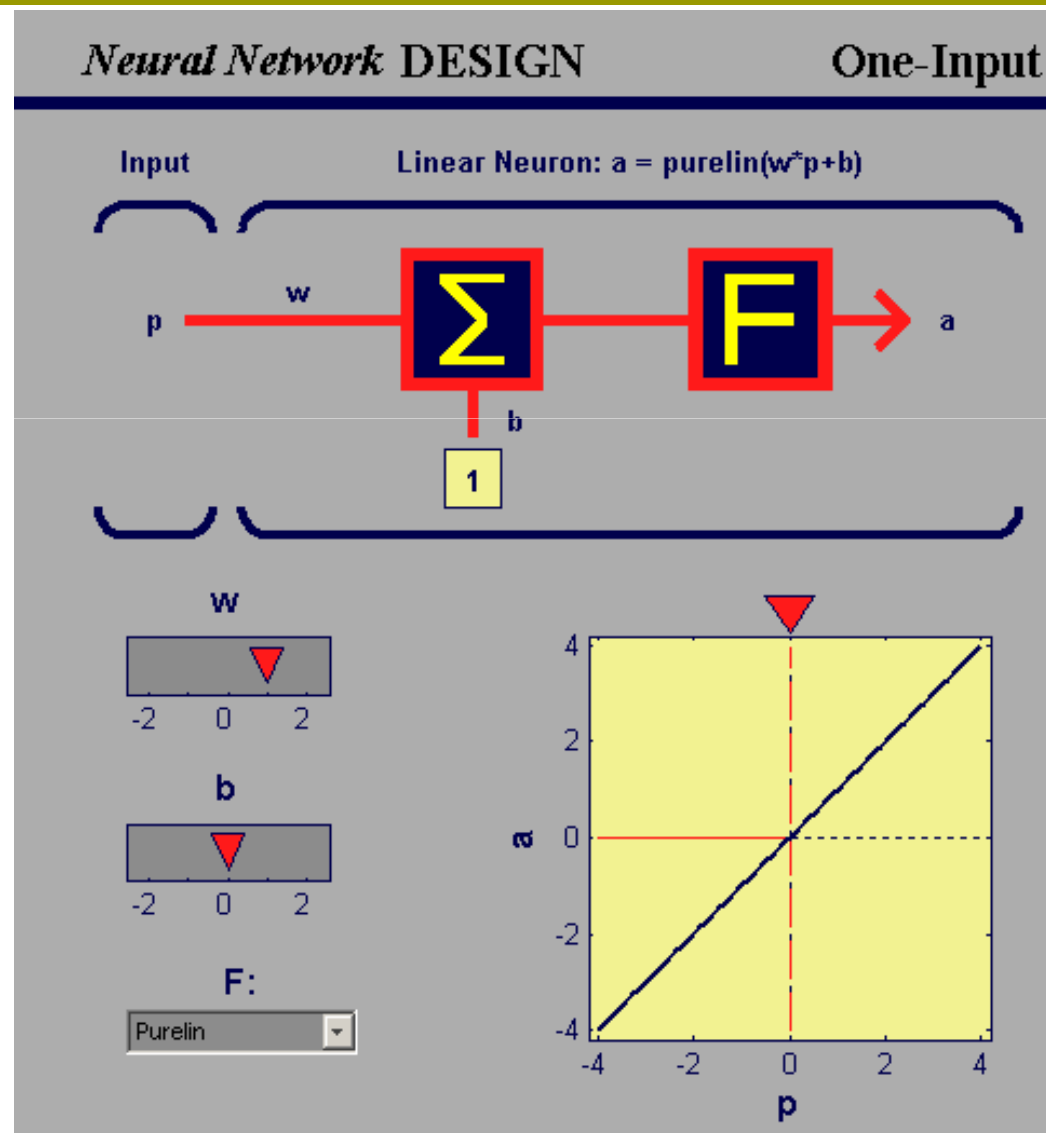


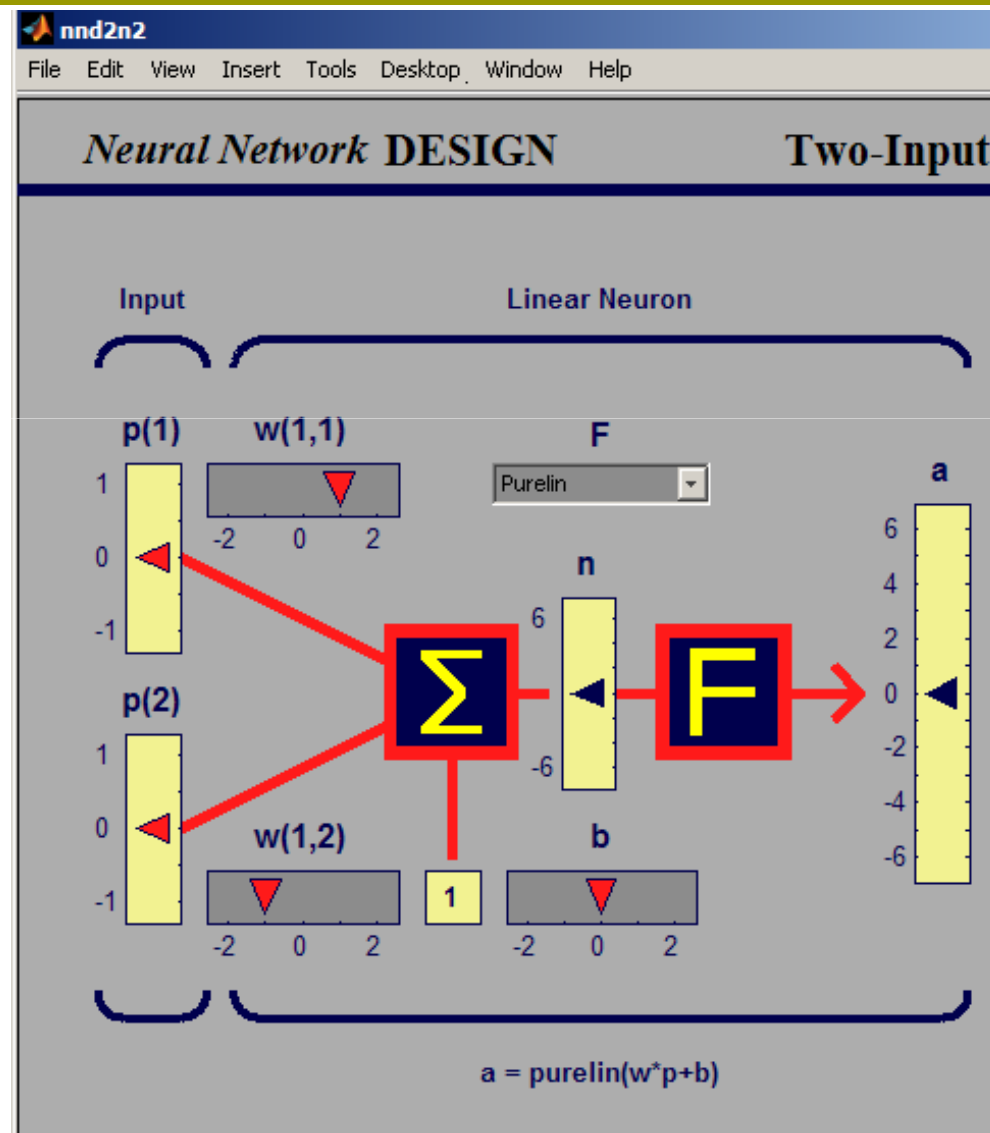
