Levenberg-Marquardt learning

- MATLAB programming
- RBF(Radial Basis Function)

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

$$\theta = [\mathbf{u}_1^T\ \mathbf{u}_2^T \cdots \mathbf{u}_M^T\ \sigma_1\ \sigma_2 \cdots \sigma_{\mathrm{M}}\ \mathrm{w}_0\ \mathrm{w}_1\ \mathrm{w}_2 \cdots \mathrm{w}_{\mathrm{M}}]^T$$

$$= [\theta_1, \ldots, \theta_{M*d+2M+1}]$$

Data Structure

- Net
  - Net.u : receptive field Mxd
  - Net.si : variance Mx1
  - Net.w : posterior weights M+1
  - Net.M : number of hidden units
  - Net.d : data dimension

LM_iniNet_RBF.m

```
M = input('M : ');
d = 2;
ep = 0.001;
L = M*d+2*M+1;
theta = rand(1,L)*2*ep - ep;
Net.u=reshape(theta(1:M*d),[d,M])';
Net.si=theta(M*d+1:M*d+M);
Net.w=theta(M*d+M+1:M*d+M+M);
Net.M = M;
Net.d = d;
```

Input & output

- x: Nxd
  - N: data size
  - d: data dimension
- y: Nx1
- Net

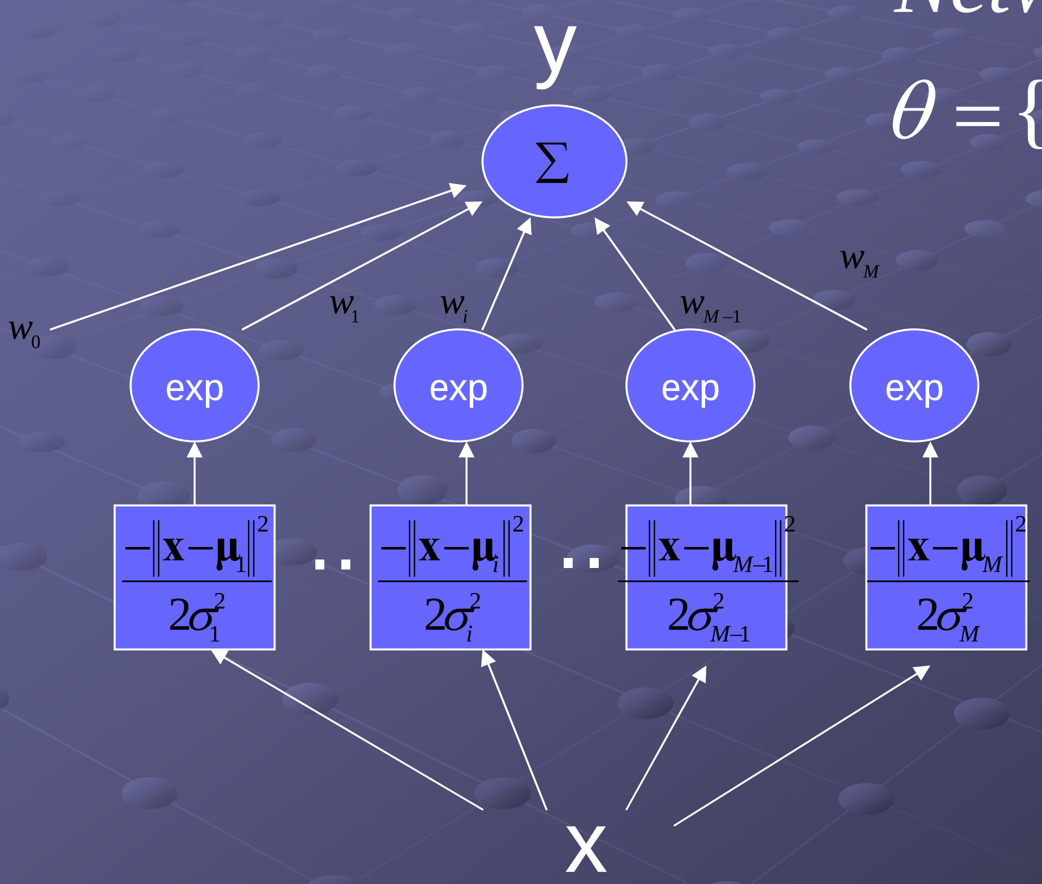
demo_LM_RBF.m

```
% x : Nxd
% y : Nx1
N=200;d=2;
x=rand(N,d)*2-1;
y=x(:,1)*0.5+x(:,2)+1;
Net=LM_RBF(x,y);
```

