Genetic Optimization

- RBF (Radial Basis Function)
- MATLAB programming

Example

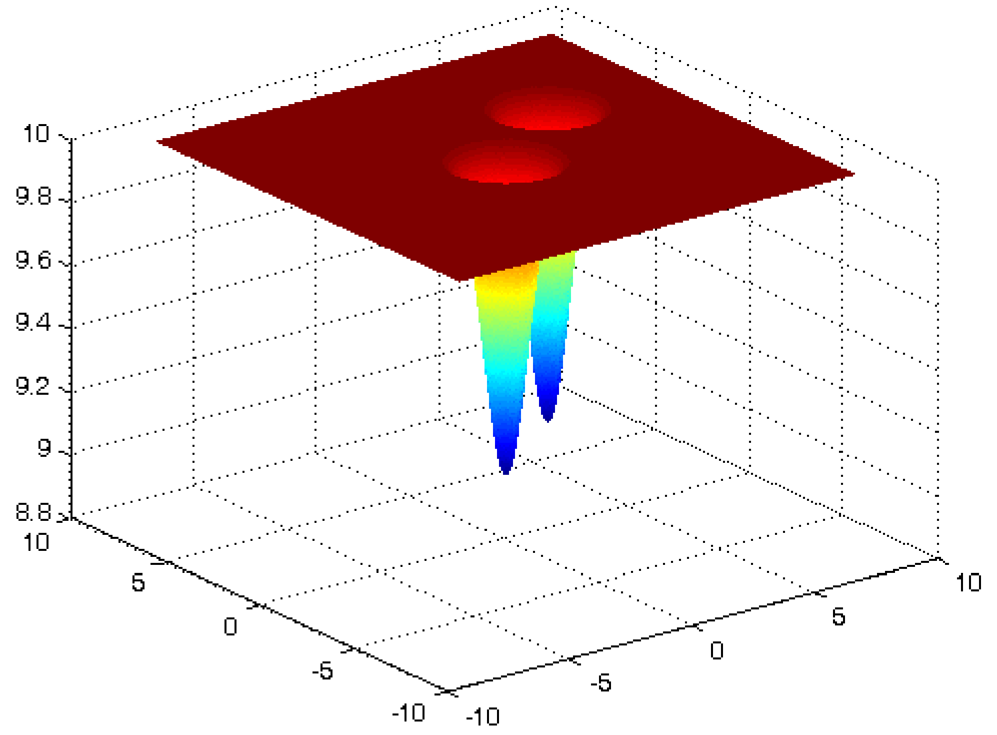
```
function z=my_fun2(x)

z=10-exp(-(x(1).^2+x(2).^2))-exp(-(x(1)-3).^2-(x(2)-4).^2);
```

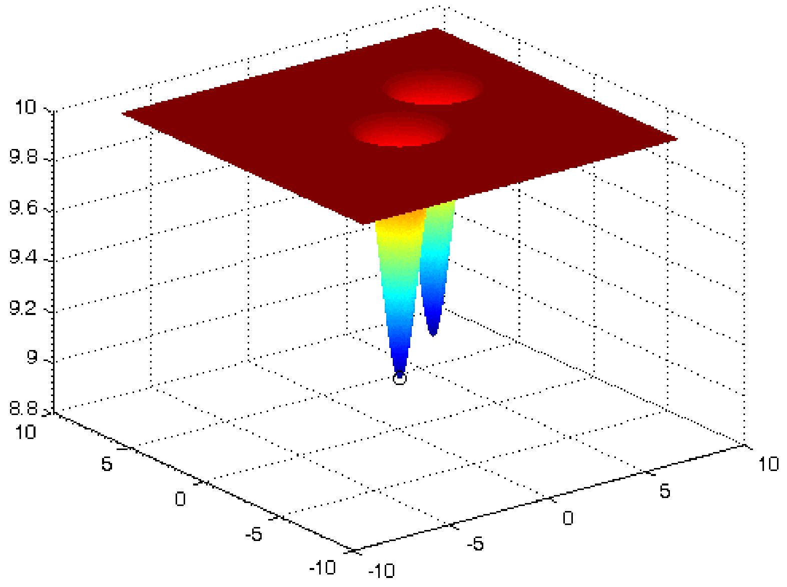
my_fun2.m

Figure

plot_myfun2.m



```
[x_min fval] = ga(@my_fun2,2);
x_min
fval
my_fun2(x_min)
hold on; plot3(x_min(1),x_min(2),fval,'ok');
```

