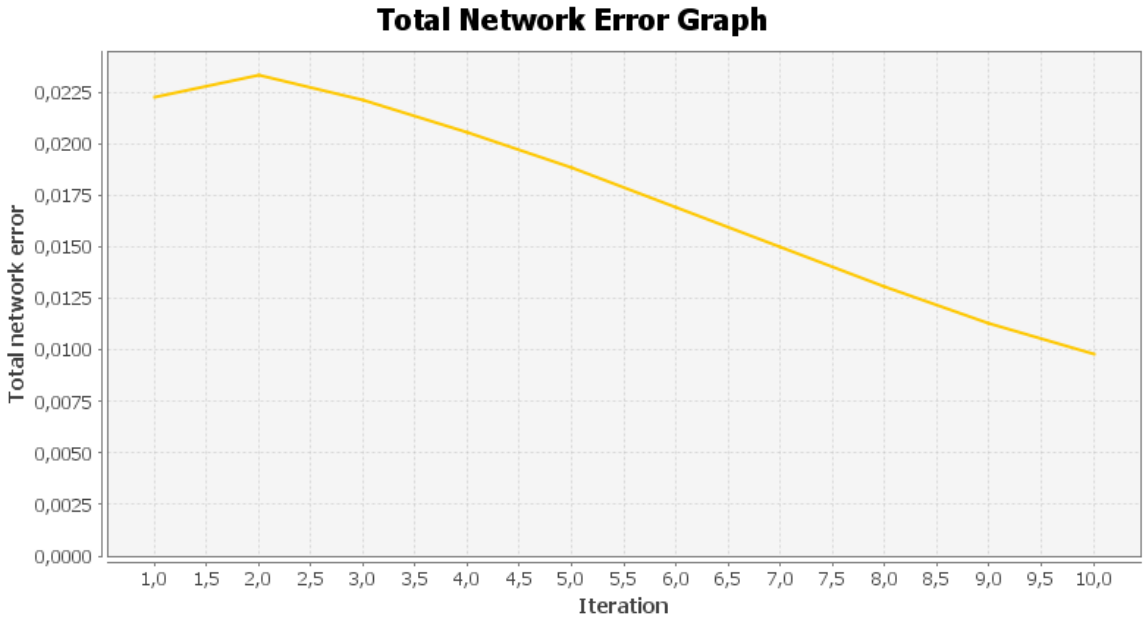
Współczynnik uczenia = 0,01
Ilość epok = 489
Błąd średniokwadratowy = 0,01



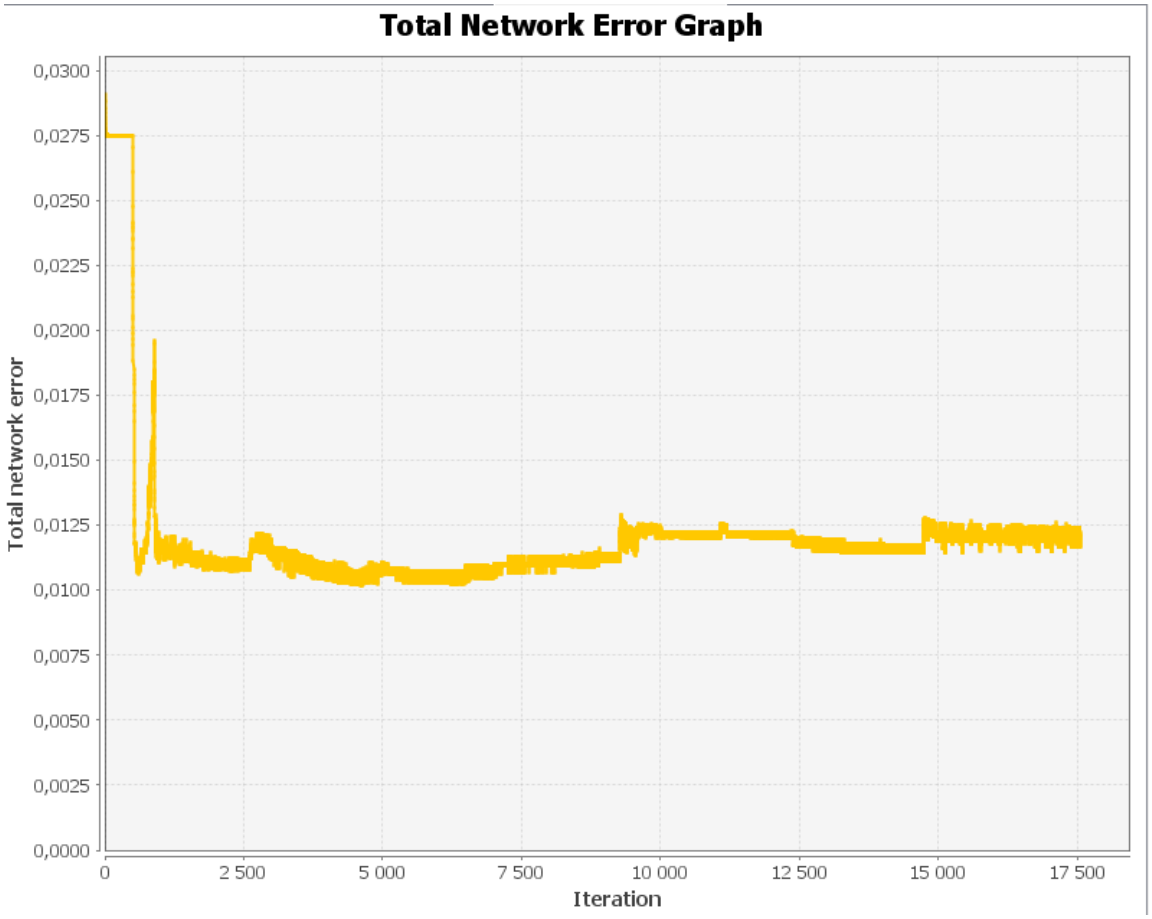
Współczynnik uczenia = 0,1
Ilość epok = 49
Błąd średniokwadratowy = 0,01



