

Uniwersytet Zielonogórski	Wykonali:	Grupa:	Nr ćwiczenia:	Ocena:
Laboratorium				
Temat ćwiczenia: Sieci RBF.		Prowadzący:	Data wyk. ćw.	Data odd. spr.

1. W zadaniu pierwszym dokonaliśmy aproksymacji punktów na płaszczyźnie. Eksperymentalnie dobraliśmy liczbę neuronów, która daje satysfakcjonujące efekty aproksymujące. Z doświadczeń wynika, że zastosowanie 8 neuronów radialnych daje zadowalające wyniki. Przy bardzo dużej liczbie wzorców uczących i równej im liczbie funkcji radialnych problem uczenia może stać się przewymiarowany. Wykorzystany skrypt i otrzymane wyniki znajdują się poniżej:

```
X = [-1.0,-0.9,-0.8,-0.7,-0.6,-0.5,-0.4,-0.3,-0.2,-0.1,0, 0.1,0.2, 0.3, 0.4, 0.5, 0.6, 0.7,
0.8, 0.9, 1.0];
Y = [-.9602,-.5770,-.0729, .3771,.6405, .6600, .4609, .1336, -.2013, -.4344, -.5000, -.3930, -
.1647, .0988, .3072,.3960, .3449, .1816,-.0312,-.2189,-.3201];
```

```
net=newrb(X,Y,0,1,8,2);
```

```
odp=sim(net,X);
blad=mse(odp-Y)
```

```
plot(X,Y,'*r',X,odp,'-g');
```

blad=3.4979e-006

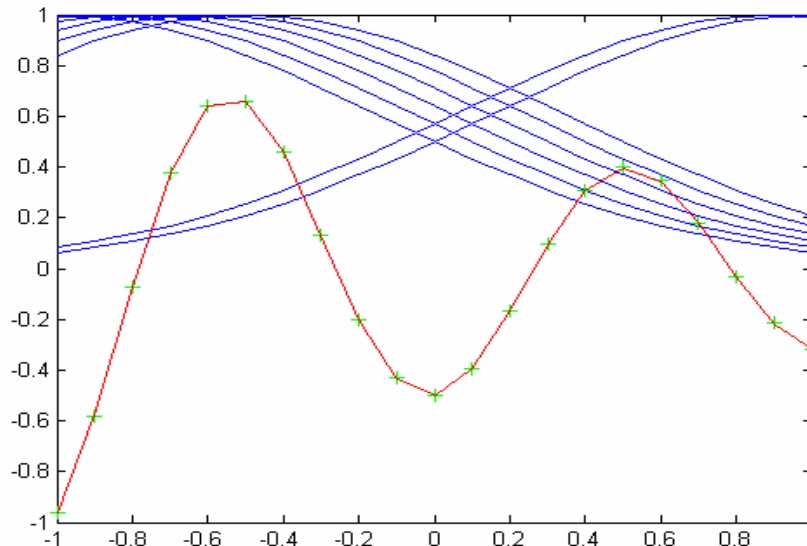
Dodatkowo dla stworzonej sieci, dla każdego neuronu radialnego wyznaczyliśmy wartości wyjść. Następnie wykonaliśmy sumowanie ważone. Wykorzystany skrypt i otrzymane wyniki zawarte są poniżej:

```
for i=1:length(net.IW{1}),
    f(i,:)=radbas((net.IW{1}(i).*ones(1,21)-X).*net.b{1}(1));
    plot(X,f(i,:));
    hold on,
end

sum=zeros(1,length(X));

for j=1:length(net.LW{2})
    sum=sum+f(j,:).*net.LW{2}(j);
end
sum=sum+net.b{2};

plot(X,sum,'+g');
plot(X,Y,'-r');
```



Rysunek 1. Wartości wyjść neuronów radialnych i wyjścia sieci.

