شبکه‌های عصبی مصنوعی

درس ۱۰

یادگیری ویدرو–هاف

Widrow-Hoff Learning

کاظم فولادی قلعه

دانشکده مهندسی، پردیس فارابی

دانشگاه تهران

http://courses.fouladi.ir/nn

Widrow-Hoff Learning

(LMS Algorithm)

یادگیری ویدرو-هاف

WIDROW-HOFF LEARNING

الگوریتم یادگیری ویدرو-هاف،
یک الگوریتم کاهش تندترین شیب تقریبی است
که در آن، شاخص کارآیی، خطای میانگین مربعات (LMS) است.

اهمیت

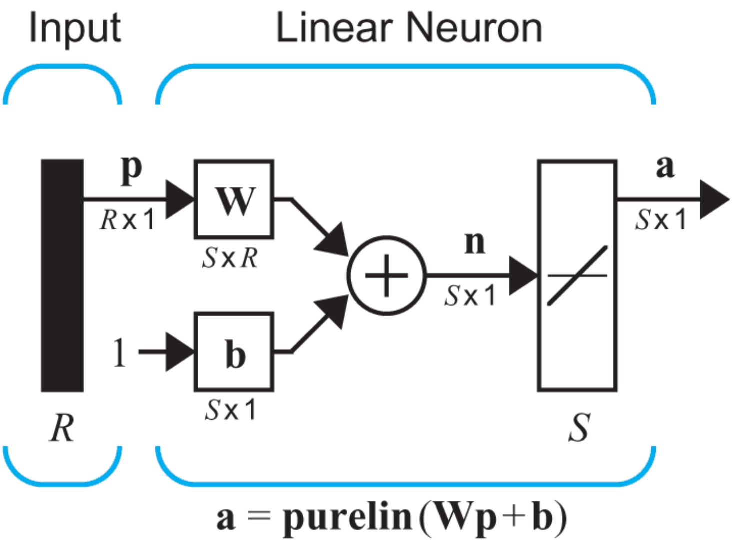
- کاربردهای متعدد آن در پردازش سیگنال (مانند حذف نویز و پژواک)
- پیش‌نمای الگوریتم پس‌انتشار خطا

یادگیری ویدرو–هاف

۱

شبکه‌ی آدالاین

ADALINE Network



$$\mathbf{a} = \mathbf{purelin}(\mathbf{W}\mathbf{p} + \mathbf{b}) = \mathbf{W}\mathbf{p} + \mathbf{b}$$

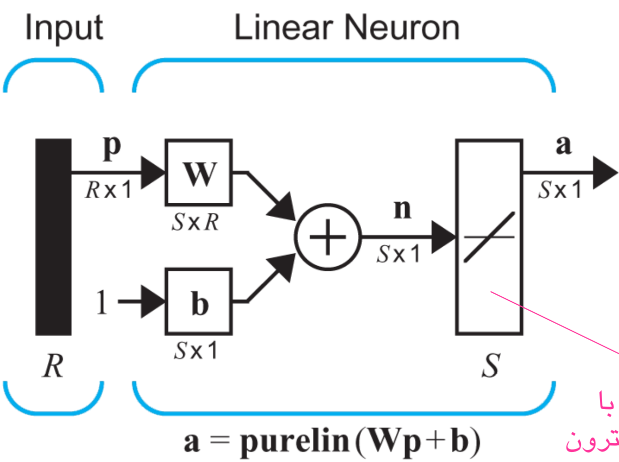
$$a_i = purelin(n_i) = purelin({}_i\mathbf{w}^T\mathbf{p} + b_i) = {}_i\mathbf{w}^T\mathbf{p} + b_i$$

$${}_i\mathbf{w} = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}$$

شبکه‌ی آدالاین

ADALINE NETWORK

شبکه‌ی آدالاین بسیار شبیه پرسپترون است، با این تفاوت که تابع انتقال آن، خطی است.



ADALINE: ADAptive LInear NEuron (NEtwork) +
LMS: Least Mean Square (learning algorithm)

$$\mathbf{a} = \mathbf{purelin}(\mathbf{W}\mathbf{p} + \mathbf{b}) = \mathbf{W}\mathbf{p} + \mathbf{b}$$

تنها تفاوت با
معماری پرسپترون

$$a_i = purelin(n_i) = purelin({}_i\mathbf{w}^T\mathbf{p} + b_i) = {}_i\mathbf{w}^T\mathbf{p} + b_i$$

$${}_i\mathbf{w} = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}$$

شبکه‌ی آدالاین

الگوریتم LMS

ADALINE Network

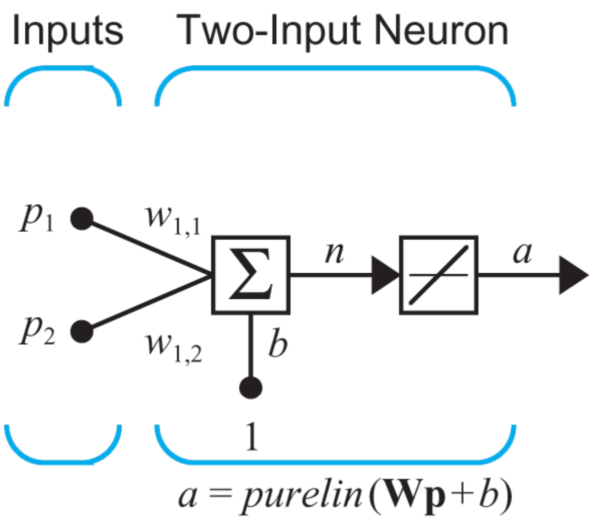
هر دو شبکه‌ی آدالاین و پرسپترون محدودیت ذاتی یکسانی دارند:
تنها می‌توانند مسائل جداپذیر خطی را حل کنند.

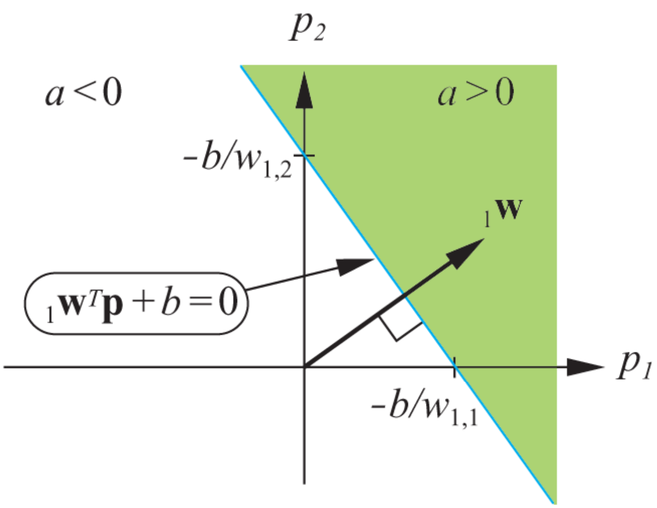
اما الگوریتم LMS قوی‌تر از قاعده‌ی یادگیری پرسپترون است:

- قاعده‌ی یادگیری پرسپترون تضمین همگرایی به راه‌حل برای الگوهای آموزشی دارد، اما شبکه‌ی حاصل می‌تواند به نویز حساس باشد (زیرا الگوها اغلب نزدیک مرز تصمیم قرار می‌گیرند).
- الگوریتم LMS خطای میانگین مربعات را می‌نیمم می‌کند و بنابراین مرزهای تصمیم‌گیری را تا جای ممکن از الگوهای آموزشی دور می‌کند.

