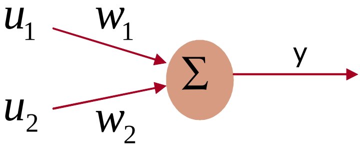
Training of NN - 2

Wen Yu

https://www.ctrl.cinvestav.mx/~yuw/
yuw@ctrl.cinvestav.mx



$$y = w_1 u_1 + w_2 u_2$$

The weights are trained as

$$w\left(k+1\right) = w\left(k\right) + \eta e\left(k\right) u_i$$
$$e\left(k\right) = y\left(k\right) - \hat{y}\left(k\right)$$

$$w_1\left(k+1\right) = w_1\left(k\right) + \eta\left[y\left(k\right) - \hat{y}\left(k\right)\right] u_1$$
$$w_2\left(k+1\right) = w_2\left(k\right) + \eta\left[y\left(k\right) - \hat{y}\left(k\right)\right] u_2$$

But

$$y = w_1 u_1 + w_2 u_2$$

$$y(1) = w_1 u_1(1) + w_2 u_2(1)$$
$$y(2) = w_1 u_1(2) + w_2 u_2(2)$$

Or

$$\begin{bmatrix} y(1) \\ y(2) \end{bmatrix} = \begin{bmatrix} u_1(1) & u_2(1) \\ u_1(2) & u_2(2) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

So

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} u_1(1) & u_2(1) \\ u_1(2) & u_2(2) \end{bmatrix}^{-1} \begin{bmatrix} y(1) \\ y(2) \end{bmatrix}$$

But if

$$y(1) = w_1 u_1(1) + w_2 u_2(1)$$
$$y(2) = w_1 u_1(2) + w_2 u_2(2)$$
$$y(3) = w_1 u_1(3) + w_2 u_2(3)$$

Least square method

Gauss (1777-1855). How to design a line

$$y = ax + b$$

by a group of data (x_i, y_i). For two indpendent data (x_1, y_1) and (x_2, y_2)

$$a = \frac{y_1 - y_2}{x_1 - x_2}, \quad b = \frac{-x_1 y_2 - x_2 y_1}{x_1 - x_2}$$

For n data (x_i, y_i), how to find a best line? We define

$$e_i = y_i - (ax_i + b)$$

The square error is

$$J = \sum_{i}^{n} e_i^2$$

Least squre is

$$\min_{(a,b)} J$$

Least square method

The solution for (a, b) is

$$\frac{\partial J}{\partial a} = 0, \quad \frac{\partial J}{\partial b} = 0$$

So

$$a = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2}$$
$$b = \frac{\sum x_i^2 \sum y_i^2 - \sum x_i \sum x_i y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2}$$

Least square method

In general case, we use linear regression

$$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_m x_m$$

where $(x_1 \cdots x_m, y)$ is known data, $(w_1 \cdots w_m)$ is parameter
For each data $(x_{i1} \cdots x_{mi}, y_i)$

$$\begin{aligned}
y_1 &= w_1 x_{11} + w_2 x_{12} + \cdots + w_m x_{1m} + e_1 \\
y_2 &= w_1 x_{21} + w_2 x_{22} + \cdots + w_m x_{2m} + e_2 \\
&\vdots \\
y_n &= w_1 x_{n1} + w_2 x_{n2} + \cdots + w_m x_{nm} + e_n
\end{aligned}$$

In matrix form

$$Y = XW + E$$

where $W = \left[w_1 \cdots w_m \right]^T$, $Y = \left[y_1 \cdots y_n \right]^T$,

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & & x_{nm} \end{bmatrix} \in R^{n \times m}$$

Least square method

Because

$$J = E^T E = \sum_i^n e_i^2, \quad \min_W J \rightarrow \text{least square}, \ \frac{\partial J}{\partial W} = 0$$

$$\begin{array}{l} \frac{\partial J}{\partial W} = \frac{\partial J}{\partial E}\frac{\partial E}{\partial W} = 2E\frac{\partial}{\partial W}\left(Y - XW\right) \\ = 2\left(Y - XW\right)^T\left(-X\right) \\ = -2Y^T X + 2W^T X^T X = 0 \end{array}$$

So

$$Y = XW + E$$

$$W = \left(X^T X\right)^{-1} X^T Y$$

Linear system

Discrete-time linear system

$$\begin{array}{l} y\left(k\right)=-a_{1}y\left(k-1\right)-\cdots-a_{n}y\left(k-n\right) \\ +b_{1}u\left(k-1\right)+\cdots+b_{m}u\left(k-m\right) \end{array}$$

It is ARX model

$$Ay\left(k\right)=Bu\left(k\right)$$

where $A=1+a_{1}q^{-1}+\cdots+a_{n}q^{-n}$, $B=b_{1}q^{-1}+\cdots+b_{m}q^{-m}$.
With color noise, it is ARMAX model

$$Ay\left(k\right)=Bu\left(k\right)+C\xi\left(k\right)$$

Parameter-in-linear

$$y = \theta_1 x_1 + \cdots + \theta_n x_n$$

where

$$x_i = \phi\left(u\right), \qquad \text{or } x_i = \phi\left(Vu\right)$$

In matrix form

$$Y = X\Theta + E$$

where $Y = \left[y_1 \cdots y_m\right]^T$, $X = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & & x_{mn} \end{bmatrix} \in R^{m\times n}$,
$\Theta = \left[\theta_1 \cdots \theta_n\right]^T$, $E = \left[e_1 \cdots e_m\right]^T$.