# Neurális hálózatok és a MATLAB



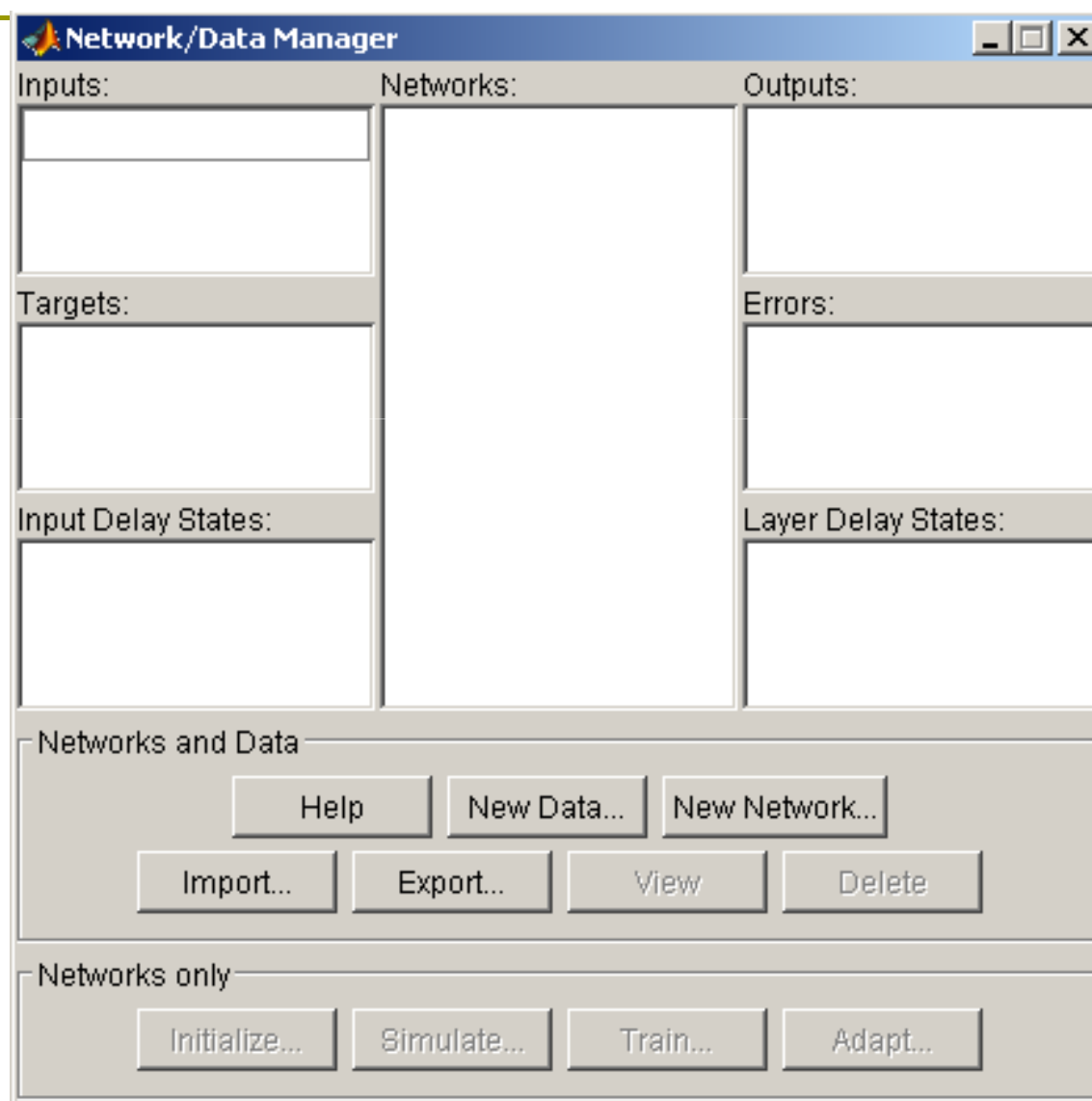
# Neuron modell (1 input)