الگوریتم LMS کاربردهای عملی بیشتری نسبت به قاعده‌ی یادگیری پرسپترون پیدا کرده است.
(به‌ویژه در پردازش سیگنال دیجیتال DSP)

Two-Input ADALINE



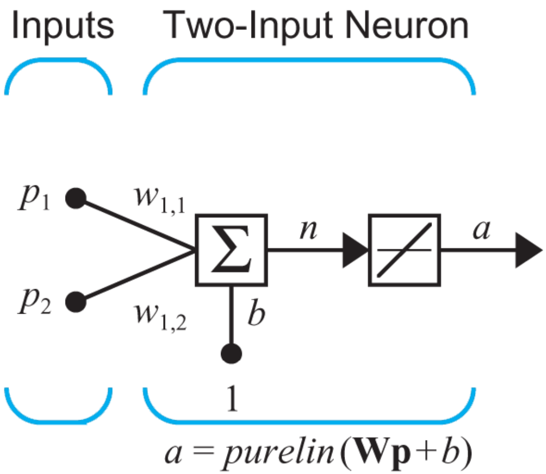


$$a = purelin(n) = purelin({}_1\mathbf{w}^T\mathbf{p} + b) = {}_1\mathbf{w}^T\mathbf{p} + b$$

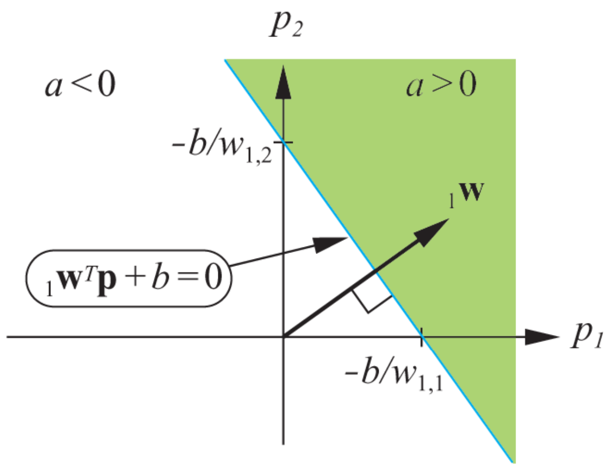
$$a = {}_1\mathbf{w}^T\mathbf{p} + b = w_{1,1}p_1 + w_{1,2}p_2 + b$$

آدالاین دو–ورودی

TWO-INPUT ADALINE



تفکیک‌گر خطی درست مانند پرسپترون



$$a = purelin(n) = purelin({}_1\mathbf{w}^T\mathbf{p}+b) = {}_1\mathbf{w}^T\mathbf{p}+b$$

$$a = {}_1\mathbf{w}^T\mathbf{p}+b = w_{1,1}p_1 + w_{1,2}p_2 + b$$

مرز تصمیم با $n = 0$ به‌دست می‌آید که یک خط است: یک طرف خروجی مثبت و یک‌طرف خروجی منفی

یادگیری ویدرو–هاف

۲

خطای میانگین مربعات

Mean Square Error

Training Set:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

Input: $\mathbf{p}_q$ Target: $\mathbf{t}_q$

Notation:

$$\mathbf{x} = \begin{bmatrix} {}_1\mathbf{w} \\ b \end{bmatrix} \qquad \mathbf{z} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \qquad a = {}_1\mathbf{w}^T\mathbf{p} + b \Rightarrow a = \mathbf{x}^T\mathbf{z}$$

Mean Square Error:

$$F(\mathbf{x}) = E[e^2] = E[(t-a)^2] = E[(t - \mathbf{x}^T\mathbf{z})^2]$$

خطای میانگین مربعات

MEAN SQUARE ERROR

Training Set:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \dots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

Input: $\mathbf{p}_q$ Target: $\mathbf{t}_q$

Notation:

$$\mathbf{x} = \begin{bmatrix} {}_1\mathbf{w} \\ b \end{bmatrix} \qquad \mathbf{z} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \qquad a = {}_1\mathbf{w}^T\mathbf{p} + b \Rightarrow a = \mathbf{x}^T\mathbf{z}$$

Mean Square Error:

$$F(\mathbf{x}) = E[e^2] = E[(t-a)^2] = E[(t-\mathbf{x}^T\mathbf{z})^2]$$

امید ریاضی: متوسط زمانی برای سیگنال‌های قطعی

Error Analysis

$$F(\mathbf{x}) = E[e^2] = E[(t-a)^2] = E[(t-\mathbf{x}^T\mathbf{z})^2]$$

$$F(\mathbf{x}) = E[t^2 - 2t\mathbf{x}^T\mathbf{z} + \mathbf{x}^T\mathbf{z}\mathbf{z}^T\mathbf{x}]$$

$$F(\mathbf{x}) = E[t^2] - 2\mathbf{x}^T\mathrm{E}[t\mathbf{z}] + \mathbf{x}^T E[\mathbf{z}\mathbf{z}^T]\mathbf{x}$$

$$\boxed{F(\mathbf{x}) = c - 2\mathbf{x}^T\mathbf{h} + \mathbf{x}^T\mathbf{R}\mathbf{x}}$$

$$c = E[t^2] \qquad \mathbf{h} = E[t\mathbf{z}] \qquad \mathbf{R} = E[\mathbf{z}\mathbf{z}^T]$$

The mean square error for the ADALINE Network is a quadratic function:

$$F(\mathbf{x}) = c + \mathbf{d}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}$$

$$\mathbf{d} = -2\mathbf{h} \qquad \mathbf{A} = 2\mathbf{R}$$

تحلیل خطا

Error Analysis

$$F(\mathbf{x}) = E[e^2] = E[(t-a)^2] = E[(t-\mathbf{x}^T\mathbf{z})^2]$$

$$F(\mathbf{x}) = E[t^2 - 2t\mathbf{x}^T\mathbf{z} + \mathbf{x}^T\mathbf{z}\mathbf{z}^T\mathbf{x}]$$

$$F(\mathbf{x}) = E[t^2] - 2\mathbf{x}^T \mathrm{E}[t\mathbf{z}] + \mathbf{x}^T E[\mathbf{z}\mathbf{z}^T]\mathbf{x}$$

$$\boxed{F(\mathbf{x}) = c - 2\mathbf{x}^T\mathbf{h} + \mathbf{x}^T\mathbf{R}\mathbf{x}}$$

$$c = E[t^2] \qquad \mathbf{h} = E[t\mathbf{z}] \qquad \mathbf{R} = E[\mathbf{z}\mathbf{z}^T]$$

همبستگی متقابل میان بردار ورودی و تارگت مربوطه

ماتریس همبستگی ورودی‌ها

The mean square error for the ADALINE Network is a quadratic function:

$$F(\mathbf{x}) = c + \mathbf{d}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}$$

$$\mathbf{d} = -2\mathbf{h} \qquad \mathbf{A} = 2\mathbf{R}$$

Stationary Point

Hessian Matrix:

$$\mathbf{A} = 2\mathbf{R}$$

The correlation matrix **R** must be at least positive semidefinite. If there are any zero eigenvalues, the performance index will either have a weak minumum or else no stationary point, otherwise there will be a unique global minimum **x***.

$$\nabla F(\mathbf{x}) = \nabla\left(c + \mathbf{d}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}\right) = \mathbf{d} + \mathbf{A}\mathbf{x} = -2\mathbf{h} + 2\mathbf{R}\mathbf{x}$$

$$-2\mathbf{h} + 2\mathbf{R}\mathbf{x} = \mathbf{0}$$

If R is positive definite:

$$\mathbf{x}^* = \mathbf{R}^{-1}\mathbf{h}$$

نقطه‌ی ایستان

STATIONARY POINT

Hessian Matrix:

$$\mathbf{A} = 2\mathbf{R}$$

The correlation matrix **R** must be at least positive semidefinite. If there are any zero eigenvalues, the performance index will either have a weak minumum or else no stationary point, otherwise there will be a unique global minimum **x***.

$$\nabla F(\mathbf{x}) = \nabla\left(c + \mathbf{d}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}\right) = \mathbf{d} + \mathbf{A}\mathbf{x} = -2\mathbf{h} + 2\mathbf{R}\mathbf{x}$$

$$-2\mathbf{h} + 2\mathbf{R}\mathbf{x} = \mathbf{0}$$

If R is positive definite:

$$\mathbf{x}^* = \mathbf{R}^{-1}\mathbf{h}$$

اگر $\mathbf{A} > 0$ باشد، این نقطه می‌نیمم سراسری است.