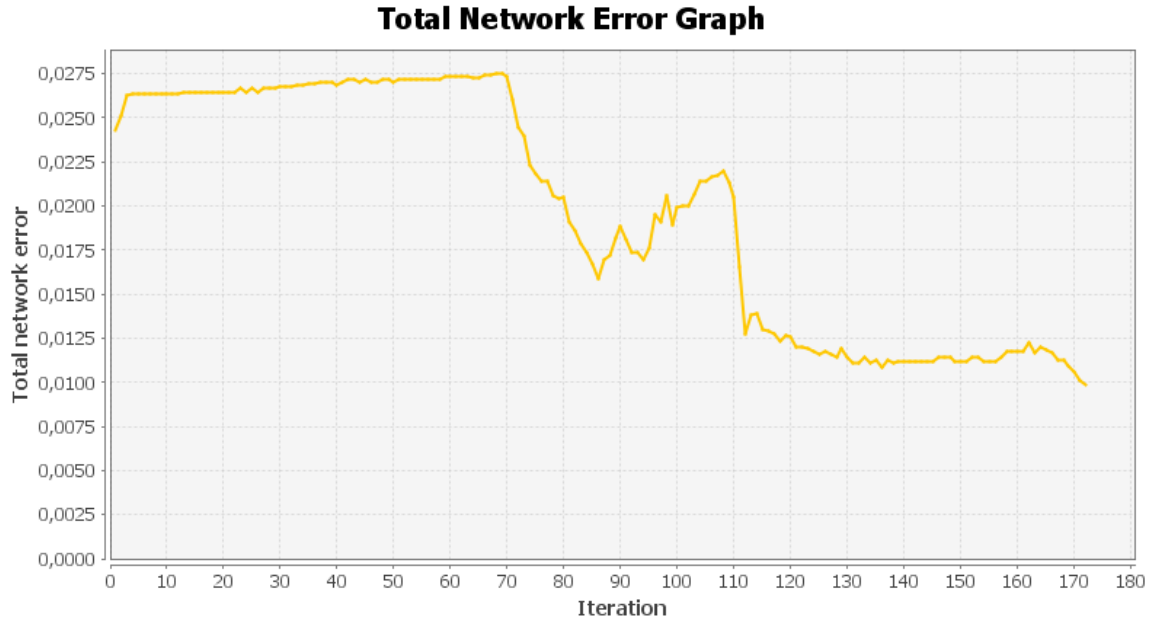
Współczynnik uczenia = 0,5
Ilość epok = 10
Błąd średniokwadratowy = 0,01