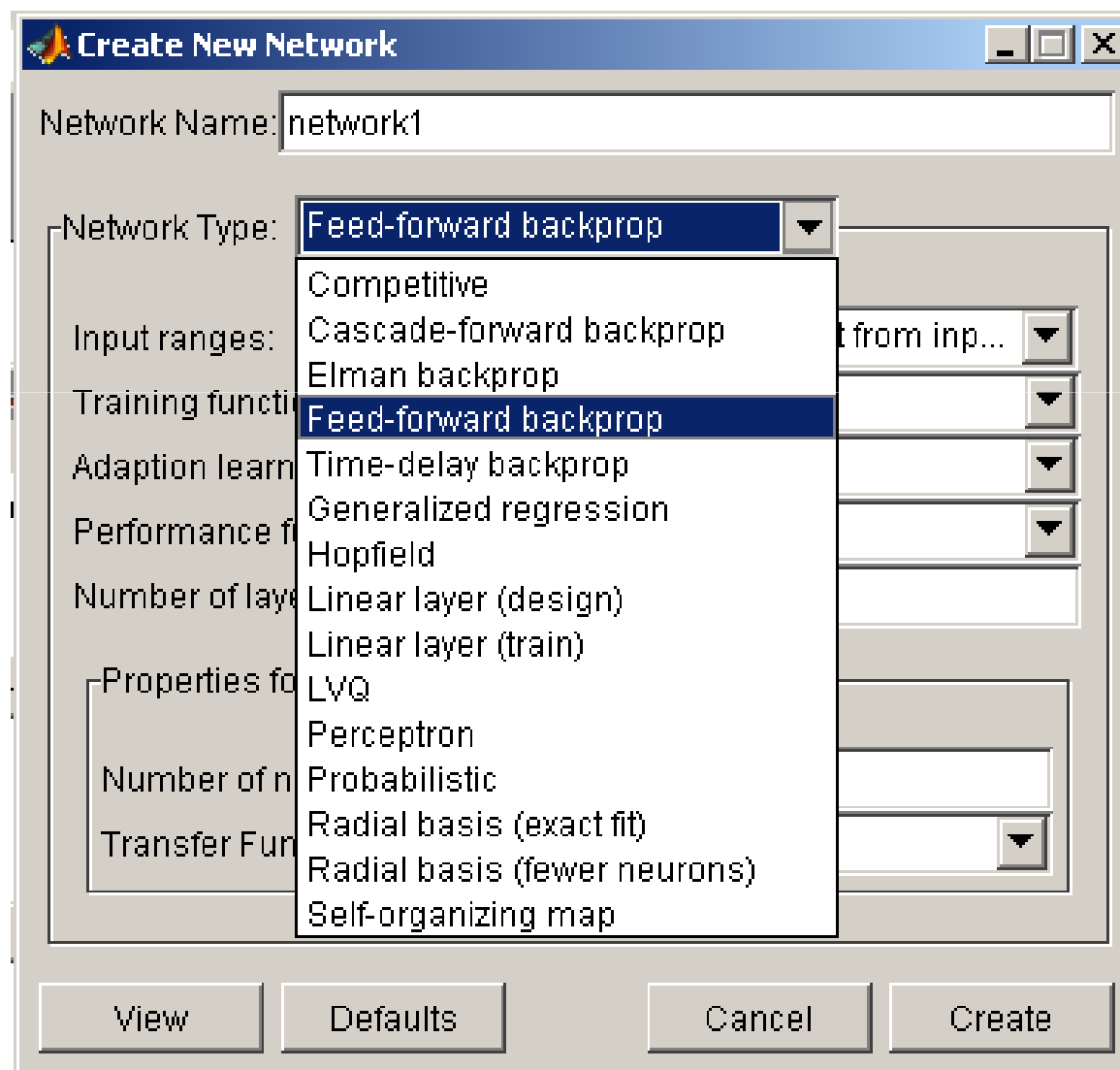
# Két input:



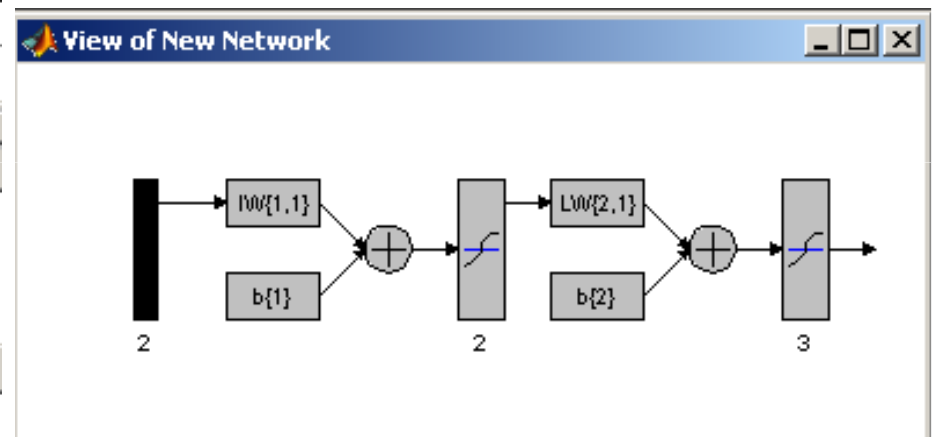
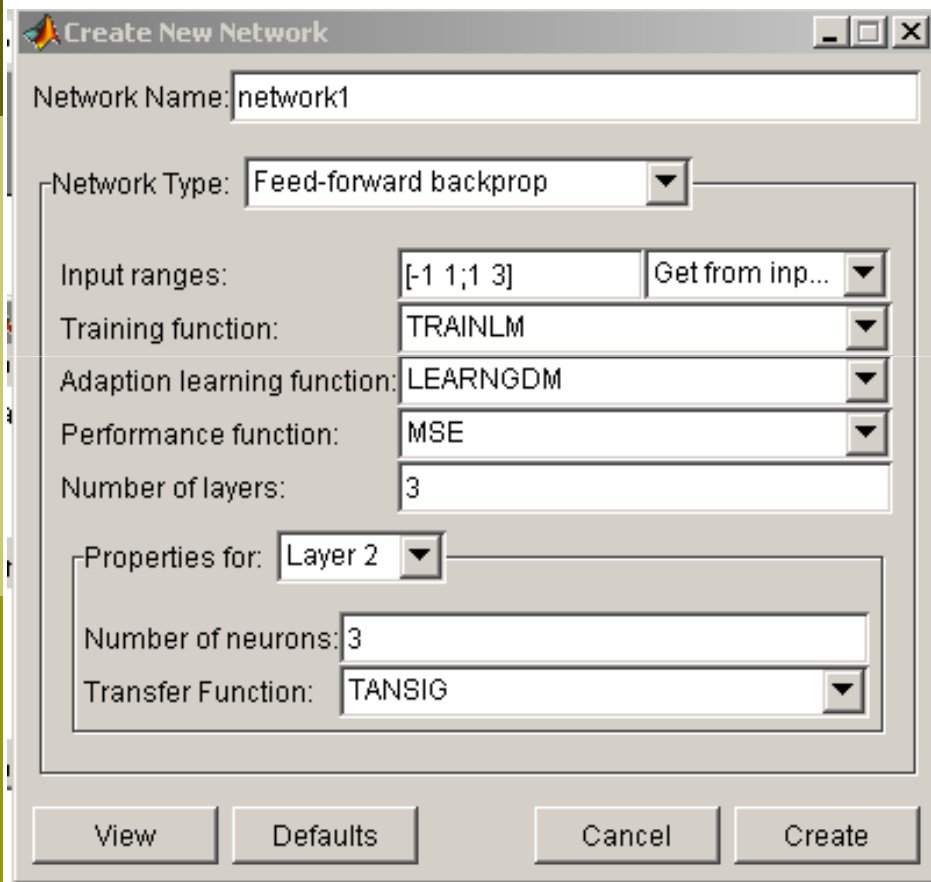
>> nntool