* اگر نخواهیم $\mathbf{R}^{-1}$ را محاسبه کنیم، می‌توانیم از الگوریتم کاهش تندترین شیب استفاده کنیم.

Approximate Steepest Descent

Approximate mean square error (one sample):

$$\hat{F}(\mathbf{x}) = (t(k) - a(k))^2 = e^2(k)$$

Approximate (stochastic) gradient:

$$\hat{\nabla} F(\mathbf{x}) = \nabla e^2(k)$$

$$[\nabla e^2(k)]_j = \frac{\partial e^2(k)}{\partial w_{1,j}} = 2e(k)\frac{\partial e(k)}{\partial w_{1,j}} \qquad j = 1, 2, \ldots, R$$

$$[\nabla e^2(k)]_{R+1} = \frac{\partial e^2(k)}{\partial b} = 2e(k)\frac{\partial e(k)}{\partial b}$$

کاهش تندترین شیب تقریبی

Approximate Steepest Descent

خطای میانگین مربعات را با خطا روی یک نمونه تقریب می‌زنیم. (E برداشته می‌شود.)

Approximate mean square error (one sample):

$$\hat{F}(\mathbf{x}) = (t(k) - a(k))^2 = e^2(k)$$

Approximate (stochastic) gradient:

گرادیان تقریبی (تصادفی):

$$\hat{\nabla} F(\mathbf{x}) = \nabla e^2(k)$$

مربوط به وزن‌ها :

$$[\nabla e^2(k)]_j = \frac{\partial e^2(k)}{\partial w_{1,j}} = 2e(k)\frac{\partial e(k)}{\partial w_{1,j}} \qquad j = 1, 2, \dots, R$$

مربوط به بایاس :

$$[\nabla e^2(k)]_{R+1} = \frac{\partial e^2(k)}{\partial b} = 2e(k)\frac{\partial e(k)}{\partial b}$$

* یادگیری **برخط** (online) / **افزایشی** (incremental) ← با استفاده از گرادیان تقریبی

Approximate Gradient Calculation

$$\frac{\partial e(k)}{\partial w_{1,j}} = \frac{\partial[t(k) - a(k)]}{\partial w_{1,j}} = \frac{\partial}{\partial w_{1,j}}[t(k) - ({}_1\mathbf{w}^T\mathbf{p}(k) + b)]$$

$$\frac{\partial e(k)}{\partial w_{1,j}} = \frac{\partial}{\partial w_{1,j}}\left[t(k) - \left(\sum_{i=1}^{R} w_{1,i}p_i(k) + b\right)\right]$$

$$\frac{\partial e(k)}{\partial w_{1,j}} = -p_j(k) \qquad \frac{\partial e(k)}{\partial b} = -1$$

$$\hat{\nabla} F(\mathbf{x}) = \nabla e^2(k) = -2e(k)\mathbf{z}(k)$$

محاسبه‌ی گرادیان تقریبی

Approximate Gradient Calculation

محاسبه‌ی مشتقات جزئی:

$$\frac{\partial e(k)}{\partial w_{1,j}} = \frac{\partial[t(k) - a(k)]}{\partial w_{1,j}} = \frac{\partial}{\partial w_{1,j}}[t(k) - ({}_1\mathbf{w}^T \mathbf{p}(k) + b)]$$

$$\frac{\partial e(k)}{\partial w_{1,j}} = \frac{\partial}{\partial w_{1,j}}\left[t(k) - \left(\sum_{i=1}^{R} w_{1,i} p_i(k) + b\right)\right]$$

$$\frac{\partial e(k)}{\partial w_{1,j}} = -p_j(k) \qquad \frac{\partial e(k)}{\partial b} = -1$$

فقط در یک جمله غیر صفر می‌شود.

$$\mathbf{z}(k) = \begin{bmatrix} p_1(k) \\ p_2(k) \\ \vdots \\ p_R(k) \\ 1 \end{bmatrix}$$

$$\hat{\nabla} F(\mathbf{x}) = \nabla e^2(k) = -2e(k)\mathbf{z}(k)$$

تنها لازم است خطا را در ورودی ضرب کنیم (و ضرب در -2)

یادگیری ویدرو–هاف

۳

الگوریتم LMS

LMS Algorithm

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla F(\mathbf{x}) \Big|_{\mathbf{x} = \mathbf{x}_k}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + 2\alpha e(k) \mathbf{z}(k)$$

$${}_1\mathbf{w}(k+1) = {}_1\mathbf{w}(k) + 2\alpha e(k) \mathbf{p}(k)$$

$$b(k+1) = b(k) + 2\alpha e(k)$$

الگوریتم LMS

LMS Algorithm

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla F(\mathbf{x})\Big|_{\mathbf{x} = \mathbf{x}_k}$$

$$\Downarrow$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + 2\alpha e(k)\mathbf{z}(k)$$

$$\Downarrow$$

$$\left\{\begin{aligned} {}_1\mathbf{w}(k+1) &= {}_1\mathbf{w}(k) + 2\alpha e(k)\mathbf{p}(k) \\ b(k+1) &= b(k) + 2\alpha e(k) \end{aligned}\right.$$

الگوریتم LMS

قاعده‌ی ویدرو–هاف

قاعده‌ی دلتا

Multiple-Neuron Case

$$ {}_i\mathbf{w}(k+1) = {}_i\mathbf{w}(k) + 2\alpha e_i(k)\mathbf{p}(k) $$

$$ b_i(k+1) = b_i(k) + 2\alpha e_i(k) $$

Matrix Form:

$$ \mathbf{W}(k+1) = \mathbf{W}(k) + 2\alpha\mathbf{e}(k)\mathbf{p}^T(k) $$

$$ \mathbf{b}(k+1) = \mathbf{b}(k) + 2\alpha\mathbf{e}(k) $$

الگوریتم LMS

در حالت چند نرونی (چند خروجی)

MULTIPLE-NEURON CASE

سطر iام ماتریس **W**

$$ {}_i\mathbf{w}(k+1) = {}_i\mathbf{w}(k) + 2\alpha e_i(k)\mathbf{p}(k) $$

$$ b_i(k+1) = b_i(k) + 2\alpha e_i(k) $$

Matrix Form:

$$ \mathbf{W}(k+1) = \mathbf{W}(k) + 2\alpha\mathbf{e}(k)\mathbf{p}^T(k) $$

$$ \mathbf{b}(k+1) = \mathbf{b}(k) + 2\alpha\mathbf{e}(k) $$

شبکه‌های عصبی مصنوعی

یادگیری ویدرو–هاف

۴

تحلیل همگرایی

تحلیل همگرایی الگوریتم LMS

ANALYSIS OF CONVERGENCE

$$\mathbf{x}_{k+1} = \mathbf{x}_k + 2\alpha e(k)\mathbf{z}(k)$$

$\mathbf{x}_k$ فقط تابعی است از

$$\mathbf{z}(k-1), \mathbf{z}(k-2), \ldots, \mathbf{z}(0)$$

با فرض مستقل آماری بودن بردارهای ورودی متوالی،
$\mathbf{x}_k$ مستقل از $\mathbf{z}(k)$ می‌شود.