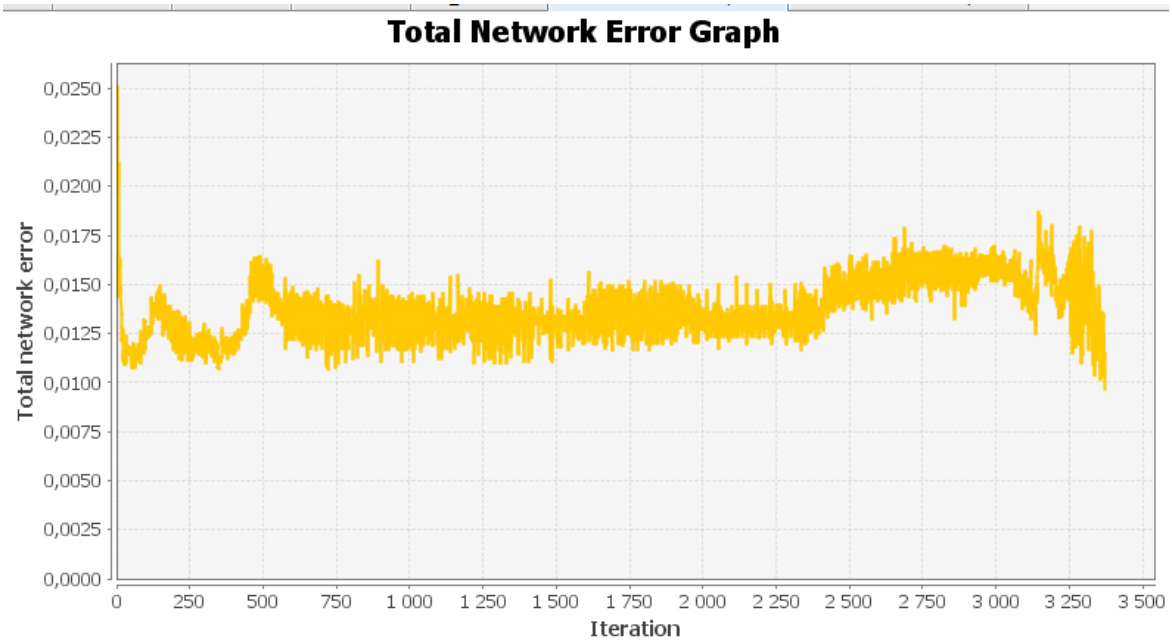
Współczynnik uczenia = 0,01
Ilość epok = minimum ok. 4900
Błąd średniokwadratowy = 0,0114
Jak widać w tym procesie uczenia sięc nie osiągnęła zadanej skuteczności. Można natomiast zaobserwować przeuczenie.



Współczynnik uczenia = 0,1
Ilość epok = minimum ok. 172
Błąd średniokwadratowy = 0,01



Współczynnik uczenia = 0,5
Ilość epok = minimum ok. 3300
Błąd średniokwadratowy = 0,01



3) Analiza i dyskusja błędów uczenia i testowania opracowanej sieci w zależności od wartości współczynnika uczenia oraz ilości warstw i neuronów.

   Na pierwszych 6 wykresach widać bardzo wyraźny wpływ współczynnika uczenia na ilośc epok potrzebnych do nauczenia sieci. Mianowicie im wyższy współczynnik uczenia tym szybciej sieć się uczy.

   Sieć uczy się także szybciej w przypadku gdy zbudowana jest z 2 a nie jednej warstwy. Trudno jednak wskazać na poprawność takiej korelacji zważywszy na wyniki testów sieci w konfiguracji 3,3,3. W przypadku większej ilości warstw sytuacja mocno się komplikuje. Sieć może nigdy nie osiągnąć oczekiwanej sprawności lub zajmuje jej to dużo więcej czasu. Widać także wahania błędu wkolejnych epokach. Sieć „skacze” wokół poprawnego rozwiązania.