# Hálózat típusának kiválasztása



# Generált perceptron hálózat



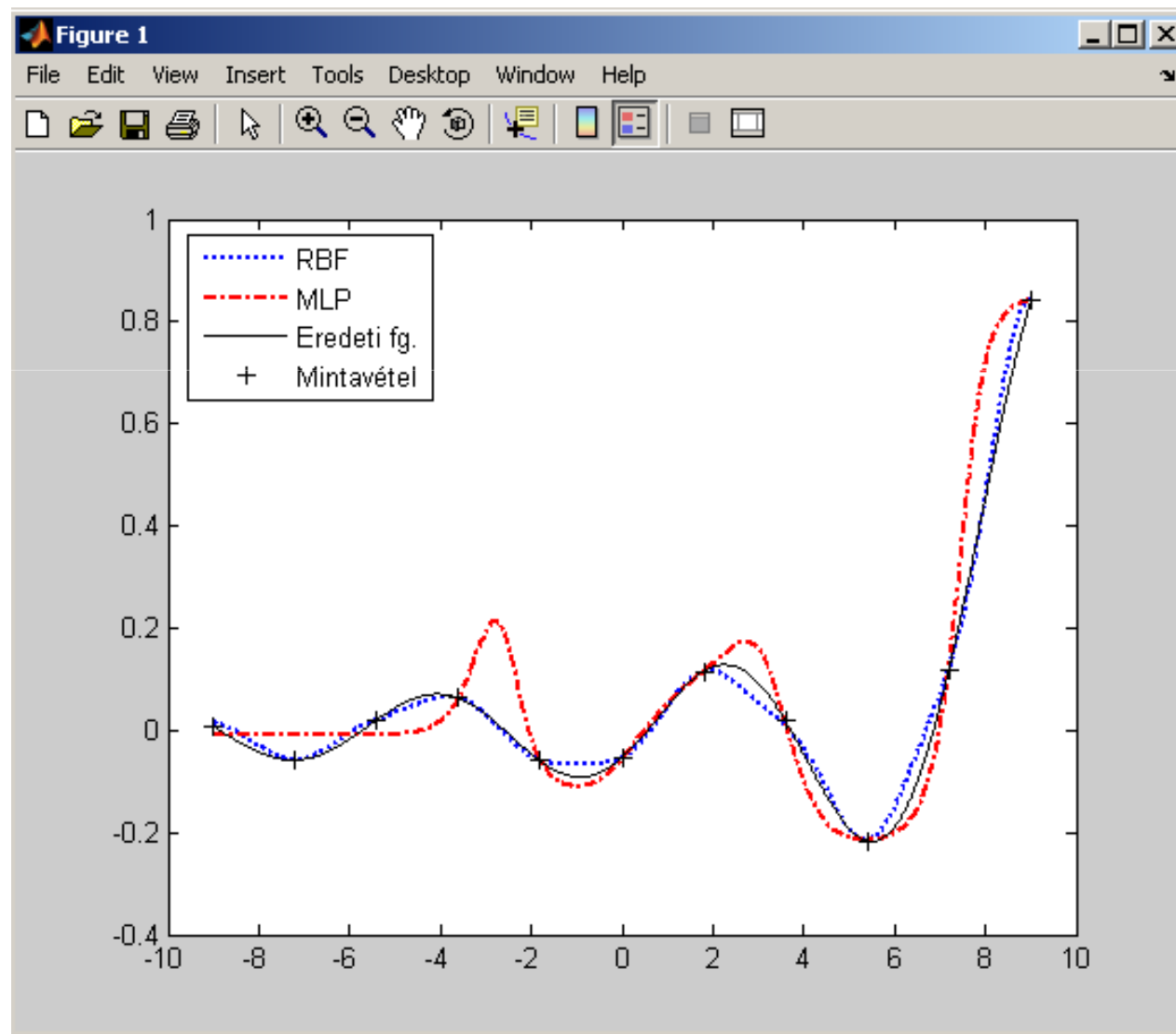
# Közelítsük a $\sin(x-10)/(x-10)$ függvényt MLP hálózattal!

## Hasonlítsuk össze az RBF (Radial Basis Function Network) hálózattal!

---

```
clear all;
xe=-9; %intervallum eleje
xv=9; %intervallum vége
psz=10; %mintavételi pontok száma -1
%eredeti függvény
X=xe:0.01:xv;Y=sin(X-10)./(X-10);
%tanulópontok
iv=xe:(abs(xe)+abs(xv))/psz:xv; Pontok=sin(iv-10)./(iv-10);
%RBF
eg=0.001;sc=1;netRBF=newrb(iv,Pontok,eg,sc,11,1);
Yrbf=sim(netRBF,X);
%MLP
netMLP=newff(minmax(iv),[5,5,1],...
    {'logsig','tansig','tansig'},'traincgf');
netMLP.trainParam.show=10;
netMLP.trainParam.epochs=500;
netMLP.trainParam.goal=1e-5;
[netMLP,tr]=train(netMLP,iv,Pontok);Ymlp=sim(netMLP,X);
%kirajzolás
figure;plot(X,Yrbf,'b:','LineWidth',2);hold on;
plot(X,Ymlp,'r-.','LineWidth',2); hold on;
plot(X,Y,'k-',iv,Pontok,'k+');
legend('RBF','MLP','Eredeti fg.','Mintavétel',2);
```