اثبات می‌کنیم که
برای فرآیندهای ورودی ایستان که شرایط فوق را برآورده کند،
مقدار مورد انتظار (امید ریاضی) بردار وزن همگرا می‌شود به:

$$\mathbf{x}^* = \mathbf{R}^{-1}\mathbf{h}$$

Analysis of Convergence

$$\mathbf{x}_{k+1} = \mathbf{x}_k + 2\alpha e(k)\mathbf{z}(k)$$

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha E[e(k)\mathbf{z}(k)]$$

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha\{E[t(k)\mathbf{z}(k)] - E[(\mathbf{x}_k^T\mathbf{z}(k))\mathbf{z}(k)]\}$$

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha\{E[t_k\mathbf{z}(k)] - E[(\mathbf{z}(k)\mathbf{z}^T(k))\mathbf{x}_k]\}$$

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha\{\mathbf{h} - \mathbf{R}E[\mathbf{x}_k]\}$$

$$E[\mathbf{x}_{k+1}] = \underbrace{[\mathbf{I} - 2\alpha\mathbf{R}]}_{\text{For stability, the eigenvalues of this matrix must fall inside the unit circle.}}E[\mathbf{x}_k] + 2\alpha\mathbf{h}$$

تحلیل همگرایی الگوریتم LMS

ANALYSIS OF CONVERGENCE

از LMS داریم:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + 2\alpha e(k)\mathbf{z}(k)$$

از طرفین امید ریاضی می‌گیریم:

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha E[e(k)\mathbf{z}(k)]$$

$$\boxed{e(k) = t(k) - \mathbf{x}_k^T\mathbf{z}(k)}$$

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha\{E[t(k)\mathbf{z}(k)] - E[(\mathbf{x}_k^T\mathbf{z}(k))\mathbf{z}(k)]\}$$

$\mathbf{z}^{T}(k)\mathbf{x}_k$ را به‌جای $\mathbf{x}_k^{T}\mathbf{z}(k)$ جایگزین می‌کنیم:

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha\{E[t_k\mathbf{z}(k)] - E[(\mathbf{z}(k)\mathbf{z}^T(k))\mathbf{x}_k]\}$$

چون $\mathbf{x}_k$ مستقل از $\mathbf{z}(k)$ است، داریم:

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha\{\mathbf{h} - \mathbf{R}E[\mathbf{x}_k]\}$$

$$E[\mathbf{x}_{k+1}] = \underbrace{[\mathbf{I} - 2\alpha\mathbf{R}]}_{}E[\mathbf{x}_k] + 2\alpha\mathbf{h}$$

For stability, the eigenvalues of this matrix must fall inside the unit circle.

شرط پایداری سیستم دینامیکی خطی مرتبه اول:
مقادیر ویژه باید داخل دایره‌ی واحد باشند.

Conditions for Stability

$$|eig([\mathbf{I} - 2\alpha\mathbf{R}])| = |1 - 2\alpha\lambda_i| < 1$$

(where λ_i is an eigenvalue of **R**)

Since $\lambda_i > 0$, $1 - 2\alpha\lambda_i < 1$.

Therefore the stability condition simplifies to

$$1 - 2\alpha\lambda_i > -1$$

$$\alpha < 1/\lambda_i \quad \text{for all } i$$

$$\boxed{0 < \alpha < 1/\lambda_{max}}$$

تحلیل همگرایی الگوریتم LMS

شرایط پایداری

CONDITIONS FOR STABILITY

$$\left| eig([\mathbf{I} - 2\alpha\mathbf{R}]) \right| = \left| 1 - 2\alpha\lambda_i \right| < 1$$

(where λ_i is an eigenvalue of $\mathbf{R}$)

Since $\lambda_i > 0$, $1 - 2\alpha\lambda_i < 1$.

Therefore the stability condition simplifies to

$$1 - 2\alpha\lambda_i > -1$$

$$\alpha < 1/\lambda_i \quad \text{for all } i$$

$$\boxed{0 < \alpha < 1/\lambda_{max}}$$

10

Steady State Response

$$E[\mathbf{x}_{k+1}] = [\mathbf{I} - 2\alpha\mathbf{R}]E[\mathbf{x}_k] + 2\alpha\mathbf{h}$$

If the system is stable, then a steady state condition will be reached.

$$E[\mathbf{x}_{ss}] = [\mathbf{I} - 2\alpha\mathbf{R}]E[\mathbf{x}_{ss}] + 2\alpha\mathbf{h}$$

The solution to this equation is

$$E[\mathbf{x}_{ss}] = \mathbf{R}^{-1}\mathbf{h} = \mathbf{x}^*$$

This is also the strong minimum of the performance index.

پاسخ حالت ماندگار

Steady State Response

$$E[\mathbf{x}_{k+1}] = [\mathbf{I} - 2\alpha\mathbf{R}]E[\mathbf{x}_k] + 2\alpha\mathbf{h}$$

If the system is stable, then a steady state condition will be reached.

در حالت ماندگار (دائمی):

$$E[\mathbf{x}_{ss}] = [\mathbf{I} - 2\alpha\mathbf{R}]E[\mathbf{x}_{ss}] + 2\alpha\mathbf{h}$$

The solution to this equation is

$$E[\mathbf{x}_{ss}] = \mathbf{R}^{-1}\mathbf{h} = \mathbf{x}^*$$

This is also the strong minimum of the performance index.

نقطه‌ی می‌نیمم سراسری شاخص کارآیی

Example

$$\text{Banana} \quad \left\{ \mathbf{p}_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}, \mathbf{t}_1 = \begin{bmatrix} -1 \end{bmatrix} \right\} \qquad \text{Apple} \quad \left\{ \mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \mathbf{t}_2 = \begin{bmatrix} 1 \end{bmatrix} \right\}$$

$$\mathbf{R} = E[\mathbf{p}\mathbf{p}^T] = \frac{1}{2}\mathbf{p}_1\mathbf{p}_1^T + \frac{1}{2}\mathbf{p}_2\mathbf{p}_2^T$$

$$\mathbf{R} = \frac{1}{2}\begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}\begin{bmatrix} -1 & 1 & -1 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}\begin{bmatrix} 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$$\lambda_1 = 1.0, \qquad \lambda_2 = 0.0, \qquad \lambda_3 = 2.0$$

$$\alpha < \frac{1}{\lambda_{max}} = \frac{1}{2.0} = 0.5$$

الگوریتم LMS

مثال (۱ از ۴)

EXAMPLE

$$\text{Banana}\quad \left\{\mathbf{p}_1 = \begin{bmatrix}-1\\1\\-1\end{bmatrix}, \mathbf{t}_1 = \begin{bmatrix}-1\end{bmatrix}\right\} \qquad \text{Apple}\quad \left\{\mathbf{p}_2 = \begin{bmatrix}1\\1\\-1\end{bmatrix}, \mathbf{t}_2 = \begin{bmatrix}1\end{bmatrix}\right\}$$

$$\mathbf{R} = E[\mathbf{p}\mathbf{p}^T] = \frac{1}{2}\mathbf{p}_1\mathbf{p}_1^T + \frac{1}{2}\mathbf{p}_2\mathbf{p}_2^T$$

$$\mathbf{R} = \frac{1}{2}\begin{bmatrix}-1\\1\\-1\end{bmatrix}\begin{bmatrix}-1 & 1 & -1\end{bmatrix} + \frac{1}{2}\begin{bmatrix}1\\1\\-1\end{bmatrix}\begin{bmatrix}1 & 1 & -1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\0 & 1 & -1\\0 & -1 & 1\end{bmatrix}$$

$$\lambda_1 = 1.0, \qquad \lambda_2 = 0.0, \qquad \lambda_3 = 2.0$$

ابتدا محاسبه‌ی نرخ یادگیری پایدار α

$$\alpha < \frac{1}{\lambda_{max}} = \frac{1}{2.0} = 0.5$$

در کاربردهای واقعی ممکن است نتوانیم **R** را محاسبه کنیم و α باید با سعی و خطا انتخاب شود.