4) Wnioski

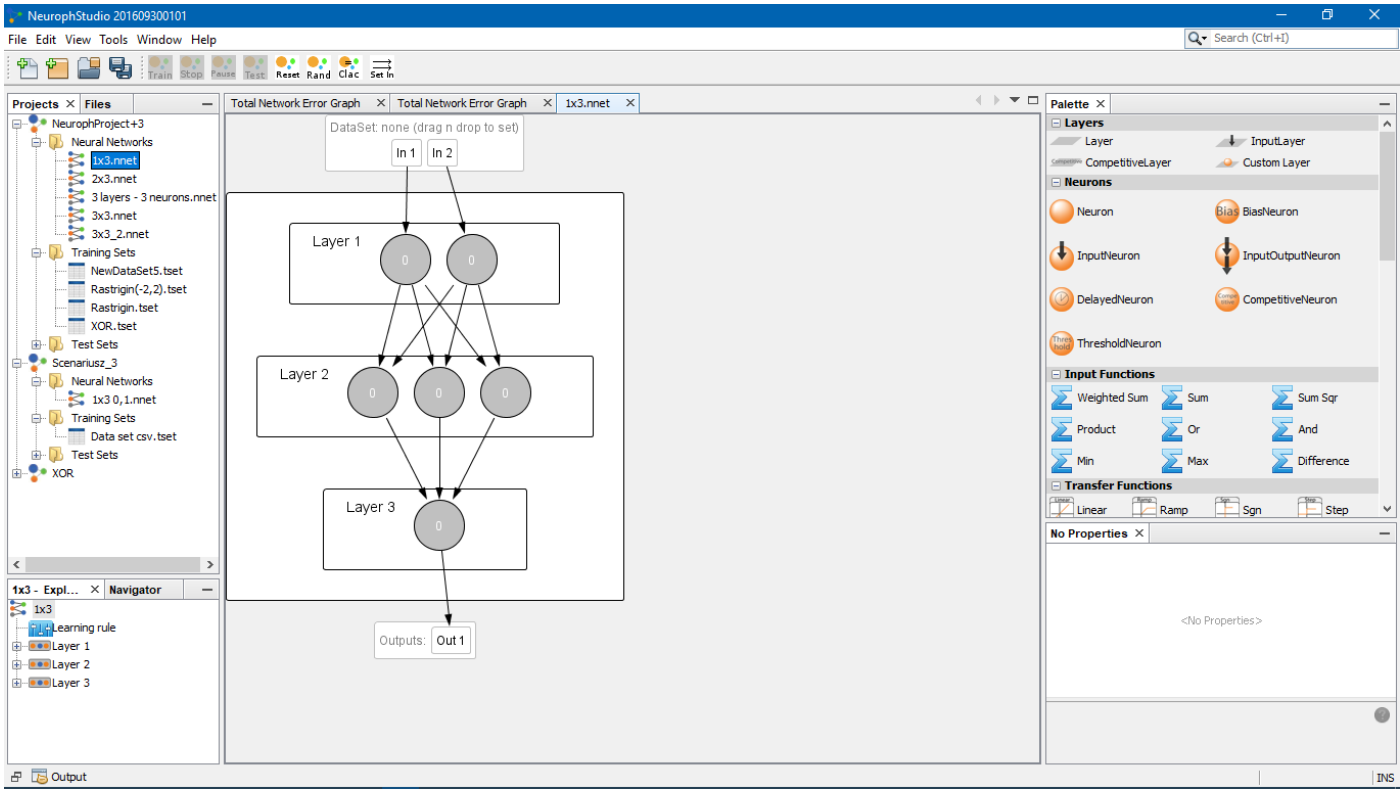
   Można sformułować wniosek mówiący iż w większości przypadków wraz ze wzrostem współczynnika uczenia maleje ilośc epok potrzebnych do nauczenia sieci. Nie jest to zasada kompletna zważywszy na wyniki uczenia sieci 3,3,3 o współczynniku 0,1 i 0,5, jednak nawet w przypadku sieci o 3 warstwach występuje (słaba) korelacja. Należy jednak pamiętać iż zwiększając współczynnik uczenia zawsze ryzykujemy „przeskoczenie” optymalnego rozwiązania.

   Należy także zauważyć iż wzrost ilości warstw nie musi przekładać się na skuteczność epok. Można stworzyć sieć zbyt skompikowaną dla danego zagadnienia.