# A kapott eredmény:





# Generáljunk mintát két normális eloszlásból!

## Osztályozzuk a pontokat MLP hálózattal!

---

### **%mlposzt.m**

```
clear all;
```

```
szoras=diag([0.8 0.8]); Mdb=100;
```

```
Minta=mvnrnd([0 0]',szoras, Mdb);
```

```
Osztaly(1:Mdb,1)=1;
```

```
Minta(Mdb+1:2*Mdb,:)=mvnrnd([2 2]',szoras,Mdb);
```

```
Osztaly(Mdb+1:2*Mdb,1)=-1;
```

### **%MLP**

```
netMLP=newff(minmax(Minta'),[12,8,1],...  
    {'tansig' 'tansig' 'tansig'}, 'traincgf');
```

```
netMLP.trainParam.show=10;
```

```
netMLP.trainParam.epochs=300;
```

```
netMLP.trainParam.goal=1e-3;
```

```
[netMLP,tr]=train(netMLP,Minta',Osztaly');
```

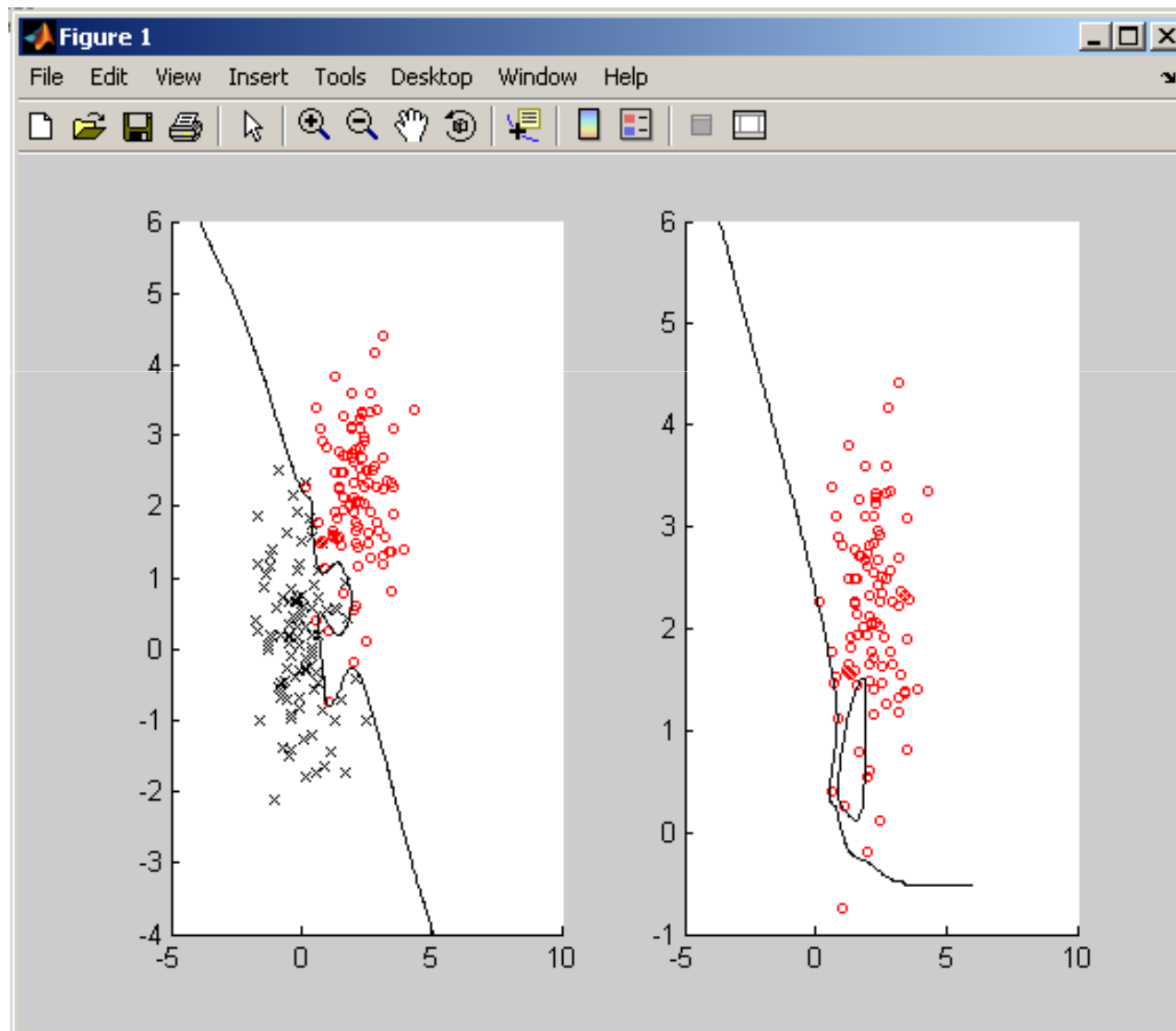
# Folytatás:

---

## **%MLP1**

```
netMLP1=newff(minmax(Minta'),[6,4,2,1],...
    {'logsig' 'logsig' 'tansig' 'tansig'}, 'traincgf');
netMLP1.trainParam.show=10;
netMLP1.trainParam.epochs=300;
netMLP1.trainParam.goal=1e-2;
[netMLP1,tr]=train(netMLP1,Minta',Osztaly');
%
a=[-4 6 -4 6]; [X,Y]=meshgrid(a(1):0.05:a(2),a(3):0.05:a(4));
X2=[reshape(X,prod(size(X)),1) reshape(Y,prod(size(X)),1)];
z=sim(netMLP,X2'); z=reshape(z,size(X));
z1=sim(netMLP1,X2'); z1=reshape(z1,size(X));
figure; %kirajzolás
subplot(1,2,1); scatter(Minta(1:Mdb,1), Minta(1:Mdb,2),20,'kx');
hold on; ...
scatter(Minta(Mdb+1:2*Mdb,1),Minta(Mdb+1:2*Mdb,2),20,'r');
contour(X,Y,z,[+0 +0],'k'); hold on; subplot(1,2,2);
scatter(Minta(1>Mdb,1),Minta(1>Mdb,2),20,'kx');hold on;
scatter(Minta(Mdb+1:2*Mdb,1),Minta(Mdb+1:2*Mdb,2),20,'r');
contour(X,Y,z1,[+0 +0],'k');
```

# A kapott eredmény:



# Közelítsük a $\sin(x-10)/(x-10)$ függvényt RBF-hálózattal!

---

## **%rbffv.m**

xe=-9;%intervallum eleje

xv=9; %intervallum vége

psz=10; %mintavételi pontok száma

## **%eredeti fv**

X=xe:0.01:xv; Y=sin(X-10)./(X-10);

## **%tanulópontok**

iv=xe:(abs(xe)+abs(xv))/psz:xv;Pontok=sin(iv-10)./(iv-10);

## **%RBF1**

eg=0.001;sc=.4;

netRBF1=newrb(iv, Pontok,eg,sc,psz+1,1);Yrbf1=sim(netRBF1,X);

## **%RBF2**

eg=0.001;sc=3;

netRBF2=newrb(iv,Pontok,eg,sc,5,1);Yrbf2=sim(netRBF2,X);

## **%kirajzolás**

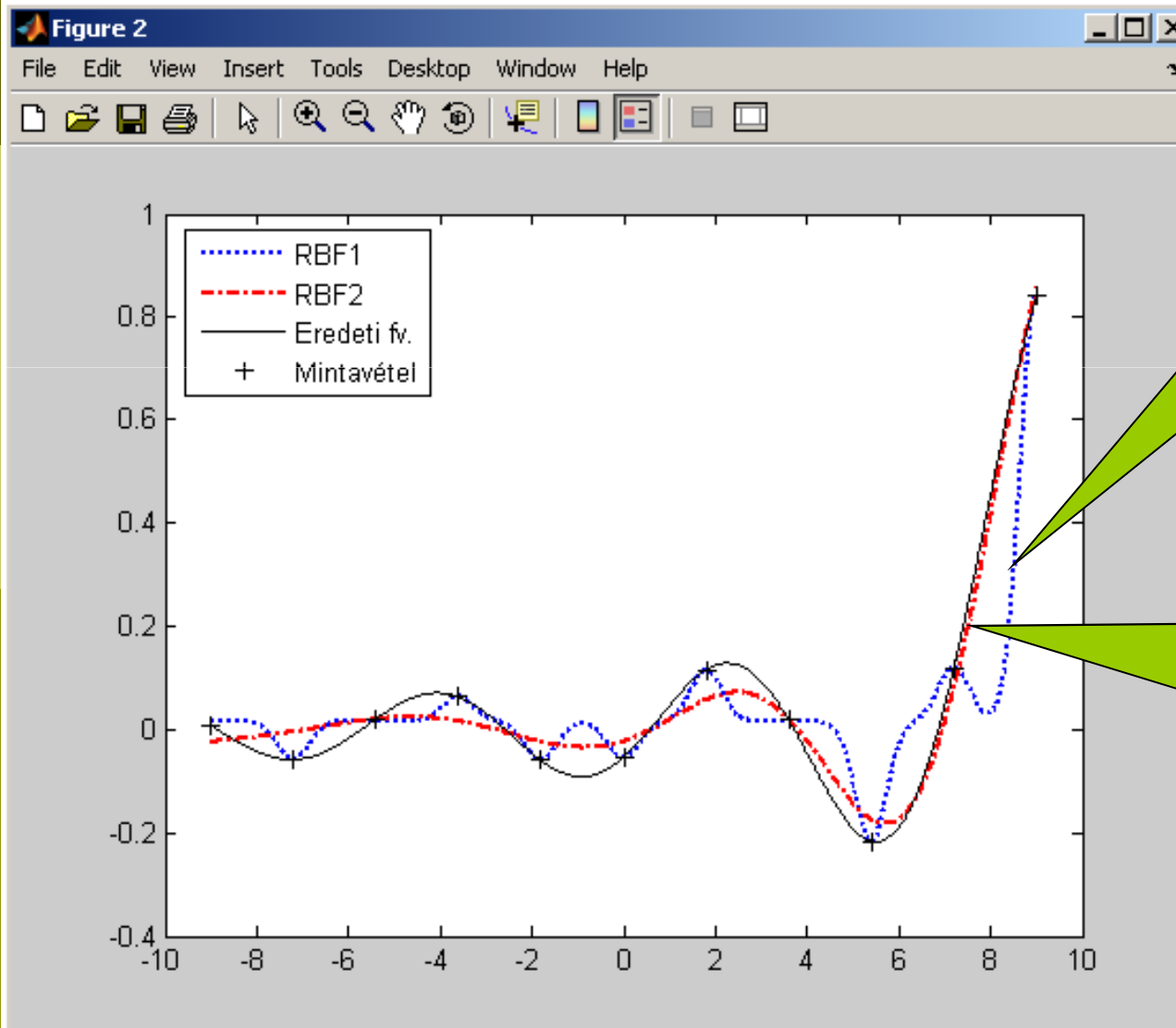
figure; plot(X,Yrbf1,'b:','LineWidth',2);hold on;