RBF

Network parameter

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



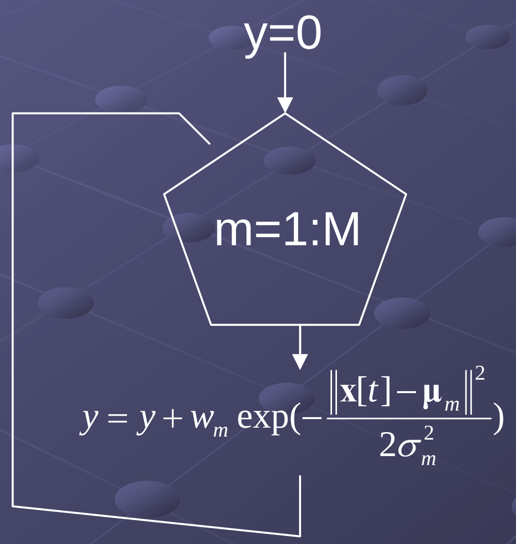
RBF Evaluation

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

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

Function 1

function y=eva_f(Net,x)



X: Nxd

u: Mxd

$$D_{ij} = (\mathbf{x}_i - \mathbf{u}_j)(\mathbf{x}_i^T - \mathbf{u}_j^T)$$

$$= \mathbf{x}_i \mathbf{x}_i^T - 2\mathbf{x}_i \mathbf{u}_j^T + \mathbf{u}_j \mathbf{u}_j^T$$

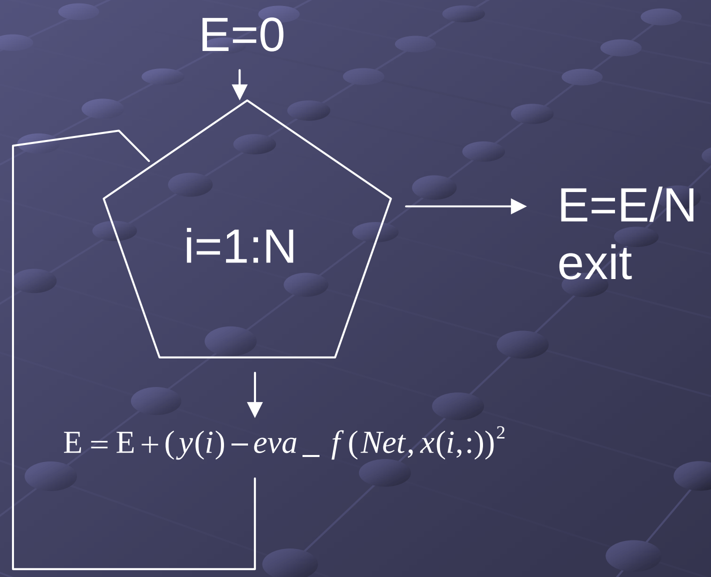
$$= A_{ii} + B_{ij} + C_{jj}$$

```
M=net.M; u=net.u; N=size(x,1);
D=x*u';
D=-2*D+repmat(sum(x.*x,2),1,M);
D=D+repmat(sum(u.*u,2)',N,1);
```

Current error

- Approximating error
  - e=y-yhat
  - Mean square error
    - mean(e.^2)

Function2
function E= DC(Net,x,y)



Derivative

derivative.m

```
function [gtheta]=derivative(Net,x,theta,h,v)
```

gtheta=[gu gsi gw]