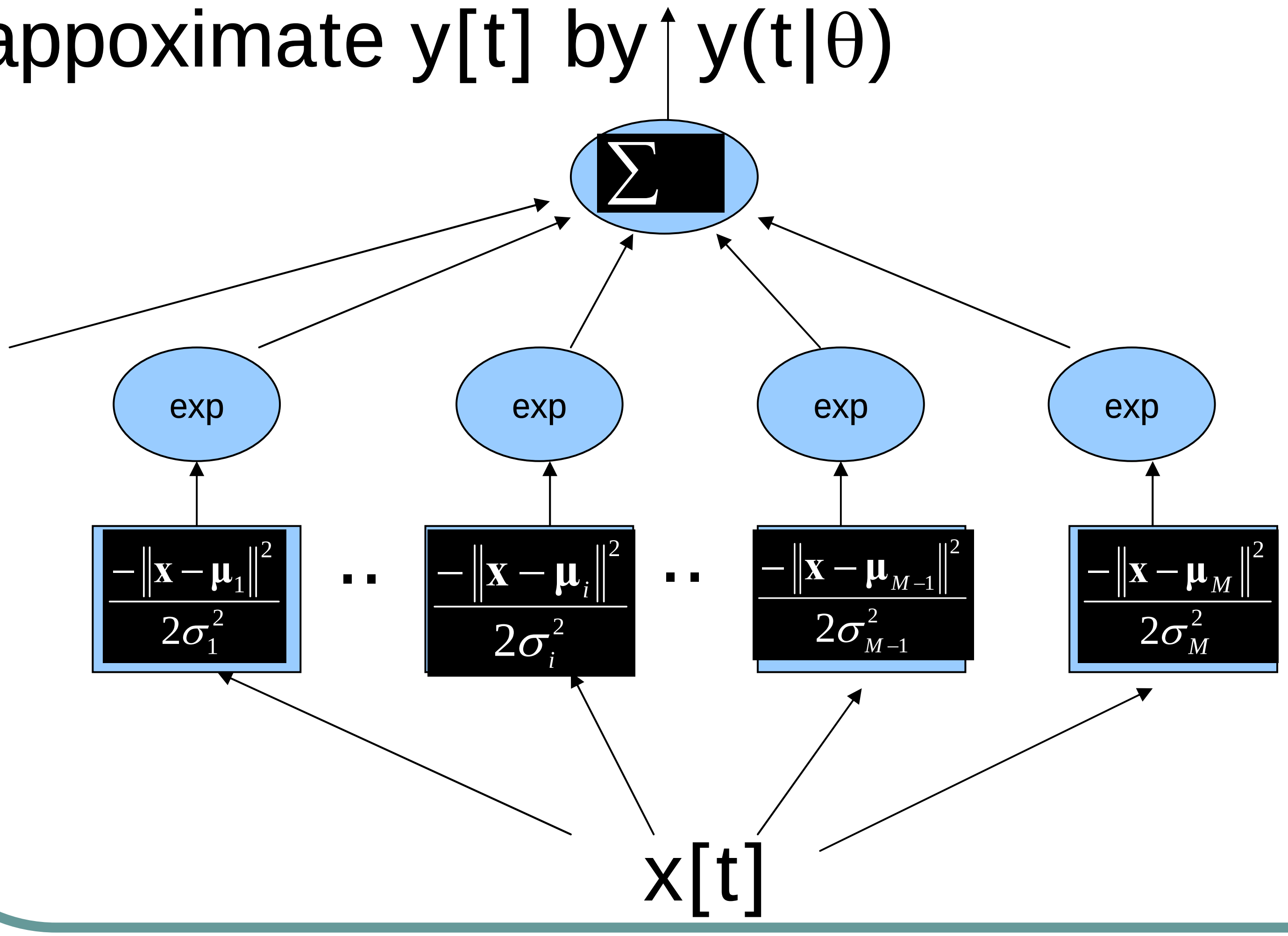


Learning RBF

- Derive an objective function
- Apply GA optimization to minimize the objective function

Requirement

appoximate y[t] by y(t|θ)



$\sum$
exp
exp
exp
exp
$\frac{-\|\mathbf{x}-\boldsymbol{\mu}_1\|^2}{2\sigma_1^2}$
..
$\frac{-\|\mathbf{x}-\boldsymbol{\mu}_i\|^2}{2\sigma_i^2}$
..
$\frac{-\|\mathbf{x}-\boldsymbol{\mu}_{M-1}\|^2}{2\sigma_{M-1}^2}$
$\frac{-\|\mathbf{x}-\boldsymbol{\mu}_M\|^2}{2\sigma_M^2}$
x[t]

RBF Network function

