ARTIFICIAL NEURAL NETWORK

Swetha R.
Department of Electronics and Communication Engineering

ARTIFICIAL NEURAL NETWORK

Class 6:

LINEAR LEAST SQUARES FILTER

Swetha R.

Department of Electronics and Communication Engineering

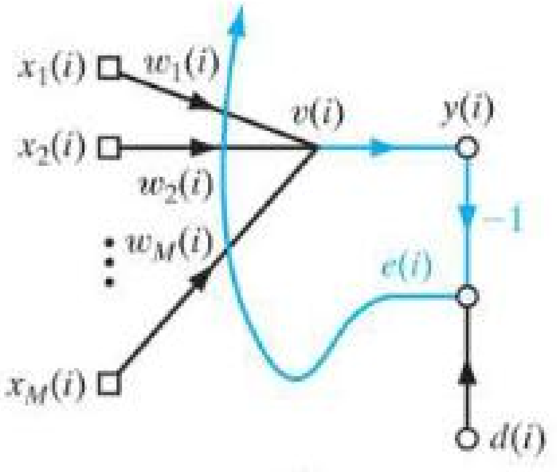
OUTLINE

- Linear Least Squares Filter
- Wiener Filter

Swetha R.
Department of Electronics and Communication Engineering

LINEAR LEAST-SQUARES FILTER

- Deterministic method
- Built with the help of single linear neuron
- The cost function is the sum of squared errors produced by the filter over a finite set of (training) data



LINEAR LEAST-SQUARES FILTER

Error vector,

e(n) = d(n) – $[X(1), X(2), \ldots, X(n)]^T$ w(n)

= d(n) – X^T(n)w(n) -------------------- (1)

Where, d(n) = $[d(1), d(2), \ldots, d(n)]^T$ n x 1 desired response vector

X(n) = $[X(1), X(2), \ldots, X(n)]$ m x n data matrix

w(n) – m x 1 weight matrix

LINEAR LEAST-SQUARES FILTER

Case 1: assuming error e = 0 in equation (1)

0 = d(n) – $[\mathrm{W}(1), \mathrm{W}(2), \ldots, \mathrm{W}(\mathrm{n})]^T X$(n)

=> $W^T X = d(n)$-------------------- (2)

Take Transpose of the above equation, We get

$X^T W = d^T(n)$

Multiply with X on both sides

=> X(n)$d^T(n)$ =$\mathrm{X}(n)\mathrm{X}^T$(n)w(n)

$\Rightarrow$ w(n) = $(\mathrm{X}(n)\mathrm{X}^T(\mathrm{n}))^{-1}$ X(n)$d^T(n)$

$\Rightarrow$ For inverse to exist, | $\mathrm{X}(n)\mathrm{X}^T(\mathrm{n})$| is non zero and hence rank of this matrix must be full

PES UNIVERSITY ONLINE

LINEAR LEAST-SQUARES FILTER

Case 2: However, error is not equal to zero always, e $\neq 0$, In such cases, we have to define the cost function as squared error

$$E = \frac{1}{2}\sum_{i=1}^{n} e^2(i)$$

In matrix form,

$$E = \frac{1}{2}e(n)e^T(n)$$

Where, e(n) = [e(1) e(2) … e(n)]

LINEAR LEAST-SQUARES FILTER

In matrix form,

$$E = \frac{1}{2}e(n)e^T(n) = \frac{1}{2}(\mathrm{d(n)} - \mathrm{X}^T(\mathrm{n})\mathrm{w(n)})(\mathrm{d(n)} - \mathrm{X}^T(\mathrm{n})\mathrm{w(n)})^T$$

$$= \frac{1}{2}(\mathrm{d} - \mathrm{X}^T\mathrm{w})(\mathrm{d} - \mathrm{X}^T\mathrm{w})^T \text{(neglecting time steps)}$$

$$= \frac{1}{2}(\mathrm{dd}^T - 2\mathrm{dX}^T\mathrm{w} + \mathrm{w}^T X\mathrm{X}^T\mathrm{w})$$

To find an optimal weight, diff. the cost function and equate that to zero

$$\nabla E = -\mathrm{Xd}^T + X\mathrm{X}^T\mathrm{w} = 0$$

=> W = $(\mathrm{XX}^T)^{-1}\,\mathrm{Xd}^T$ => Linear least squares solution

Derivative Identity:

Let a, b $\in \mathbb{R}^n$, then

$$\frac{\partial(a^T b)}{\partial a} = \frac{\partial(b^T a)}{\partial a} \triangleq \begin{pmatrix} \frac{\partial(b^T a)}{\partial a_1} \\ \cdots \\ \frac{\partial(b^T a)}{\partial a_n} \end{pmatrix} = \begin{pmatrix} b_1 \\ \cdots \\ b_n \end{pmatrix} = b$$

