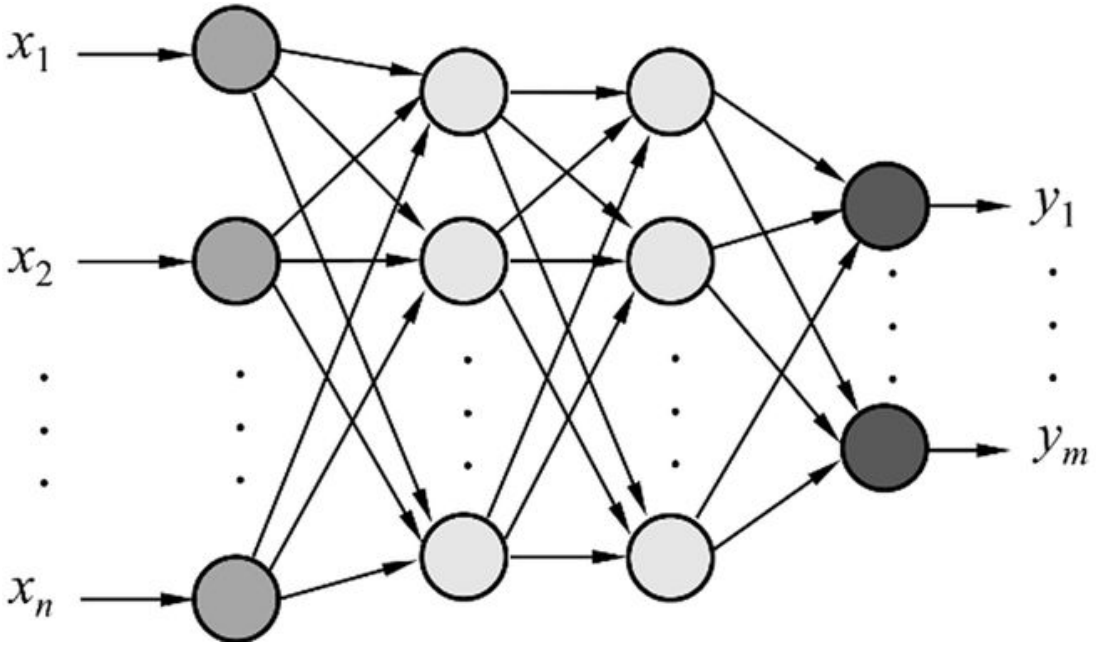
Maciej Stawarz Nr albumu 284310	Inżynieria Obliczeniowa Podstawy Sztucznej Inteligencji
Projekt 3 - Budowa i działanie sieci wielowarstwowej typu feedforward - Sprawozdanie	

1) Cel ćwiczenia:
   Celem poniższego ćwiczenia było poznanie budowy i działania wielowarstwowych sieci neuronowych poprzez uczenie kształtu funkcji matematycznej z użyciem algorytmu wstecznej propagacji błędu.

2) Opis użytej struktury i algorytmu uczenia, oraz funkcji przykładowej:
   Sztuczna sieć neuronowa - struktura matematyczno-programowa, realizująca różne funkcje, najczęściej wzorowane na działaniu biologicznego mózgu.

   W tym przypadku użyto wielowarstwowych sieci typu feedforward - sieci w których połączenia między węzłami nie tworzą pętli ani obiegów. Informacja (sygnał) przepływa tylko w jednym kierunku, z węzłów wejściowych, przez ukryte, do wyjściowych.



   Propagacja wsteczna to sposób uczenia nadzorowanego wyżej wspomnianych sieci, oparty na minimalizacji funkcji błędu (np. średniokwadratowego) wykorzystując metodę największego spadku. W praktyce polega to na zmianie wag sygnałów wejściowych w każdym neuronie, tak by wartość błędu dla kolejnych par uczących była jak najmniejsza.

Kroki:

1. Wyznaczenie odpowiedzi sieci na zadany sygnał wejściowy.
2. Wyznaczenie błędu na neuronach sieci wyjściowej i przesłanie go wstecz, do warstwy wejściowej.
3. Zmiana wag neuronów. Dla aktualizacji po akcji każdego elementu jest ona dana wzorem:

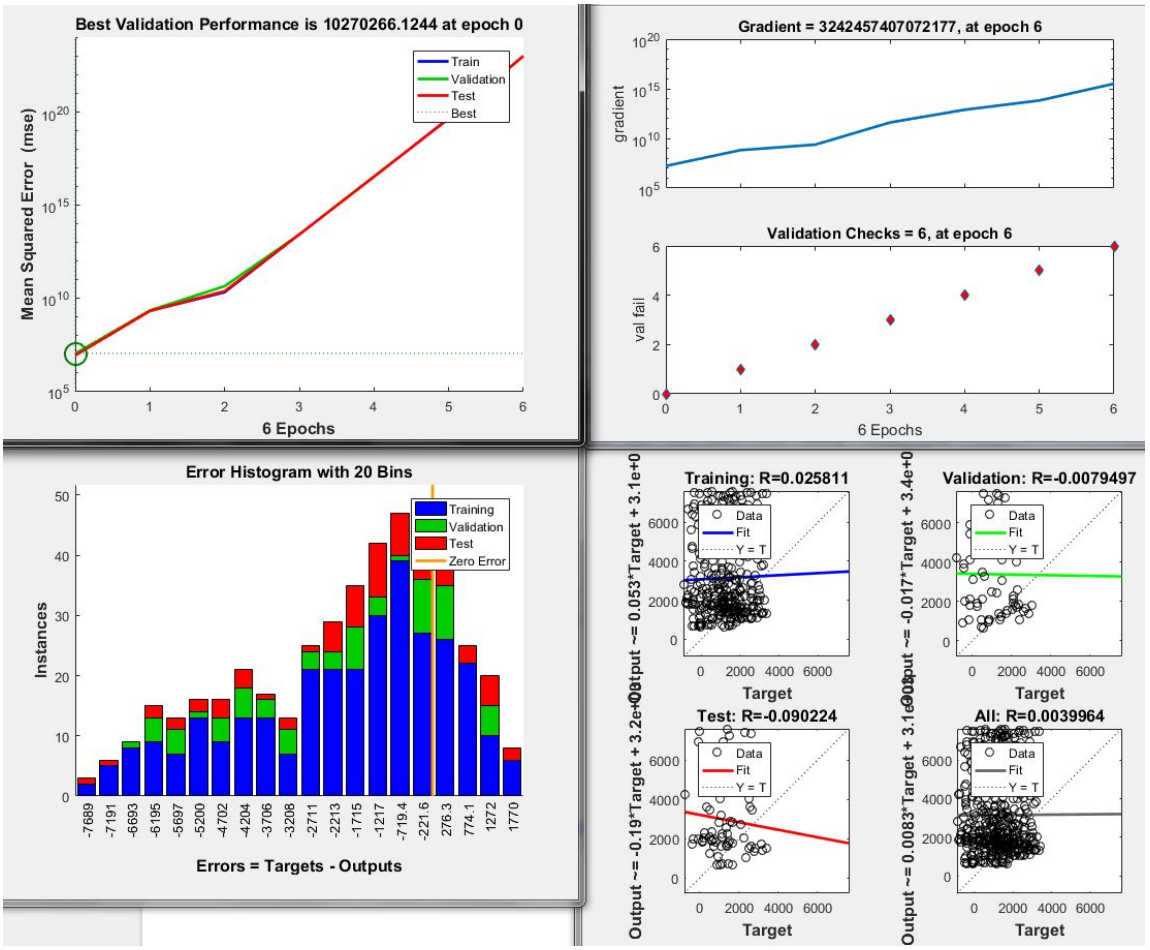
$$E = \frac{1}{2}\sum_{k=1}^{m}\left(z_k(t) - y_k(t)\right)^2$$

W tym przypadku sieć jest sprawdzana na przykładnie funkcji Rastrigin - funkcji używanej do testowania efektywności algorytmów i sieci optymalizacyjnych. Znalezienie jej minimum jest trudne ze względu na dużą ilość ekstremów lokalnych. Dana jest ona wzorem:

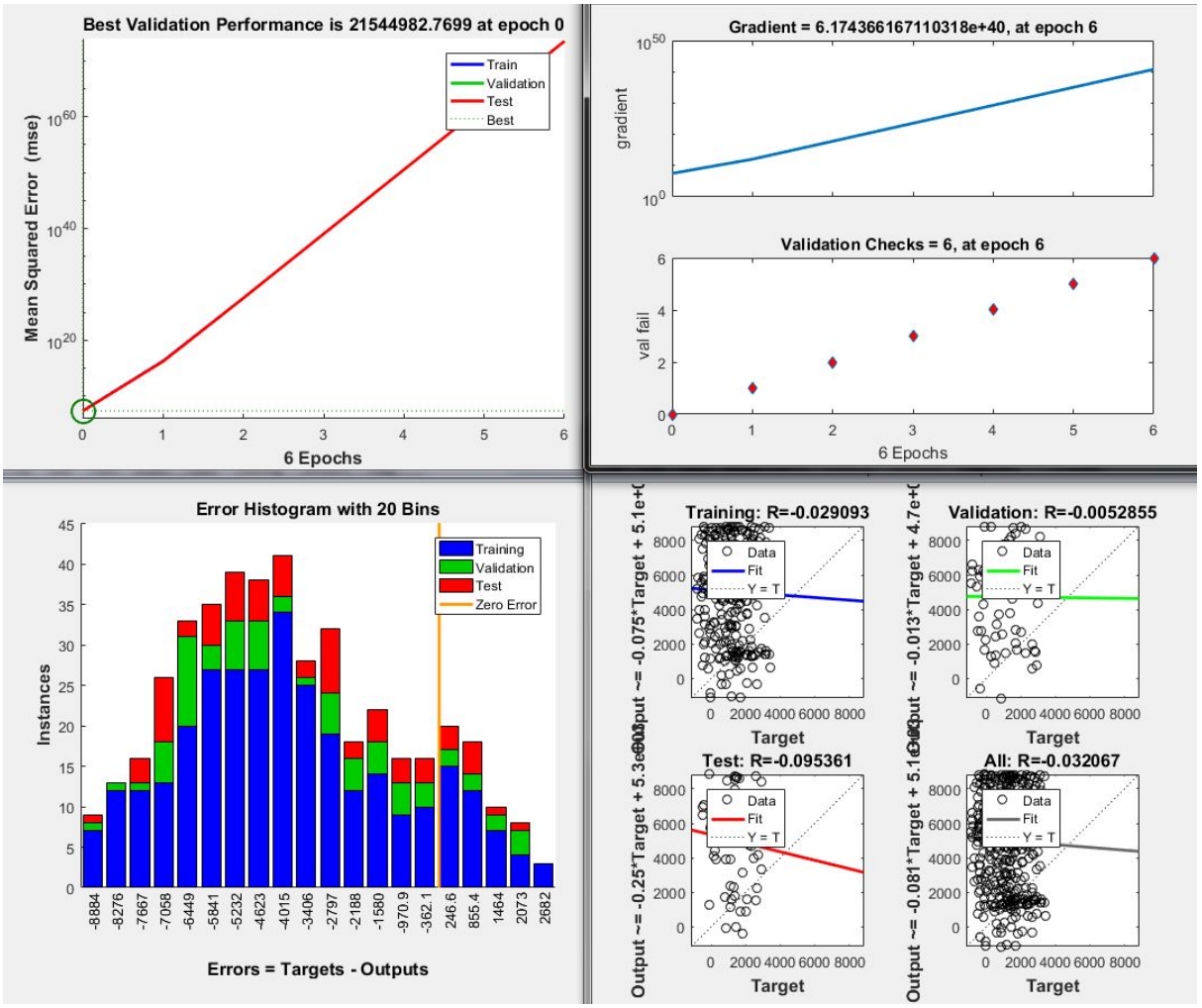
$$f(\mathbf{x}) = An + \sum_{i=1}^{n}\left[x_i^2 - A\cos(2\pi x_i)\right]$$