Iteration One

Banana $$a(0) = \mathbf{W}(0)\mathbf{p}(0) = \mathbf{W}(0)\mathbf{p}_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = 0$$

$$e(0) = t(0) - a(0) = t_1 - a(0) = -1 - 0 = -1$$

$$\mathbf{W}(1) = \mathbf{W}(0) + 2\alpha e(0)\mathbf{p}^T(0)$$

$$\mathbf{W}(1) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} + 2(0.2)(-1) \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}^T = \begin{bmatrix} 0.4 & -0.4 & 0.4 \end{bmatrix}$$

الگوریتم LMS

مثال (۲ از ۴)

EXAMPLE

Banana $$a(0) = \mathbf{W}(0)\mathbf{p}(0) = \mathbf{W}(0)\mathbf{p}_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = 0$$

$$e(0) = t(0) - a(0) = t_1 - a(0) = -1 - 0 = -1$$

$$\mathbf{W}(1) = \mathbf{W}(0) + 2\alpha e(0)\mathbf{p}^T(0)$$

$$\mathbf{W}(1) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} + 2(0.2)(-1)\begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}^T = \begin{bmatrix} 0.4 & -0.4 & 0.4 \end{bmatrix}$$

Iteration Two

Apple $a(1) = \mathbf{W}(1)\mathbf{p}(1) = \mathbf{W}(1)\mathbf{p}_2 = \begin{bmatrix} 0.4 & -0.4 & 0.4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = -0.4$

$$e(1) = t(1) - a(1) = t_2 - a(1) = 1 - (-0.4) = 1.4$$

$$\mathbf{W}(2) = \begin{bmatrix} 0.4 & -0.4 & 0.4 \end{bmatrix} + 2(0.2)(1.4) \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}^T = \begin{bmatrix} 0.96 & 0.16 & -0.16 \end{bmatrix}$$

الگوریتم LMS

مثال (۳ از ۴)

EXAMPLE

Apple $$a(1) = \mathbf{W}(1)\mathbf{p}(1) = \mathbf{W}(1)\mathbf{p}_2 = \begin{bmatrix} 0.4 & -0.4 & 0.4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = -0.4$$

$$e(1) = t(1) - a(1) = t_2 - a(1) = 1 - (-0.4) = 1.4$$

$$\mathbf{W}(2) = \begin{bmatrix} 0.4 & -0.4 & 0.4 \end{bmatrix} + 2(0.2)(1.4) \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}^T = \begin{bmatrix} 0.96 & 0.16 & -0.16 \end{bmatrix}$$

Iteration Three

$$a(2) = \mathbf{W}(2)\mathbf{p}(2) = \mathbf{W}(2)\mathbf{p}_1 = \begin{bmatrix} 0.96 & 0.16 & -0.16 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = -0.64$$

$$e(2) = t(2) - a(2) = t_1 - a(2) = -1 - (-0.64) = -0.36$$

$$\mathbf{W}(3) = \mathbf{W}(2) + 2\alpha e(2)\mathbf{p}^T(2) = \begin{bmatrix} 1.1040 & 0.0160 & -0.0160 \end{bmatrix}$$

$$\mathbf{W}(\infty) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

الگوریتم LMS

مثال (۴ از ۴)

EXAMPLE

$$a(2) = \mathbf{W}(2)\mathbf{p}(2) = \mathbf{W}(2)\mathbf{p}_1 = \begin{bmatrix} 0.96 & 0.16 & -0.16 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = -0.64$$

$$e(2) = t(2) - a(2) = t_1 - a(2) = -1 - (-0.64) = -0.36$$

$$\mathbf{W}(3) = \mathbf{W}(2) + 2\alpha e(2)\mathbf{p}^T(2) = \begin{bmatrix} 1.1040 & 0.0160 & -0.0160 \end{bmatrix}$$

$$\mathbf{W}(\infty) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

مرز تصمیم پیدا شده با روش LMS بهتر از پرسپترون است (فاصله‌ی بیشتر مرز از الگوها).

یادگیری ویدرو–هاف

۵

فیلتر کردن وفقی

فیلتر کردن وفقی

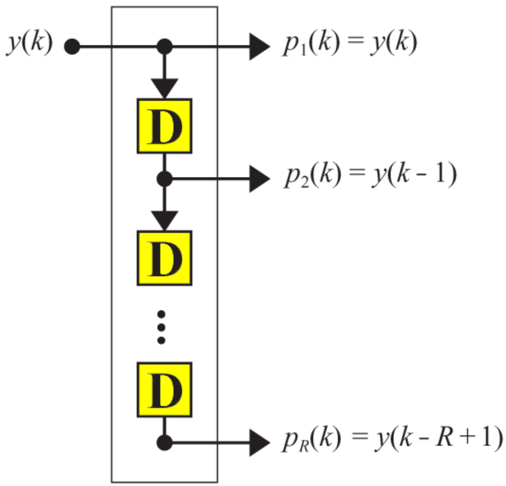
ADAPTIVE FILTERING

یکی از کاربردهای گسترده‌ی آدالاین: **فیلترهای وفقی**

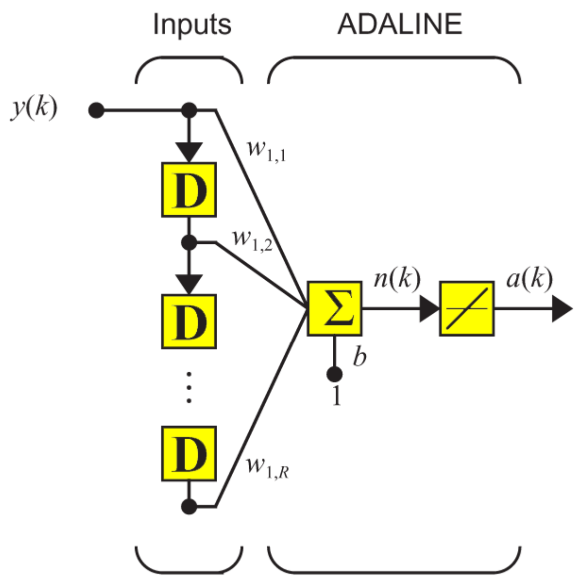
ترکیب خط شنود تأخیری با شبکه‌ی آدالاین، یک فیلتر وفقی ایجاد می‌کند.

Adaptive Filtering

Tapped Delay Line



Adaptive Filter



$a(k) = purelin(\mathbf{W}\mathbf{p}(k) + b)$

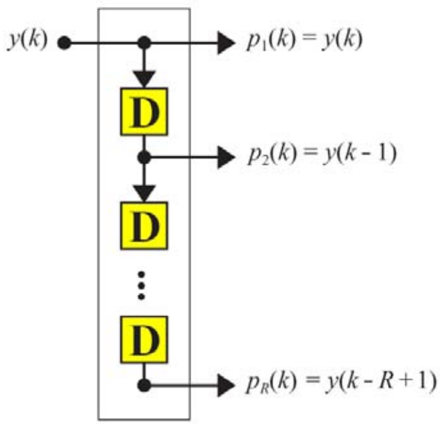
$$a(k) = purelin(\mathbf{W}\mathbf{p} + b) = \sum_{i=1}^{R} w_{1,i}\, y(k - i + 1) + b$$

فیلتر کردن وفقی

Adaptive Filtering

خط شنود تأخیری

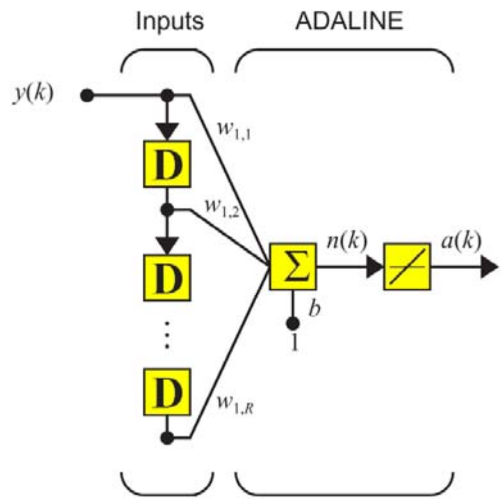
Tapped Delay Line



خط شنود تأخیری با R خروجی
(ورودی جاری و $R-1$ ورودی قبلی)