And Let P = $P^T \in \mathbb{R}^{nxn}$

$$\frac{\partial(a^T P a)}{\partial a} = 2Pa$$

If XX^T is singular, i.e., rank of XX^T is equal to m, then it is customary practice to add diagonal matrix δI , which results in

$$\boxed{W = (XX^T + \delta I)^{-1}\, Xd^T} \text{--------------------}(3)$$

Where δ is a positive constant and I is the identity matrix

Eq(3) is the solution to the cost function

$$E = \frac{1}{2} e(n)e^T(n) + \frac{\delta}{2}\|w^T w\|$$

LINEAR LEAST-SQUARES FILTER

WKT, $e(n) = d(n) - X^T(n)w(n)$

$$\nabla e(n) = -X(n)$$

But Jacobian, $J(n) = \nabla e(n)^T = -X(n)^T$

$$\boxed{W = -(J^T J)^{-1} J^T d}$$

Limitations:

When data size is large, computing inverse matrix would be difficult. In such cases, iterative approach will be better to use.

LINEAR LEAST-SQUARES FILTER

WKT, $e(n) = d(n) - X^T(n)w(n)$

$$\nabla e(n) = -X(n)$$

But Jacobian, $J(n) = \nabla e(n)^T = -X(n)^T$

Wkt, by Gauss Newton method

$$w(n+1) = w(n) - \left(J^T(n)J(n)\right)^{-1} J^T(n)e(n)$$

$$= w(n) - (\left(X(n)X^T(n)\right)^{-1} X(n))(d(n) - X^T(n)w(n))$$

$$= \left(X(n)X^T(n)\right)^{-1} X(n)d(n)$$

LINEAR LEAST-SQUARES FILTER

$$w(n+1) = \left(X(n)X^T(n)\right)^{-1} X(n)\mathrm{d(n)}$$

Let $X^+ = \left(X(n)X^T(n)\right)^{-1} X(n) = pseudo$ inverse.

$$w(n+1) = X^+(n)d(n)$$

WIENER FILTER

Department of Electronics and Communication Engineering

WIENER FILTER

- Limiting form of the Linear Least-Squares Filter for an Ergodic Environment
- Consider the input vector x(i) and desired response d(i) are drawn from an ergodic environment that is also stationary.

Second order statistics of such process:

- Correlation matrix of the input vector x(i); it is denoted by R_x,
- Cross-correlation vector between the input vector x(i) and desired response d(i);it is denoted by r_{xd}.

$$R_x = E[x(i))x^T(i)] = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} x(i))x^T(i) = \lim_{n\to\infty}\frac{1}{n}(X^T(n)X(n)$$

$$r_{xd} = E\big[x(i)\big)d\ (i)] = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} x(i))d(i) = \lim_{n\to\infty}\frac{1}{n}(X^T(n)d(n)$$

WIENER FILTER

Cost Function ,

$$\xi \triangleq E[e^2(k)]$$

where, E - Expectation

In matrix form,

$$e(k) = D - W^T X$$

$$e^2(k) = (D - W^T X)(D - W^T X)^T$$

$$= DD^T - DX^T W - W^T X D^T + W^T X X^T W$$

WIENER FILTER

$$\Rightarrow\ \xi = E[e^2(k)] = E[DD^T - DX^TW - W^TXD^T + \ W^TXX^TW\ ]$$

$$= r_d - r_{dx}W - W^Tr_{xd} + W^TR_xW$$

Where, $r_d = E[DD^T]$, $R_x = E[XX^T]$

$$r_{dx} = E[DX^T] = r_{xd} = E[XD^T]$$

WIENER FILTER

$$\Rightarrow \frac{\partial \xi}{\partial W} = - r_{dx} - r_{xd} + W^T R_x + R_x W$$

$$\Rightarrow - 2r_{dx} + 2R_x W = 0$$

$$\boxed{W = R_x^{-1} r_{dx}}$$

=>wiener solution to optimum filtering problem

WIENER FILTER

- For an ergodic process, the linear least−squares filter asymptotically approaches the Wiener filter as the number of observations approaches infinity.
- Proof:

$$w_0 = \lim_{n \to \infty} w(n+1)$$

$$= \lim_{n \to \infty} (X^T(n)X(n))^{-1} X^T(n)d(n)$$

$$= \lim_{n \to \infty} \frac{1}{n}(X^T(n)X(n))^{-1} \lim_{n \to \infty} \frac{1}{n} X^T(n)d(n)$$

$$= R_x^{-1} r_{dx}$$

WIENER FILTER

- Designing of Wiener filter requires knowledge of the second-order statistics: the correlation matrix of the input vector x(n) and the cross-correlation vector between x(n) and the desired response d(n).
- However, this information is not available in many important situations encountered in practice.
- So, we go for linear adaptive filter.
- The widely used algorithm is the least-mean-square algorithm(LMS)

THANK YOU

Swetha R.
Department of Electronics and Communication Engineering

swethar@pes.edu

+91 80 2672 1983 Extn 753