least square

$$Y = X\Theta + E, \qquad J = E^T E = \sum_i^m e_i^2$$

$\min\limits_{\Theta} J \longrightarrow$least square, $\frac{\partial J}{\partial \Theta} = 0$

$$\begin{array}{l} \frac{\partial J}{\partial \Theta} = \frac{\partial J}{\partial E}\frac{\partial E}{\partial \Theta} = 2E\frac{\partial}{\partial \Theta}\left(Y - X\Theta\right) \\ = 2\left(Y - X\Theta\right)^T\left(-X\right) \\ = -2Y^T X + 2\Theta^T X^T X \\ = 0 \end{array}$$

So

$$\Theta = \left(X^T X\right)^{-1} X^T Y$$

Linear model

In linear regressive form

$$y(k) = \varphi^T(k)\,\hat{\theta} + e(k)$$

where

$$\varphi(k) = \left[-y(k-1)\cdots - y(k-n),\, u(k-1)\cdots u(k-m)\right]^T$$
$$\hat{\theta} = \left[\hat{a}_1 \cdots \hat{a}_n, \hat{b}_1 \cdots \hat{b}_m\right]^T$$

For l data $y(k)$ and $\varphi(k)$
Least square

$$\hat{\theta} = \left[\sum_{k=1}^{l} \varphi^T(k)\,\varphi(k)\right]^{-1} \sum_{k=1}^{l} y(k)\,\varphi(k)$$

where $l >> \max(m,n)$. With this θ

$$\min J = \min \sum_{k=1}^{l} e^2(k)$$

Regressive least square

To find the relation

$$\hat{\theta} = \left[\sum_{k=1}^{l} \varphi^T(k)\,\varphi(k) \right]^{-1} \sum_{k=1}^{l} y(k)\,\varphi(k)$$

$$\theta_{k+1} = \theta_k + P_{k+1}\varphi(k+1)\left[y(k+1) - \varphi^T(k+1)\,\theta_k \right]$$
$$P_{k+1} = P_k - \frac{P_{kN}\varphi^T(k+1)\varphi(k+1)P_k}{1+\varphi(k+1)P_k\varphi^T(k+1)}, \qquad P_k >> 0$$

It is feedforward neural network with a single hidden layer, the output active function is linear,

$$\hat{y} = W\phi\left(Vx\right) \tag{1}$$

- the weights V in the hidden layer are random
- W will be trained by the least square method

The linear regression is (NN) $\phi_m\left(Vx_i\right)$

$$\hat{y} = w_1x_1 + w_2x_2 + \cdots + w_mx_m$$

The real data is

$$y_i = w_1\phi_m\left(Vx_i\right) + w_2\phi_m\left(Vx_i\right) + \cdots + w_m\phi_m\left(Vx_i\right) + e_i$$

where

$$e_i = y_i - \hat{y}$$

Extreme learning machines (ELM)

$$Y = ZW + E, \qquad Z = \phi\left(Vx_i\right)$$
$$y_i = W\phi\left(Vx_i\right) + e_i \tag{2}$$

If we want to

$$J = E^T E = \sum_i^n e_i^2, \quad \min_W J$$

then

$$W = \left(Z^T Z\right)^{-1} Z^T Y$$

where $Y = [y_1 \cdots y_n]$, $Z = \begin{bmatrix} \phi\left(Vx_{11}\right) & \cdots & \phi\left(Vx_{1m}\right) \\ \vdots & \ddots & \vdots \\ \phi\left(Vx_{n1}\right) & & \phi\left(Vx_{nm}\right) \end{bmatrix} \in R^{n\times m}$

ELM - Matlab code

```
for i=1:M "training time"
      for j=1:N
           I(j,i)=0; "input to each hidden node"
           for k=1:L
                 I(j,i)=I(j,i)+W(k,j)*II(k,i);
           end
           O(j,i)=(exp(I(j,i))-exp(-I(j,i)))/(exp(I(j,i))+exp(-I(j,i)));
      end
      Y(i)=y(i); "target output (teacher)"
end
V=Y*pinv(O); "weights in uotput layer, Moore-Penrose pseudoinverse
training"
```

Variations on learning rate

- Fixed rate much smaller than 1
- Start with large η, gradually decrease its value
- Start with a small η, steadily double it until MSE start to increase
- Find the maximum safe step size at each stage of learning (to avoid overshoot the minimum J when increasing η)
- Adaptive learning rate (Conjugate gradient minimisation)
  - Each weight w_{ij} has its own rate η_{ij}
  - If Δw_{ij} remains in the same direction, increase η_{ij}
  - If Δw_{ij} changes the direction, decrease η_{ij}