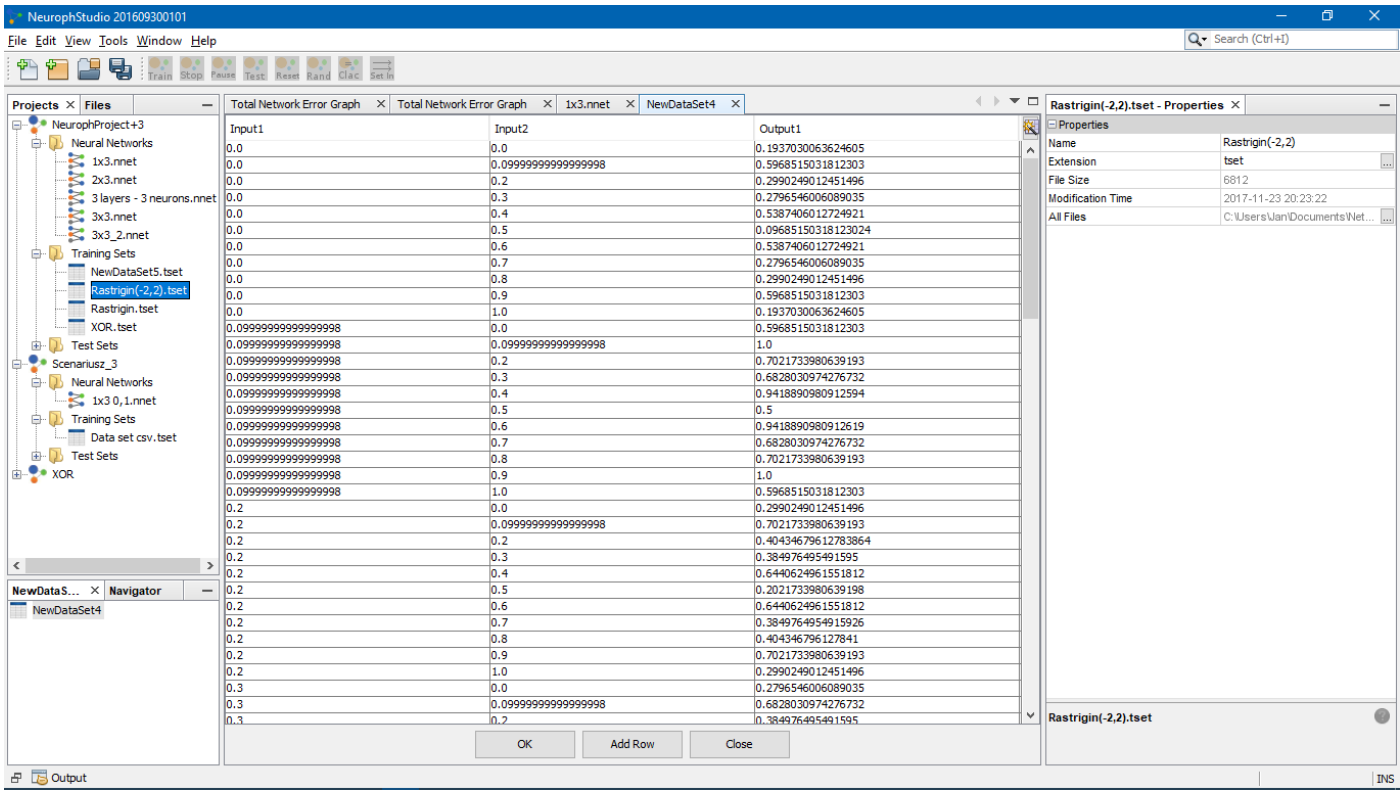
   Można także wspomnieć, że sieci neuronowe dobrze sprawdzają się w przypadku przewidywania wartości funkcji w zadanym zakresie. Warunkiem poprawności tego stwierdzenia jest posiadania dostatecznej ilości danych uczących w podanym przedziale.