فیلتر وفقی

Adaptive Filter



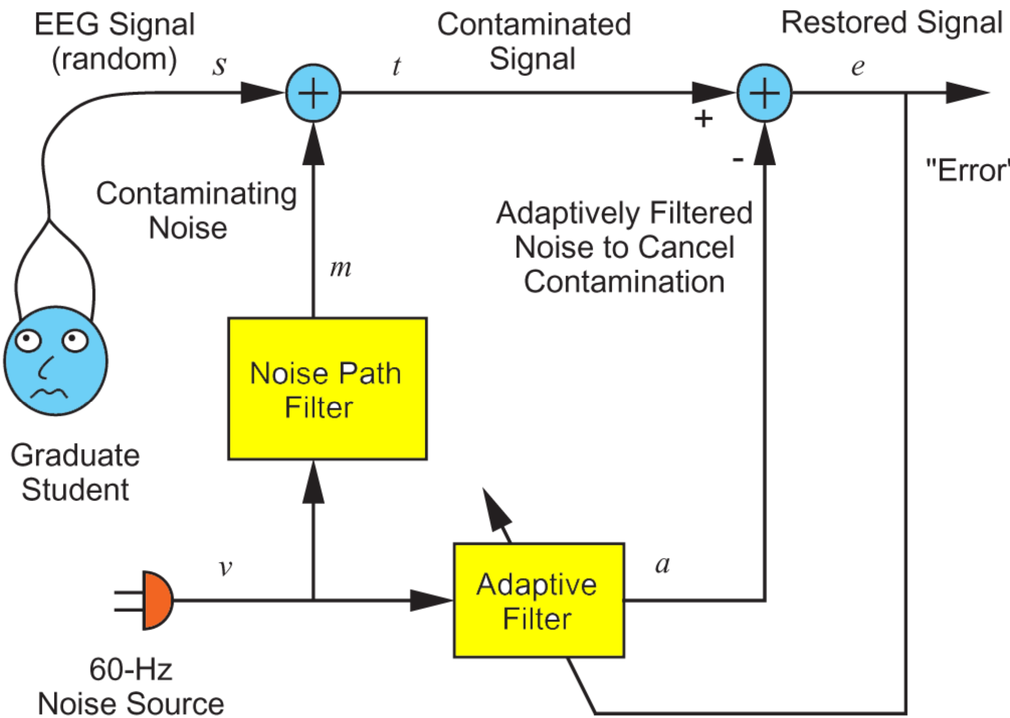
$$a(k) = purelin(\mathbf{W}\mathbf{p}(k) + b)$$

$$a(k) = purelin(\mathbf{W}\mathbf{p} + b) = \sum_{i=1}^{R} w_{1,i}\, y(k-i+1) + b$$

این یک فیلتر FIR است (پاسخ ضربه‌ی متناهی: Finite Impulse Response).

10

Example: Noise Cancellation

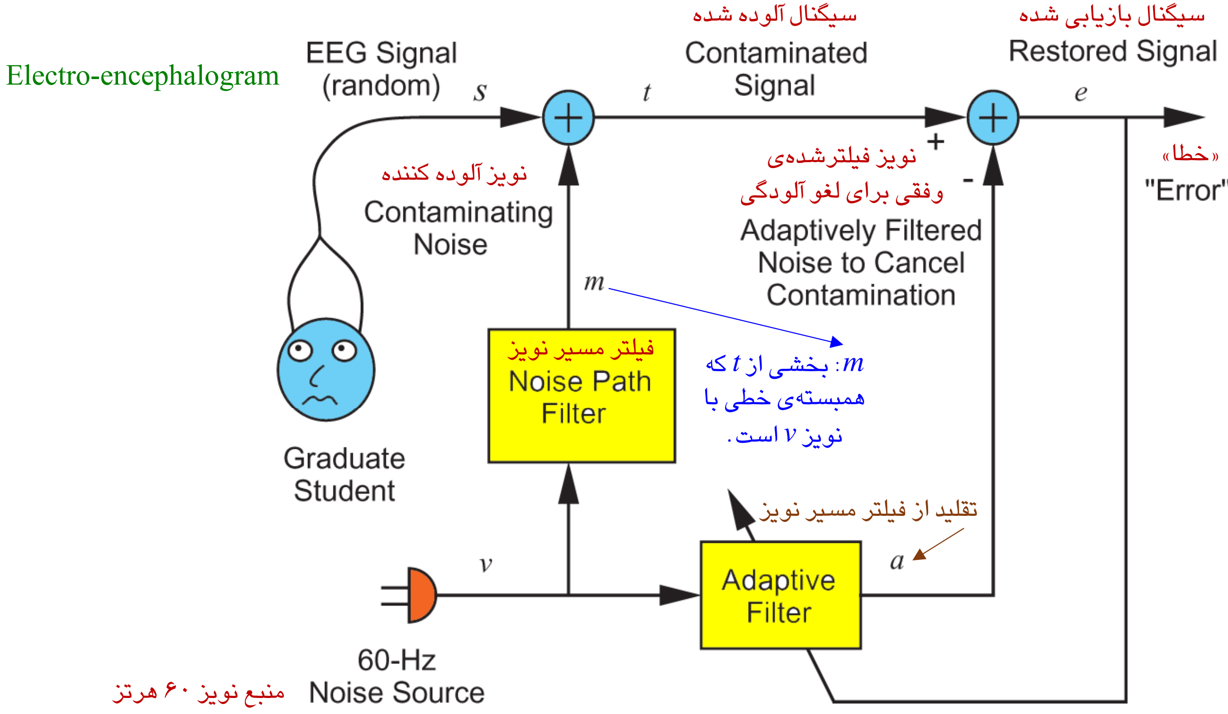


Adaptive Filter Adjusts to Minimize Error (and in doing this removes 60-Hz noise from contaminated signal)

فیلتر کردن وفقی

مثال: لغو نویز وفقی

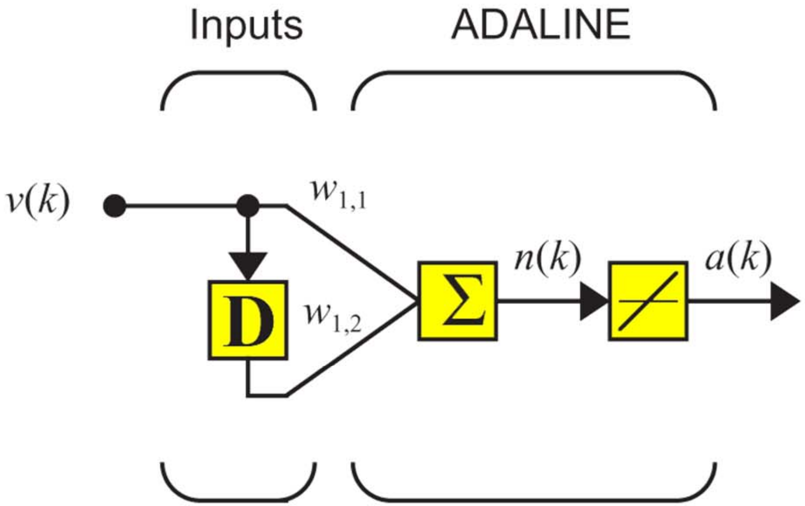
EXAMPLE: NOISE CANCELLATION



Adaptive Filter Adjusts to Minimize Error (and in doing this removes 60-Hz noise from contaminated signal)

فیلتر وفقی تنظیم کننده برای می‌نیمم‌سازی خطا
(که انجام آن نویز ۶۰ هرتز را از سیگنال آلوده شده حذف می‌کند.)