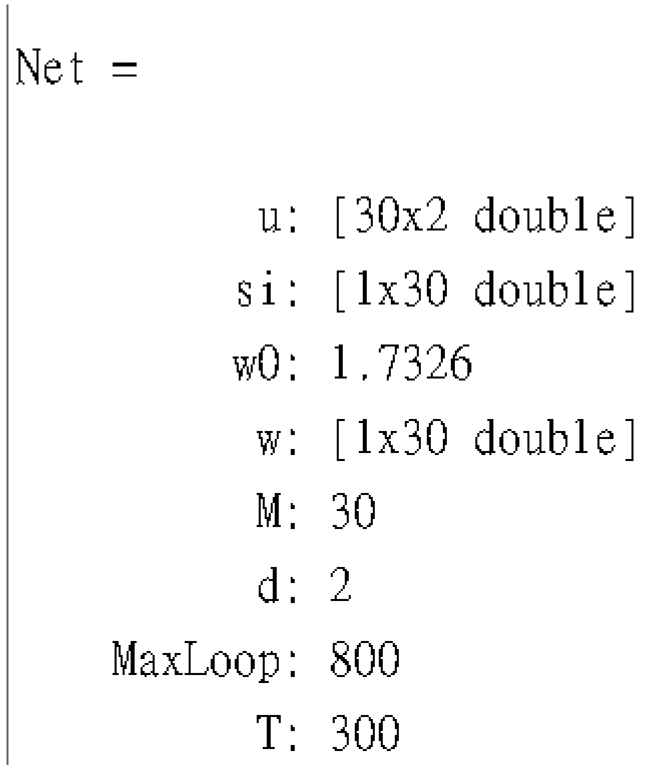
$$y(t \mid \theta) = G(\mathbf{x}[t] \mid \theta)$$

$$= w_0 + \sum_{m=1}^{M} w_m \exp\left(-\frac{\left\|\mathbf{x}[t] - \boldsymbol{\mu}_m\right\|^2}{2\sigma_m^2}\right)$$

Network parameter

$$\theta = \{\mathrm{w}_{\mathrm{i}}\}_i \cup \{\boldsymbol{\mu}_i\}_i \cup \{\sigma_i\}_i$$

Data Structure



```
Net =

          u: [30x2 double]
         si: [1x30 double]
         w0: 1.7326
          w: [1x30 double]
          M: 30
          d: 2
    MaxLoop: 800
          T: 300
```