5) Listing

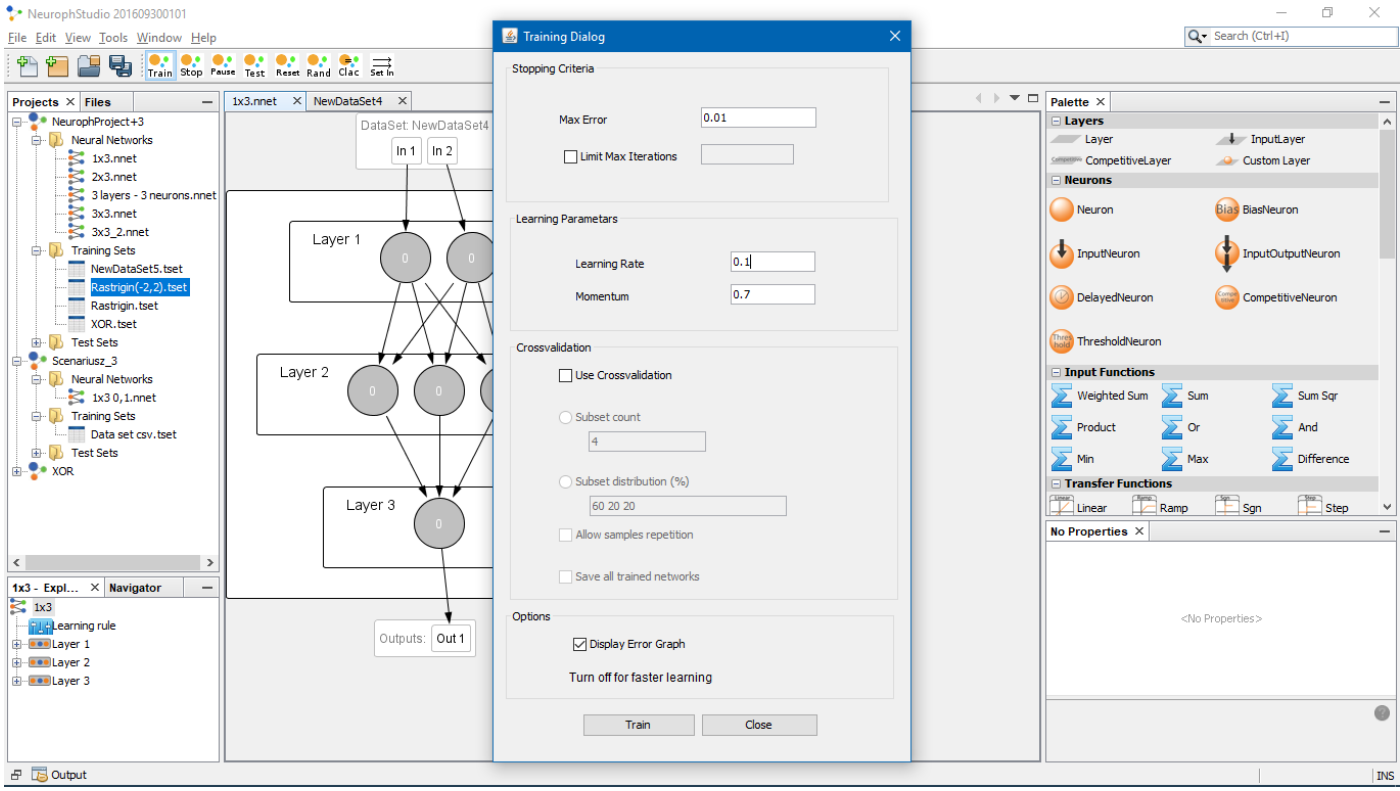
   Sieci przygotowano w programie Neuroph Studio
   Główne okno i budowa sieci:



Zetaw danych uczących:



Okno konfiguracji uczenia:



Program generujący dane uczące:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Scenariusz_3_Data_Generator
{
    class Program
    {
        static public double Rastrigin3D(double x1, double x2)
        {

            double suma = 20;
            suma += x1 * x1 - (10 * Math.Cos(2 * Math.PI * x1));
            suma += x2 * x2 - (10 * Math.Cos(2 * Math.PI * x2));
            return suma;
        }

        static void Main(string[] args)
        {
            int N = 11;
            int M = 11;
            string[] lines = new string[N*M];
            string line;
            int start = -2;
            int stop = 2;
            double step = 0.4;
            double x =start;
            double y=start;
            for (int i = 0; i < N; i++)
            {
                x = start+ i * step;
                for (int j = 0; j < M; j++)
                {
                    y = start +j * step;
                    line = Convert.ToString(x) + ";"+Convert.ToString(y) + ";"+
Convert.ToString(Rastrigin3D(x, y));
                    lines[i*N+j] = line;
```

```
                line = "";
            }
        }
        using (System.IO.StreamWriter file = new System.IO.StreamWriter("DataSet.txt"))
        {
            foreach (string l in lines)
            {
                // If the line doesn't contain the word 'Second', write the line to the file.
                //if (!l.Contains("Second"))
                {
                    file.WriteLine(l);
                }
            }
        }


    }
}
```