Variations on learning rate

- time-varying rate as $\eta\left(k\right)=\frac{c}{k}$ can give optimally fast convergence, but it results in slow convergence to bad solutions when c and k are small.
- Modification $\eta\left(k\right)=\frac{\eta_0}{1+a_1\frac{c}{k}+a_2\left(\frac{c}{k}\right)^2}$.
- Stable learning $\eta\left(k\right)=\frac{\eta_0}{1+\|\varphi(x_k)\|}$
- new learning rate as $\eta\left(k\right)=\frac{\eta_0}{1+a_1\frac{c}{k}+\|G\left(x_k\right)\|}$

Plant

$$y(k) = \varphi^T(k)\,\theta^*$$

Model

$$\hat{y}(k) = \varphi^T(k)\,\theta$$

$$\theta(k+1) = \theta(k) - \eta_k \varphi(k)\, e(k), \quad e(k) = \hat{y}(k) - y(k) \tag{3}$$

Gradient

Gradient

$$\eta_k = \eta$$

$$\theta(k+1) = \theta(k) - \eta\varphi(k)e(k)$$

Least Square

Least square $\eta_k \in R^{n \times n}$

$$\eta_k = \frac{P_k}{1 + \varphi P_k^T \varphi^T}$$

$$\begin{array}{l} \theta_{k+1} = \theta_k + \eta_k \varphi e_k \\ P_{k+1} = P_k - \frac{P_{kN} \varphi^T \varphi P_k}{1 + \varphi P_k \varphi^T}, \quad P_1 >> 0 \end{array}$$

Kalman

Kalman $\eta_k \in R^{n\times n}$ is matrix

$$\eta_k = \frac{P_k}{R_2 + \varphi P_k \varphi^T}$$

$$\begin{aligned} \theta_{k+1} &= \theta_k + \eta_k \varphi e_k \\ P_{k+1} &= P_k - \frac{P_k \varphi^T \varphi P_k}{R_2 + \varphi P_k \varphi^T} + R_1 \end{aligned}$$

where $R_1, R_2 > 0$

Stable learning

Nonlinear system

$$y(k) = f[X(k)]$$

The neural networks is

$$\widehat{y}(k) = W_k \phi[X(k)]$$

If the identified nonlinear system can be represented as (matching condition)

$$y(k) = W^* \phi[X(k)]$$

The training law is

$$W_{k+1} = W_k - \eta \phi[X(k)] e(k)$$

where

$$e(k) = \widehat{y}(k) - y(k)$$

Error dynamic

$$\widehat{y}(k) = W_k \phi[X(k)]$$
$$y(k) = W^* \phi[X(k)]$$

Modeling error is

$$e(k) = \widehat{y}(k) - y(k)$$
$$= [W_k - W^*] \phi[X(k)]$$
$$= \tilde{W}_k \phi[X(k)]$$

where

$$\tilde{W}_k = W_k - W^*$$

We select Lyapunov function as

$$V_k = \left\| \tilde{W}_k \right\|^2 = tr \left\{ \tilde{W}_k^T \tilde{W}_k \right\}$$

Becasue

$$\begin{array}{l} \tilde{W}_{k+1} = \tilde{W}_k - \eta_k \phi \left[X\left(k\right)\right] e\left(k\right) \\ e\left(k\right) = \tilde{W}_k \phi \left[X\left(k\right)\right] \end{array}$$

From the updating law

$$\begin{array}{l} \Delta V_k = V_{k+1} - V_k = \left\| \tilde{W}_k - \eta_k \phi \left[X\left(k\right)\right] e\left(k\right) \right\|^2 - \left\| \tilde{W}_k \right\|^2 \\ = \left\| \tilde{W}_k \right\|^2 - 2\eta_k e\left(k\right) \phi \left[X\left(k\right)\right] \tilde{W}_k + \left\| \eta_k \phi \left[X\left(k\right)\right] e\left(k\right) \right\|^2 - \left\| \tilde{W}_k \right\|^2 \\ = \eta_k^2 e^2\left(k\right) \left\| \phi \left[X\left(k\right)\right] \right\|^2 - 2\eta_k e\left(k\right) \phi \left[X\left(k\right)\right] \tilde{W}_k \\ = \eta_k^2 e^2\left(k\right) \left\| \phi \left[X\left(k\right)\right] \right\|^2 - 2\eta_k e^2\left(k\right) \\ = \left[\eta_k \left\| \phi \left[X\left(k\right)\right] \right\|^2 - 2 \right] \eta_k e^2\left(k\right) \end{array}$$

Lyapunov

We select

$$\eta_k = \frac{2\eta_0}{1 + \|\phi[X(k)]\|^2}, \qquad 0 \le \eta_0 \le 1$$

then

$$\left[\eta_k \|\phi[X(k)]\|^2 - 2\right] = 2\left[\frac{\|\phi[X(k)]\|^2}{1 + \|\phi[X(k)]\|^2}\eta_0 - 1\right] \le 2[\eta_0 - 1] \le 0$$

so

$$\Delta V_k \le 0$$