Noise Cancellation Adaptive Filter

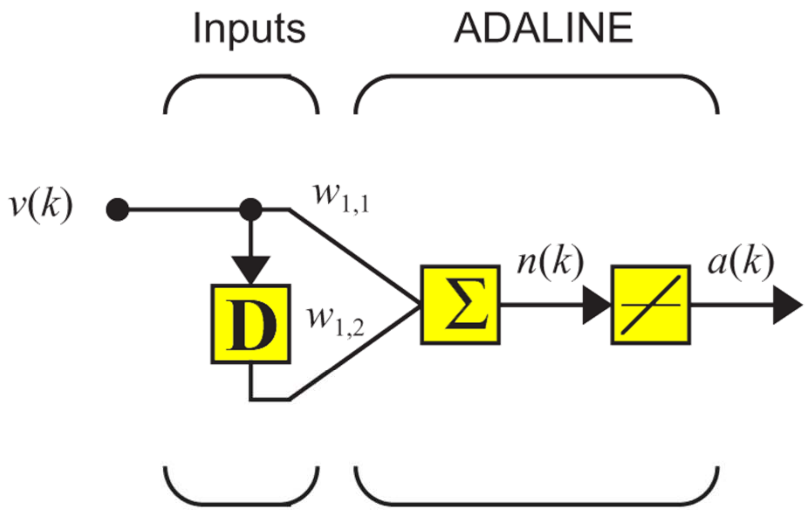


$$a(k) = w_{1,1}\, v(k) + w_{1,2}\, v(k-1)$$

فیلتر کردن وفقی

فیلتر وفقی رفع نویز

NOISE CANCELLATION ADAPTIVE FILTER



$$a(k) = w_{1,1}\, v(k) + w_{1,2}\, v(k-1)$$

Correlation Matrix

$$\mathbf{R} = [\mathbf{z}\mathbf{z}^T] \qquad\qquad \mathbf{h} = E[t\mathbf{z}]$$

$$\mathbf{z}(k) = \begin{bmatrix} v(k) \\ v(k-1) \end{bmatrix}$$

$$t(k) = s(k) + m(k)$$

$$\mathbf{R} = \begin{bmatrix} E[v^2(k)] & E[v(k)v(k-1)] \\ E[v(k-1)v(k)] & E[v^2(k-1)] \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} E[(s(k)+m(k))v(k)] \\ E[(s(k)+m(k))v(k-1)] \end{bmatrix}$$

فیلتر کردن وفقی

ماتریس همبستگی

CORRELATION MATRIX

$$\mathbf{R} = [\mathbf{z}\mathbf{z}^T] \qquad \mathbf{h} = E[t\mathbf{z}]$$

$$\mathbf{z}(k) = \begin{bmatrix} v(k) \\ v(k-1) \end{bmatrix}$$

$$t(k) = s(k) + m(k)$$

$$\mathbf{R} = \begin{bmatrix} E[v^2(k)] & E[v(k)v(k-1)] \\ E[v(k-1)v(k)] & E[v^2(k-1)] \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} E[(s(k)+m(k))v(k)] \\ E[(s(k)+m(k))v(k-1)] \end{bmatrix}$$

Signals

$$v(k) = 1.2\ \sin\left(\frac{2\pi k}{3}\right) \qquad m(k) = 1.2\ \sin\left(\frac{2\pi k}{3} - \frac{3\pi}{4}\right)$$

$$E[v^2(k)] = (1.2)^2 \frac{1}{3} \sum_{k=1}^{3} \left(\sin\left(\frac{2\pi k}{3}\right)\right)^2 = (1.2)^2 0.5 = 0.72$$

$$E[v^2(k-1)] = E[v^2(k)] = 0.72$$

$$E[v(k)v(k-1)] = \frac{1}{3} \sum_{k=1}^{3} \left(1.2\ \sin\frac{2\pi k}{3}\right)\left(1.2\ \sin\frac{2\pi(k-1)}{3}\right)$$

$$= (1.2)^2 0.5 \cos\left(\frac{2\pi}{3}\right) = -0.36$$

$$\mathbf{R} = \begin{bmatrix} 0.72 & -0.36 \\ -0.36 & 0.72 \end{bmatrix}$$

فیلتر کردن وفقی

سیگنال‌ها

SIGNALS

فرض: سیگنال EEG سیگنال تصادفی سفید (یعنی: هر لحظه با لحظه‌ی بعدی ناهمبسته است)
توزیع یکنواخت 0.2 .. 0.2−، سینوسی ۶۰ هرتز نمونه‌برداری شده ۱۸۰ هرتز

$$m(k) = 1.2 \sin\left(\frac{2\pi k}{3} - \frac{3\pi}{4}\right)$$

$$v(k) = 1.2 \sin\left(\frac{2\pi k}{3}\right)$$ منبع نویز سینوسی

$$E[v^2(k)] = (1.2)^2 \frac{1}{3} \sum_{k=1}^{3} \left(\sin\left(\frac{2\pi k}{3}\right)\right)^2 = (1.2)^2 0.5 = 0.72$$

$k = 1, 2, 3$ به‌دلیل تناوب سینوسی

$$E[v^2(k-1)] = E[v^2(k)] = 0.72$$

$$E[v(k)v(k-1)] = \frac{1}{3} \sum_{k=1}^{3} \left(1.2 \sin\frac{2\pi k}{3}\right)\left(1.2 \sin\frac{2\pi(k-1)}{3}\right)$$

$$= (1.2)^2 0.5 \cos\left(\frac{2\pi}{3}\right) = -0.36$$

$$\mathbf{R} = \begin{bmatrix} 0.72 & -0.36 \\ -0.36 & 0.72 \end{bmatrix}$$

Stationary Point

$$E[(s(k)+m(k))v(k)] = E[s(k)v(k)] + E[m(k)v(k)]$$

$$0$$

$$E[m(k)v(k)] = \frac{1}{3}\sum_{k=1}^{3}\left(1.2\ \sin\left(\frac{2\pi k}{3} - \frac{3\pi}{4}\right)\right)\left(1.2\sin\frac{2\pi k}{3}\right) = -0.51$$

$$E[(s(k)+m(k))v(k-1)] = E[s(k)v(k-1)] + E[m(k)v(k-1)]$$

$$0$$

$$E[m(k)v(k-1)] = \frac{1}{3}\sum_{k=1}^{3}\left(1.2\ \sin\left(\frac{2\pi k}{3} - \frac{3\pi}{4}\right)\right)\left(1.2\ \sin\frac{2\pi(k-1)}{3}\right) = 0.70$$

$$\mathbf{h} = \begin{bmatrix} E[(s(k)+m(k))v(k)] \\ E[(s(k)+m(k))v(k-1)] \end{bmatrix} \qquad \mathbf{h} = \begin{bmatrix} -0.51 \\ 0.70 \end{bmatrix}$$

$$\mathbf{x}^* = \mathbf{R}^{-1}\mathbf{h} = \begin{bmatrix} 0.72 & -0.36 \\ -0.36 & 0.72 \end{bmatrix}^{-1} \begin{bmatrix} -0.51 \\ 0.70 \end{bmatrix} = \begin{bmatrix} -0.30 \\ 0.82 \end{bmatrix}$$

فیلتر کردن وفقی

نقطه‌ی ایستان

STATIONARY POINT

$$E[(s(k)+m(k))v(k)] = E[s(k)v(k)] + E[m(k)v(k)]$$

$$0$$

زیرا $s(k)$ و $v(k)$ مستقل با میانگین صفر هستند.