Matlab functions

- Net2x
- x2Net
- evaRBF
- myfun

Net2x

- Input: Net
- Translate an RBF Net to a vector

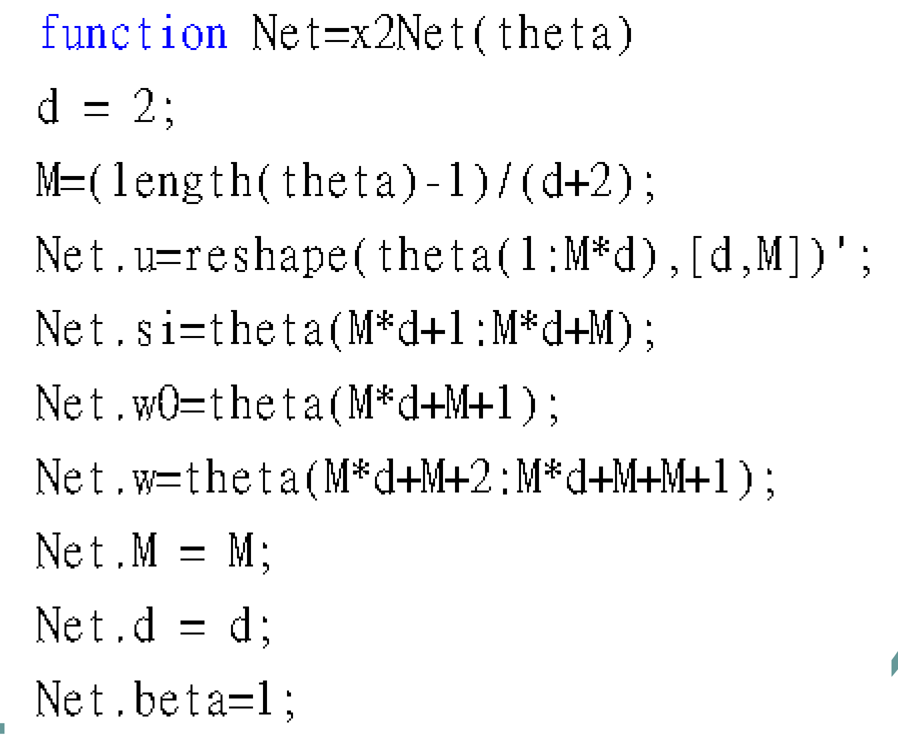
Net2x.m

```
function theta=Net2x(Net)
d=2;
M=Net.M;
L = M*d+2*M+1;
theta(1:M*d)=reshape(Net.u',1,M*d);
theta(M*d+1:M*d+M)=Net.si;
theta(M*d+M+1)=Net.w0;
theta(M*d+M+2:M*d+M+M+1)=Net.w;
```

x2Net

- Input : x
- Translate x to an RBF Net

x2Net.m



```
function Net=x2Net(theta)
d = 2;
M=(length(theta)-1)/(d+2);
Net.u=reshape(theta(1:M*d),[d,M])';
Net.si=theta(M*d+1:M*d+M);
Net.w0=theta(M*d+M+1);
Net.w=theta(M*d+M+2:M*d+M+M+1);
Net.M = M;
Net.d = d;
Net.beta=1;
```

evaRBF

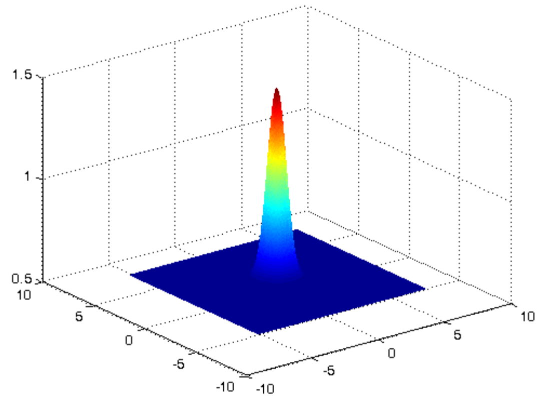
- Evaluate an RBF Net

evaRBF.m

```
function [y,D]=evaRBF(x,net)
% RBF evaluatioin
M=net.M; u=net.u; N=size(x,1);
D=cross_distance(x,u);
s=2*repmat(net.si.^2,N,1);
w=net.w;
y=sum(exp(-D./s).*repmat(w,N,1),2)+net.w0;
y=y';
return
```