2. W kolejnym zadaniu dokonaliśmy aproksymacji funkcji $z=x^2 +y^2$ przy pomocy sieci radialnej. Dane wejściowe podzieliliśmy na zbiór uczący i testujący. Dla losowych wartości danych uczących przeprowadziliśmy uczenie sieci, działanie sieci sprawdziliśmy stosując zbiór testujący. Zastosowaliśmy sieć o 200 neuronach radialnych. Wykorzystany skrypt oraz otrzymane wyniki przedstawione są poniżej:

```
xu=sort(10*rand(1,25)-5);
yu=sort(10*rand(1,25)-5);
[Xu,Yu]=meshgrid(xu,yu);
Zu=Xu.^2+Yu.^2;

xt=-5:0.1:5;
yt=xt;
[Xt,Yt]=meshgrid(xt,yt);
Zt=Xt.^2+Yt.^2;
dane_t=[Xt(:)';Yt(:)'];

dane_u=[Xu(:)';Yu(:)'];

net=newrb(dane_u,Zu(:)',0,1,200,20);

odp=sim(net,dane_u);
blad_uczenia=mse(odp-Zu(:)')

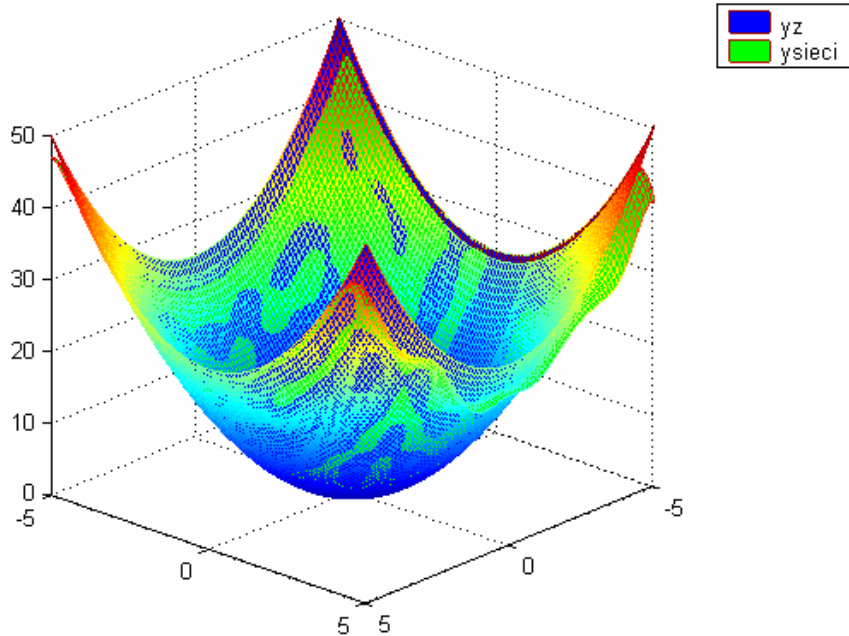
odp_t=sim(net,dane_t);
blad_testowania=mse(odp_t-Zt(:)')

figure
mesh(Xt,Yt,Zt,'FaceColor',[0 0 1])
hold on
mesh(Xt,Yt,reshape(odp_t,length(xt),length(yt)),'FaceColor',[0 1 0])
```

Wyniki:

blad_uczenia =1.4223e-004

blad_testowania =0.7797



Rysunek 2. Wykres wartości oczekiwanej i otrzymanej w wyniku symulacji.

Okazuje się, że sieć radialna nadaje się do realizacji zadanie aproksymacji. Uzyskane wyniki pokazują, że aby dokładność sieci była satysfakcjonująca, trzeba zastosować dużą liczbę neuronów. Duża ilość neuronów wpływa oczywiście na wydłużenie procesu uczenia.