3) <u>Zestawienie otrzymanych wyników:</u>

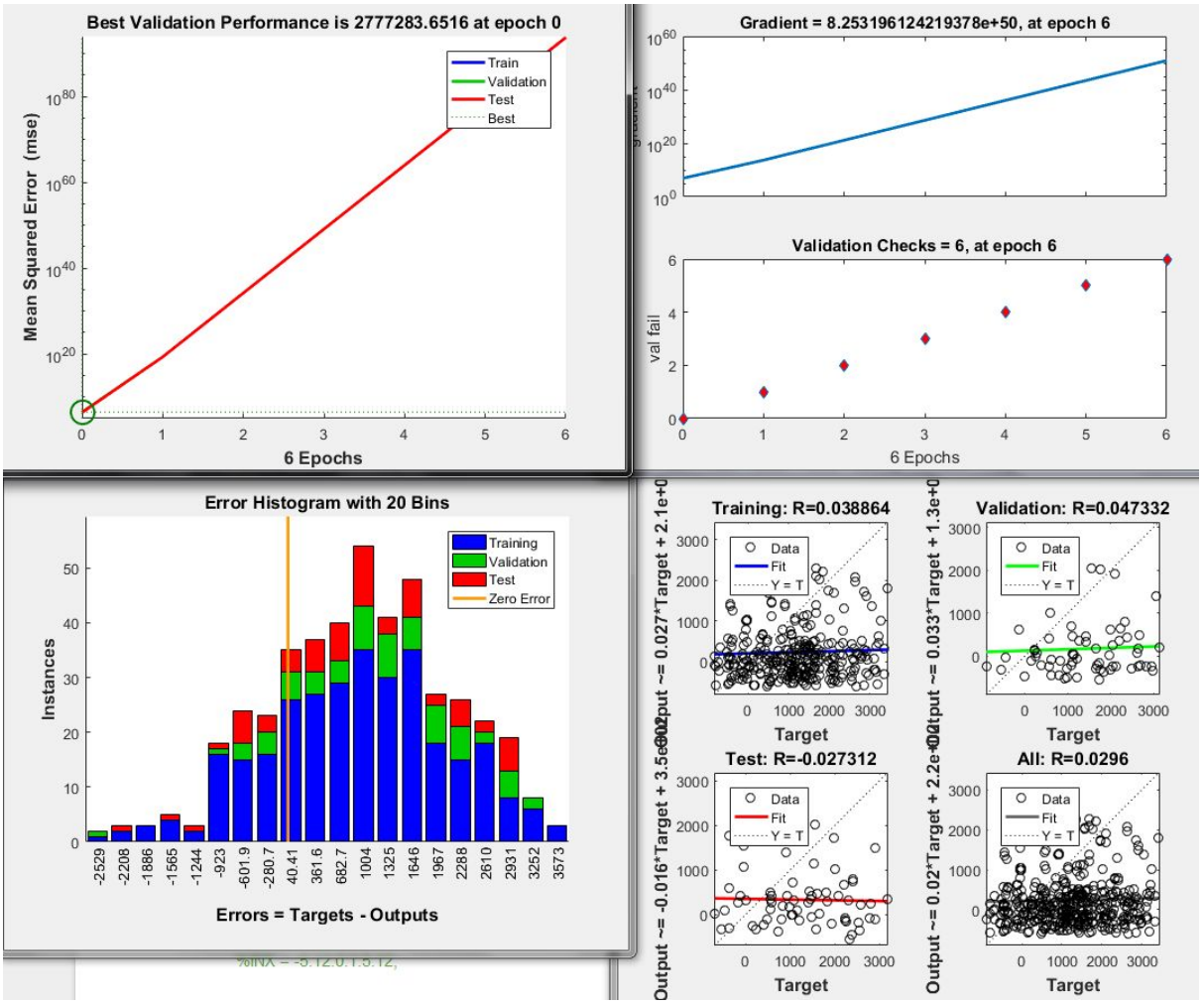
współczynnik uczenia = 0.000001, bezwładność = 0.5



współczynnik uczenia = 0.01, bezwładność = 1



współczynnik uczenia = 0.5, bezwładność = 1



4) Wnioski:

Sieci wielowarstwowe z większą liczbą neuronów i warstw ukrytych zazwyczaj uczą się lepiej, do momentu załamania, w którym sieć może zostać przeuczona. Większy współczynnik uczenia powoduje wzrost szybkości nauki, lecz dokładność wyników zaczyna się obniżać, prowadząc do zwiększenia błędu.

5) Listing kodu:

p3.m:

```
close all; clear all; clc;
INX = -2:0.2:2;
%INX = -5.12:0.1:5.12;
INY = -1*transpose(INX);

TEST = zeros(length(INX));
INPUT = [1:(length(INX)^2);1:(length(INX)^2)];
WANTED = [1:(length(INX)^2)];
for ky=1:length(INY)
        for kx=1:length(INX)
```

```
                INPUT(1,(ky-1)*length(INX)+kx)=INX(kx);
                INPUT(2,(ky-1)*length(INX)+kx)=INY(ky);
                WANTED((ky-1)*length(INX)+kx)=RastrignTest3D(INX(kx),INY(ky));
        end
end

net = feedforwardnet(5);
net.trainFcn = 'traingd'; %algorytm wstecznej propagacji
net.trainParam.lr = 0.000001; %wsp. uczenia
net.trainParam.mc = 0.5; %bezwladnosc
net = train(net, INPUT, WANTED);

OUTPUT=sim(net,INPUT);
OUT=zeros(length(INX));
OUTPUT2=rastrigin_neural2(INPUT);
OUT2=zeros(length(INX));
for ky=1:length(INY)
        for kx=1:length(INX)
                TEST(kx,ky)=RastrignTest3D(INX(kx),INY(ky));
                OUT(kx,ky)=OUTPUT((ky-1)*length(INX)+kx);
                OUT2(kx,ky)=OUTPUT2((ky-1)*length(INX)+kx);
        end
end
```

RastriginTest3D.m:

```
function f = RastrignTest3D(x,y)
        A=10;
        n=100;
        x1=x;
        y1=y;
        dx=abs(5.12-x)/n;
        dy=abs(5.12-y)/n;
        f=A*n;
        for i=1:n
                x=x1+(dx);
                y=y1+(dy);
                f=f+(x^2)-(A*cos(2*pi*x))+(y^2)-(A*cos(2*pi*y));
        end
end
```