Levenberg-Marquardt learning

- MATLAB programming
- MLP (Multilayer Perceptrons)

MLP (multilayer perceptrons)

- Weight sum of post-tanh projections

$$F(\mathbf{x};\theta) = \sum_{m=1}^{M} r_m \tanh(\mathbf{a}_m^T \mathbf{x} + b_m) + r_0$$

$$\theta = \{\mathbf{a}_m\} \cup \{b_m\} \cup \{r_m\}$$

Data Structure

- Net
  - Net.a : receptive field Mxd
  - Net.b : bias
  - Net.r : posterior weights
  - Net.M : number of hidden units
  - Net.d : data dimension

LM_iniNet.m

```
M = input('M : ');
d = 2;
ep = 0.001;
L = M*d+2*M+1;
theta = rand(1,L)*2*ep - ep;
% form a
Net.a=reshape(theta(1:M*d),[d,M])';
% form b,r ??
% Net.a = rand(M,d)*2*ep - ep;
Net.M = M;
Net.d = d;
% Net.b
% Net.r
```

Input & output

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

demo_LM.m

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

Data Structure

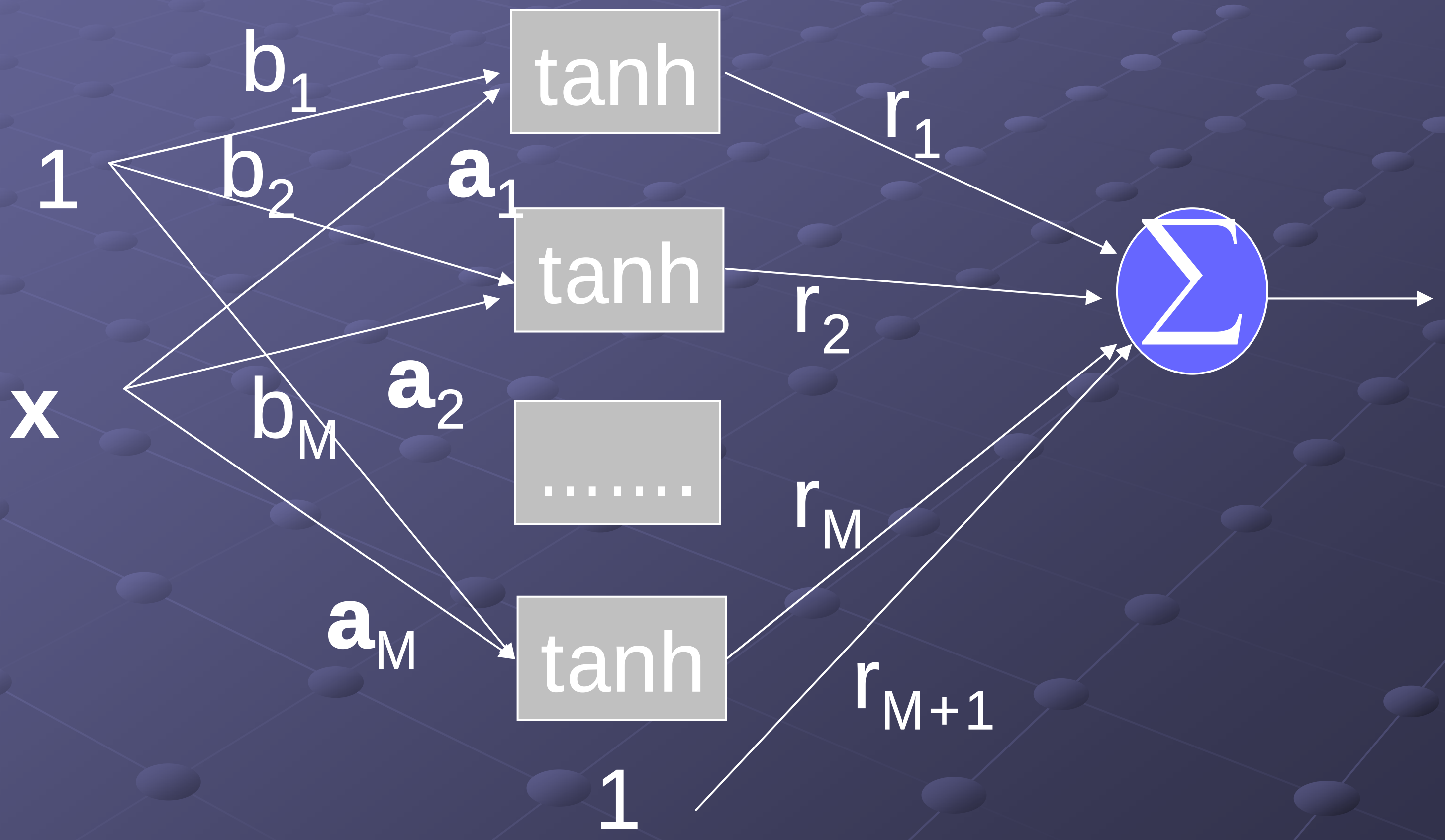
- Form a

```
a = reshape(theta(1:Net.M*Net.d),[Net.d,Net.M])';
```

- How to form b and r ?

$$\boldsymbol{\theta} = [\mathbf{a}_1^T\ \mathbf{a}_2^T \cdots \mathbf{a}_M^T\ b_1\ b_2 \cdots b_{\mathrm{M}}\ \mathrm{r}_0\ \mathrm{r}_1\ \mathrm{r}_2 \cdots \mathrm{r}_{\mathrm{M}}]^T$$
$$= [\theta_1, \ldots, \theta_{M*d+2M+1}]$$

MLP



MLP Evaluation

- Form h : NxM
  - $x*a^T$
- Add b
- Form v : NxM
  - Feed to tanh
- Multiply to r
- yhat : network output, Nx1

$$h_m = \mathbf{a}_m^T \mathbf{x} + b_m$$

$$v_m = \tanh(h_m)$$

Current error

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

Derivative

derivative.m

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

gtheta=[ga gb gr]

% gtheta : N x (Md+2M+1)
% ga : N x Md
% gb : N x M
% gr : N x (M+1)

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

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

ga : N x Md

$$\frac{dy(t\,|\,\boldsymbol{\theta})}{d\mathbf{a}_k}$$

$$= r_k(1 - \tanh(\mathbf{a}_k^T\mathbf{x}[t] + b_k)^2)\mathbf{x}[t]$$

$$= ga_{tk}$$

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

t runs from 1 to N

k runs from 1 to M

$ga_{tk} : d \times 1$ $$ga_{tk} = r_k (1 - \tanh(\mathbf{a}_k^T \mathbf{x}[t] + b_k)^2)\mathbf{x}[t]$$

```
[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

- 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

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

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

Derivative

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

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

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

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

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