plot(X,Yrbf2,'r-.','LineWidth',2);hold on;

plot(X,Y,'k-',iv,Pontok,'k+');

legend('RBF1','RBF2','Eredeti fv.','Mintavétel',2);

# Eredmény:



A kis sávszélesség és a sok neuron meglehetősen ugráló közelítést ad

A nagy sávszélesség és a kevés neuron a függvény egy „átlagolását” eredményezi

# Generáljunk mintát két normális eloszlásból!

## Osztályozzuk a pontokat RBF hálózattal!

---

### **%rbfoszt.m**

```
clear all;
```

```
szoras=diag([0.8 0.8]); Mdb=100;
```

```
Minta=mvnrnd([0 0 ],szoras,Mdb);
```

```
Osztaly(1>Mdb,1)=1;
```

```
Minta(Mdb+1:2*Mdb,:)=mvnrnd([2 2]',szoras,Mdb);
```

```
Osztaly(Mdb+1:2*Mdb,1)=-1;
```

```
eg=0.001;sc=.5;
```

```
net1=newrb(Minta',Osztaly',eg,sc,100);
```

### **%kirajzolás**

```
figure;a=[-4 6 -4 6];
```

```
[X,Y]=meshgrid(a(1):0.05:a(2),a(3):0.05:a(4));
```

```
X2=[reshape(X,prod(size(X)),1) reshape(Y,prod(size(X)),1)];
```

```
z1=sim(net1,X2'); z1=reshape(z1,size(X));
```

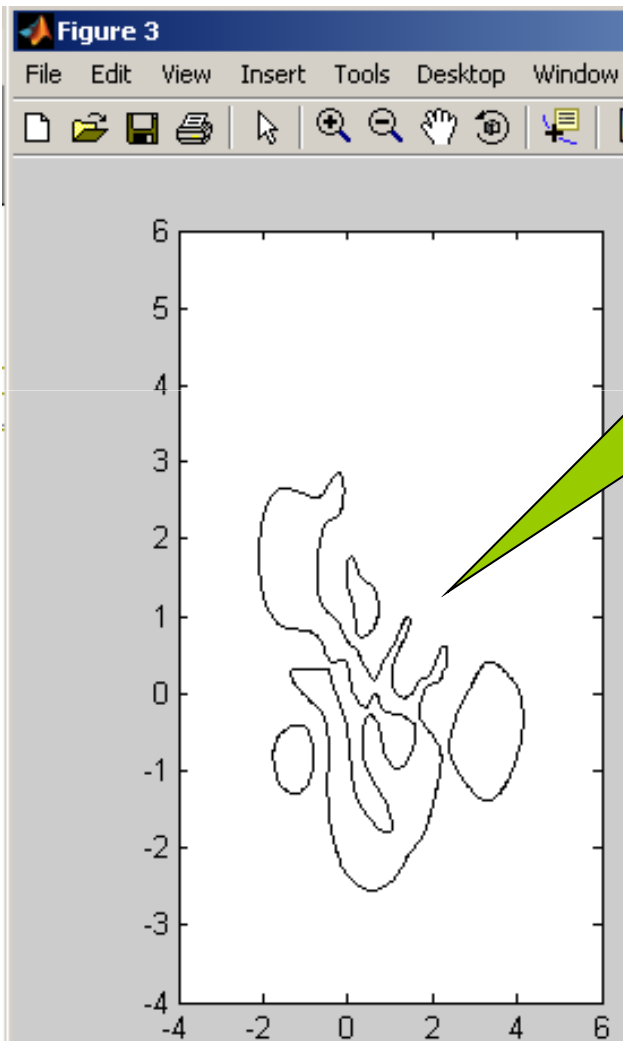
```
hold off;subplot(1,2,1);
```

```
scatter(Minta(Mdb+1:2*Mdb,1),Minta(Mdb+1:2*Mdb,2),5,'r');
```

```
contour(X,Y,z1,[+0 +0],'k');
```

```
net1=newrb(Minta',Osztaly',0.001,1.5,30);
```

# Eredmények:



A kis sávszélesség és a sok neuron összetett és kacskaringós határu tartományokat ad

A nagy sávszélesség és a kevés neuron „nagyvonalúan” osztályoz