Example: A random RBF mapping

```
M=2;d=2;
theta2=rand(1,M*d+2*M+1);
Net2=x2Net(theta2);
figure
x1=-range:0.02:range;
x2=x1;
for i=1:length(x1)
        x_test = [x1(i)*ones(1,length(x2));x2];
        y_test=fx(x_test(1,:),x_test(2,:));
        xx=x_test';
        [yhat,D]=evaRBF(xx,Net2);
        C(i,:)=yhat*max_y;
end
mesh(x1,x2,C);
```



Sampling paired data for training

```
fstr=input('input a 2D function: ','s');
fx=inline(fstr);
range=2*pi;
x1=-range:0.02:range;
x2=x1;
for i=1:length(x1)
        C(i,:)=fx(x1(i),x2);
end
mesh(x1,x2,C);
hold on;
N=input('keyin sample size:');
x(1,:)=rand(1,N)*2*range-range;
x(2,:)=rand(1,N)*2*range-range;
n=rand(1,N)*0.1-0.05;
y=fx(x(1,:),x(2,:))+n;
plot3(x(2,:),x(1,:),y,'.');
```

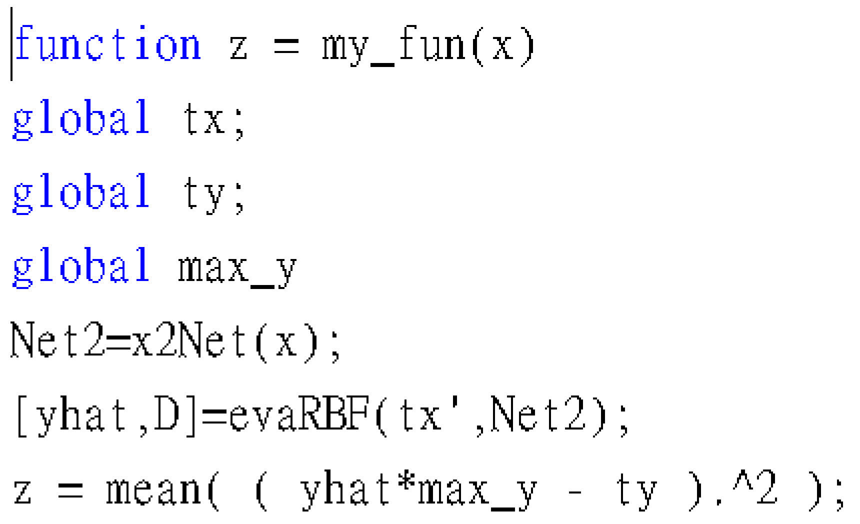
MSE(Mean Square Error)

- Objective function
- Mean approximating error

$$E_S(\theta) = \frac{1}{2N}\sum_{t=1}^{N}(y[t] - y(t\,|\,\theta))^2$$

Objective function

my_fun.m



```
function z = my_fun(x)
global tx;
global ty;
global max_y
Net2=x2Net(x);
[yhat,D]=evaRBF(tx',Net2);
z = mean( ( yhat*max_y - ty ).^2 );
```

GA optimization for RBF learning

```
global tx
global ty
global max_y
% run : sampling paired data
tx=x; ty=y; max_y=max(y);M=3;d=2;
L=M*d+2*M+1;
[theta2 fval] = ga(@my_fun, L);
fval
z = my_fun(theta2);
fprintf('ms training error derived by GA:%f\n',z);
Net2=x2Net(theta2);
% Plot RBF mapping
```

GA_RBF

demo_genetic_RBF.m