% gtheta : N x (Md+2M+1)
% gu : N x Md
% gsi : N x M
% gw : N x (M+1)

gtheta : N x (Md+2M+1)

$$\varphi(t,\boldsymbol{\theta}) = \frac{dy(t|\boldsymbol{\theta})}{d\boldsymbol{\theta}}$$

$$\frac{dy(t\,|\,\boldsymbol{\theta})}{d\mathbf{u}_m}$$

gu : N x Md

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

$$y(t\,|\,\theta)$$

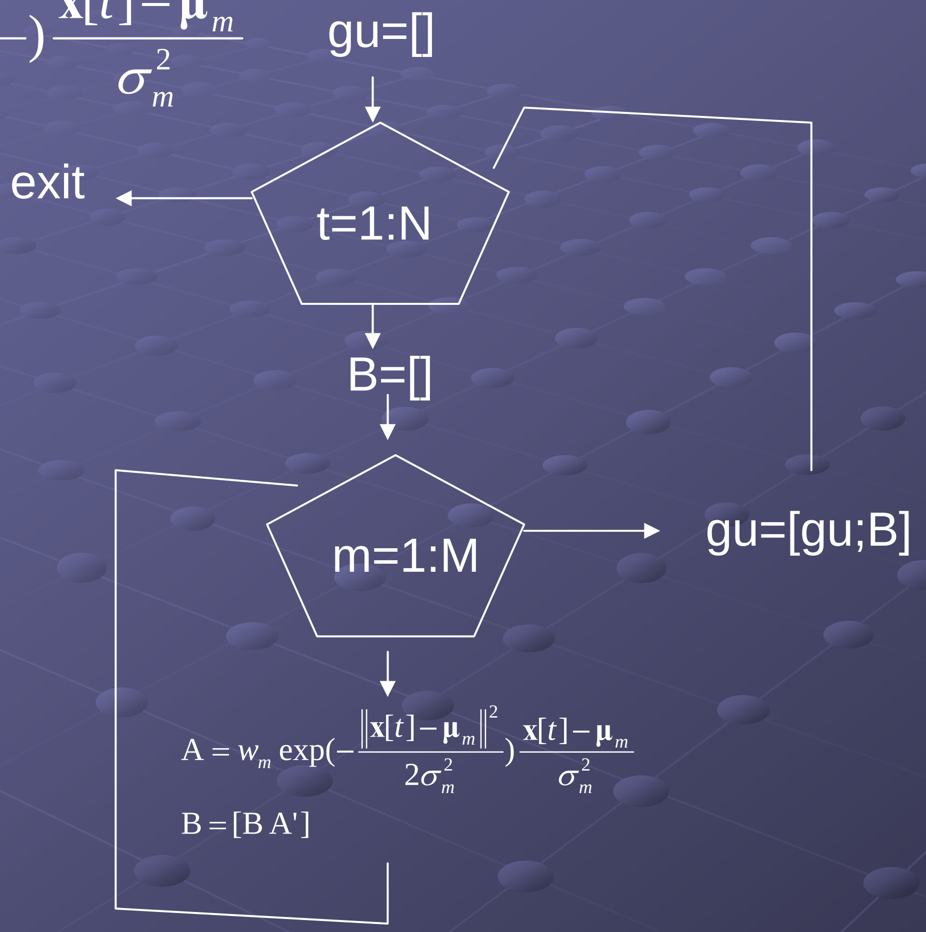
$$= g\mathbf{u}_{tm}$$

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

$$\frac{dy(t\,|\,\boldsymbol{\theta})}{d\mathbf{u}_m}$$

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

$$= g\mathbf{u}_{tm}$$



$$ga = \begin{pmatrix} ga_{11}^T & \cdots & ga_{1k}^T & \cdots \\ \vdots & & \vdots & \\ ga_{t1}^T & \cdots & ga_{tk}^T & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}_{\mathrm{N}\times Md}$$

t runs from 1 to N

k runs from 1 to M

$ga_{tk} : d\times 1$

$$g\mathbf{u}_{tm} = w_m \exp\left(-\frac{\left\|\mathbf{x}[t]-\boldsymbol{\mu}_m\right\|^2}{2\sigma_m^2}\right)\frac{\mathbf{x}[t]-\boldsymbol{\mu}_m}{\sigma_m^2}$$

gtheta : N x (Md+2M+1)

$$\varphi(t,\boldsymbol{\theta}) = \frac{dy(t|\boldsymbol{\theta})}{d\boldsymbol{\theta}}$$