3. Ostatnie zadanie polegało na napisaniu skryptu realizującego uczenie sieci RBF. Centra wyznaczyliśmy losowo, rozpiętości obliczyliśmy ze wzoru $b = \frac{d_{\max}}{\sqrt{N}}$. Wagi wyjściowe dostroiliśmy stosując regułę delty. Do sprawdzenia działania skryptu wykorzystaliśmy zadanie aproksymacji funkcji: $y=x^2-3$ oraz $y=x^3+3$. Poniżej zamieściliśmy skrypt oraz otrzymane wyniki:

```
x=-2:.1:2;%sygnal wejsciowy
y=x.^2-3;%zadany sygnal wyjsciowy

%%warstwa ukryta

liczba_neuronow=input('Podaj liczbe neuronow ');
w=4*rand(1,liczba_neuronow)-2;%wagi wejsciowe
b=max(max(dist(w)))/sqrt(liczba_neuronow);%rozpietosc

for i = 1:liczba_neuronow,
    n=ones(1,length(x)).*w(i)-x;
    n1=n.*b;
    fi(i,:)=radbas(n1);
end

%%warstwa wyjsciowa

W=initp(fi,y);%wagi poczatkowe
licznik=0;
n=0.02;%krok uczenia
liczba_krokow=500;%maksymalna liczba krokow

while licznik<liczba_krokow
    licznik=licznik+1;
    teta=W*fi;%potencjaly membranowe
    Ys=purelin(teta);%wyjście sieci
    E=y-Ys;%porównanie wyjścia sieci z wzorcem
```

```

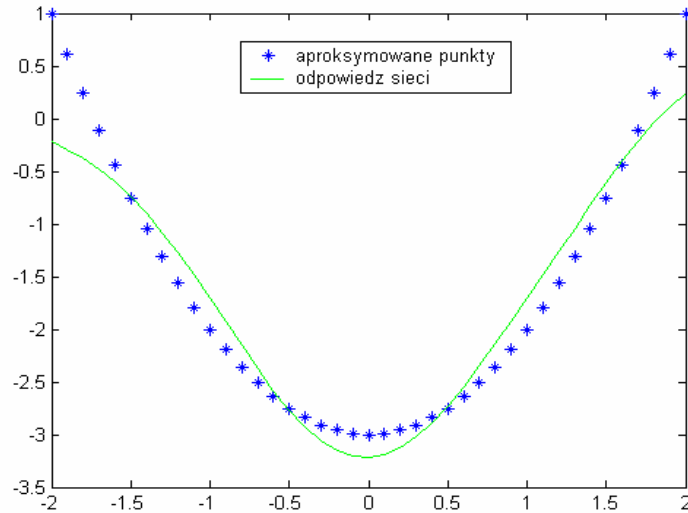
mse(E);
if(mse(E)<0.001) break; %warunek jakosciowy
end;
dW=(E*fi');%wyznaczenie korekty wag
W=W+n*dW;
end

blad=mse(E)
plot(x,y,'*');hold on
plot(x,Ys,'-g');

```

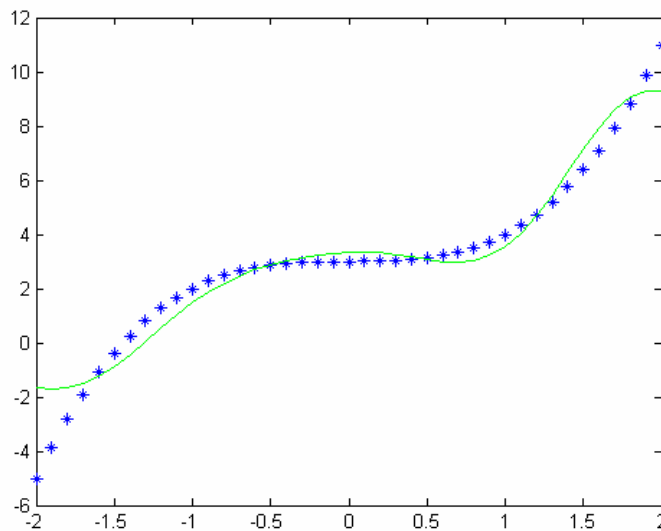
Wyniki

- a) $y=x^2-3$, 6 neuronów radialnych
 blad =0.1226



Rysunek 3.

- b) $y=x^3+3$, 8 neuronów radialnych
 blad= 0.6610;



Rysunek 4.