gsi : N x M

$$\frac{dy(t\,|\,\boldsymbol{\theta})}{d\sigma_m}$$

$$= w_m \exp\left(-\frac{\left\|\mathbf{x}[t]-\boldsymbol{\mu}_m\right\|^2}{2\sigma_m^2}\right)\frac{\left\|\mathbf{x}[t]-\boldsymbol{\mu}_m\right\|^2}{\sigma_m^3}$$

gtheta : N x (Md+2M+1)

$$\varphi(t,\boldsymbol{\theta}) = \frac{dy(t|\boldsymbol{\theta})}{d\boldsymbol{\theta}}$$

gw : N x (M+1)

$$\frac{dy(t\,|\,\boldsymbol{\theta})}{dw_0} = 1$$

$$\frac{dy(t\,|\,\boldsymbol{\theta})}{dw_m} = \exp\left(-\frac{\left\|\mathbf{x}[t]-\boldsymbol{\mu}_m\right\|^2}{2\sigma_m^2}\right)$$

```
[gtheta] = derivative(Net,x,theta,h,v);
G = -(e*gtheta/Net.T)';
R = gtheta'*gtheta/Net.T;
```

$$\nabla(\boldsymbol{\theta}_i) = \frac{-1}{N}\sum_{t=1}^{N}\varepsilon(t,\boldsymbol{\theta}_i)\psi(t,\boldsymbol{\theta}_i)$$

$$R(\boldsymbol{\theta}_i) = \frac{1}{N}\sum_{t=1}^{N}\psi(t,\boldsymbol{\theta}_i)\psi^T(t,\boldsymbol{\theta}_i)$$

LM method

LM.m

1. Initialize $\boldsymbol{\theta}_i$, i=0 and set λ
2. Calculate $\nabla(\boldsymbol{\theta}_i)$ and $R(\boldsymbol{\theta}_i)$ and $\Delta\boldsymbol{\theta}_i$
3. Update network parameters

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \Delta\boldsymbol{\theta}_i$$

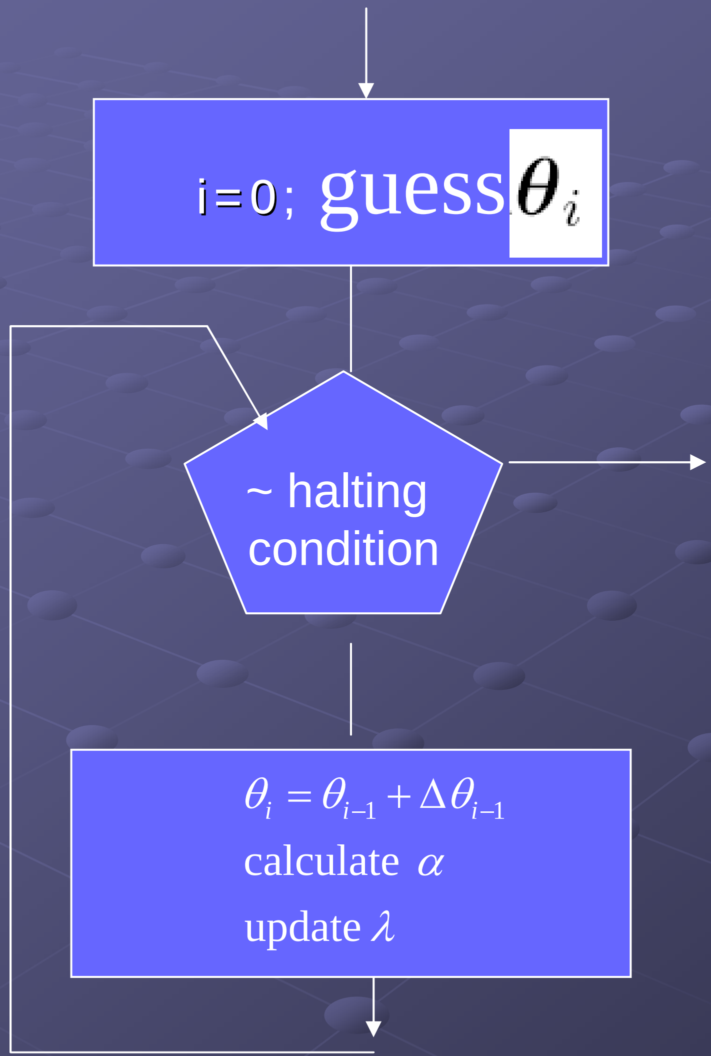
4. Calculate α_i
5. Update λ

(a) If $\alpha_i > 0.75$, $\lambda \leftarrow 0.5\lambda$.

(b) If $\alpha_i < 0.25$, $\lambda \leftarrow 2\lambda$.

6. If halting condition hold, exit otherwise go to step 2

Flow Chart



i=0; guess θ_i
~ halting condition
$\theta_i = \theta_{i-1} + \Delta\theta_{i-1}$
calculate α
update λ

- Calculate delta_theta
- Update theta

Control λ

Current parameter

Next parameter

$$\alpha_i = \frac{E_S(\boldsymbol{\theta}_i) - E_S(\boldsymbol{\theta}_i + \boldsymbol{\Delta\theta}_i)}{E_S(\boldsymbol{\theta}_i) - L_i(\boldsymbol{\theta}_i + \boldsymbol{\Delta\theta}_i)}$$

$$\frac{\text{Actual cost reduction}}{\text{Predicted cost reduction}}$$

Project

Learning RBF or MLP by the LM method

1. (30 pt) Derivation

$$\frac{dy(t \mid \theta)}{d\theta} = ?$$

2. (140 pt) Matlab codes
3. (50 pt) Testing

Derivative

$$\theta = [\mathbf{u}_1^T \ \mathbf{u}_2^T \cdots \mathbf{u}_M^T \ \sigma_1 \ \sigma_2 \cdots \sigma_\mathrm{M} \ \mathrm{w}_0 \ \mathrm{w}_1 \ \mathrm{w}_2 \cdots \mathrm{w}_\mathrm{M}]^T$$

$$= [\theta_1, \ldots, \theta_{M*d+2M+1}]$$

$$\frac{dy(t \mid \theta)}{d\theta} = [\frac{dy(t \mid \theta)}{d\theta_1}, \cdots, \frac{dy(t \mid \theta)}{d\theta_{M*d+2M+1}}]^T$$

Derivative

$$\frac{dy(t|\theta)}{d\mathbf{u}_m} = ?$$

$$\frac{dy(t|\theta)}{d\sigma_m} = ?$$

$$\frac{dy(t|\theta)}{dw_m} = ?$$

Matlab coding

- (30 pt)Main program
- Matlab Functions
  a. (10 pt)Calculate $E_S(\boldsymbol{\theta}_i)$

  b. (10 pt)Calculate $\mathrm{L}_i(\boldsymbol{\theta}_i)$

Matlab coding

- Matlab functions
  - (10 pt) Calculate $\frac{dy(t\mid\theta)}{d\mathbf{u}_m}\Big|_{\theta=\theta_i}$
  - (10 pt) Calculate $\frac{dy(t\mid\theta)}{d\sigma_m}\Big|_{\theta=\theta_i}$
  - (10 pt) Calculate $\frac{dy(t\mid\theta)}{dw_m}\Big|_{\theta=\theta_i}$

Matlab coding

- Matlab functions
  - (20 pt) Calculate $\nabla(\theta_i)$
  - (20 pt) Calculate $R(\theta_i)$
  - (20 pt) Calculate $G(\mathbf{x}\,|\,\theta)$

Test

- Give two examples to test your matlab codes for learning RBF or MLP networks by Levenberg-Marquardt method

Evaluation

- LM_iniNet 10%
  - Form a
  - Form b
  - Form r
- MLP evaluation 10%
- Calculation of mse 5%
- Determine gtheta 30%
  - Ga
  - Gb
  - Gr
- Update theta 15%
  - Delta_theta
  - Gradient
  - NG Hessian
- Calculate alpfa 10%
  - Change of Li
  - Change of E
- Update lamda 5%
- Halting condition 5%
- Executable 10%

Good performance 100%
2D function approximation 100%