$$E[m(k)v(k)] = \frac{1}{3}\sum_{k=1}^{3}\left(1.2\ \sin\left(\frac{2\pi k}{3}-\frac{3\pi}{4}\right)\right)\left(1.2\sin\frac{2\pi k}{3}\right) = -0.51$$

$$E[(s(k)+m(k))v(k-1)] = E[s(k)v(k-1)] + E[m(k)v(k-1)]$$

$$0$$

$$E[m(k)v(k-1)] = \frac{1}{3}\sum_{k=1}^{3}\left(1.2\ \sin\left(\frac{2\pi k}{3}-\frac{3\pi}{4}\right)\right)\left(1.2\ \sin\frac{2\pi(k-1)}{3}\right) = 0.70$$

$$\mathbf{h} = \begin{bmatrix} E[(s(k)+m(k))v(k)] \\ E[(s(k)+m(k))v(k-1)] \end{bmatrix} \Rightarrow \mathbf{h} = \begin{bmatrix} -0.51 \\ 0.70 \end{bmatrix}$$

$$\mathbf{x}^* = \mathbf{R}^{-1}\mathbf{h} = \begin{bmatrix} 0.72 & -0.36 \\ -0.36 & 0.72 \end{bmatrix}^{-1}\begin{bmatrix} -0.51 \\ 0.70 \end{bmatrix} = \begin{bmatrix} -0.30 \\ 0.82 \end{bmatrix}$$

Performance Index

$$F(\mathbf{x}) = c - 2\mathbf{x}^T\mathbf{h} + \mathbf{x}^T\mathbf{R}\mathbf{x}$$

$$c = E[t^2(k)] = E[(s(k) + m(k))^2]$$

$$c = E[s^2(k)] + 2E[s(k)m(k)] + E[m^2(k)]$$

$$E[s^2(k)] = \frac{1}{0.4}\int_{-0.2}^{0.2} s^2 ds = \frac{1}{3(0.4)} s^3 \Big|_{-0.2}^{0.2} = 0.0133$$

$$E[m^2(k)] = \frac{1}{3}\sum_{k=1}^{3}\left\{1.2\ \sin\left(\frac{2\pi}{3} - \frac{3\pi}{4}\right)\right\}^2 = 0.72$$

$$c = 0.0133 + 0.72 = 0.7333$$

$$F(\mathbf{x}^*) = 0.7333 - 2(0.72) + 0.72 = 0.0133$$

فیلتر کردن وفقی

شاخص کارآیی

PERFORMANCE INDEX

$$F(\mathbf{x}) = c - 2\mathbf{x}^T\mathbf{h} + \mathbf{x}^T\mathbf{R}\mathbf{x}$$

$$c = E[t^2(k)] = E[(s(k) + m(k))^2]$$

$$c = E[s^2(k)] + 2\underbrace{E[s(k)m(k)]}_{0} + E[m^2(k)]$$

زیرا $s(k)$ و $v(k)$ مستقل با میانگین صفر هستند.

$$E[s^2(k)] = \frac{1}{0.4}\int_{-0.2}^{0.2} s^2 ds = \frac{1}{3(0.4)}s^3\Big|_{-0.2}^{0.2} = 0.0133$$

$$E[m^2(k)] = \frac{1}{3}\sum_{k=1}^{3}\left\{1.2\ \sin\left(\frac{2\pi}{3} - \frac{3\pi}{4}\right)\right\}^2 = 0.72$$

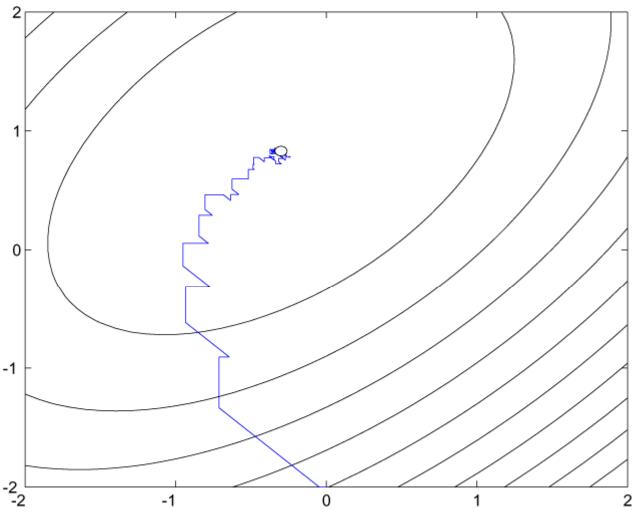
$$c = 0.0133 + 0.72 = 0.7333$$

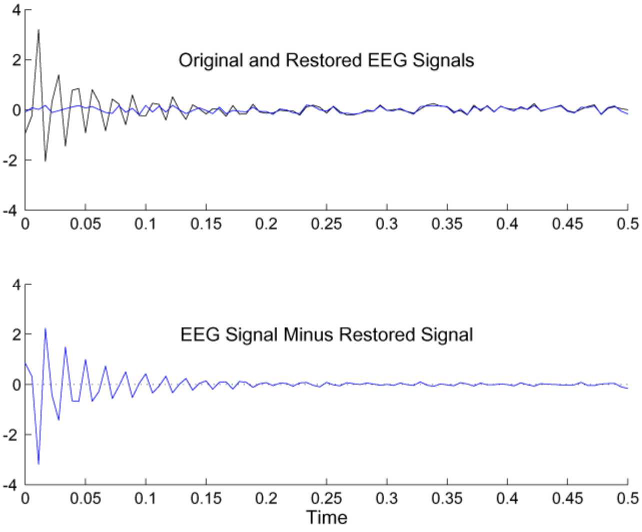
$$F(\mathbf{x}^*) = 0.7333 - 2(0.72) + 0.72 = 0.0133$$

خطای می‌نیمم مربعات (که شبیه مقدار متوسط مربعات سیگنال EEG است)

Prepared by Kazim Fouladi | Fall 2017 | 2nd Edition
کاظم فولادی

LMS Response

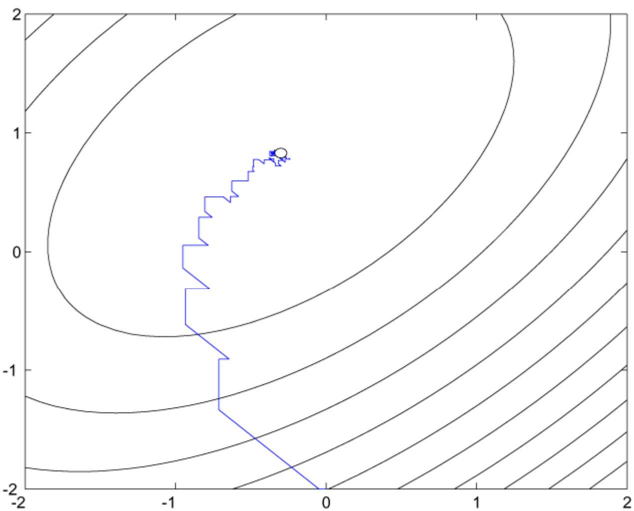


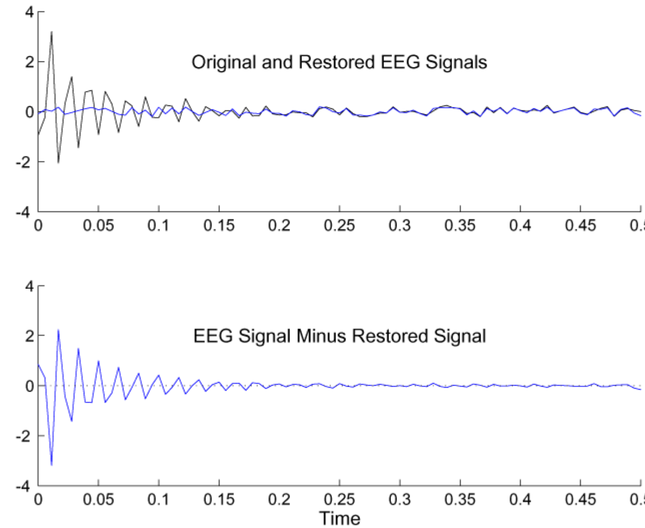


فیلتر کردن وفقی

پاسخ LMS

LMS Response





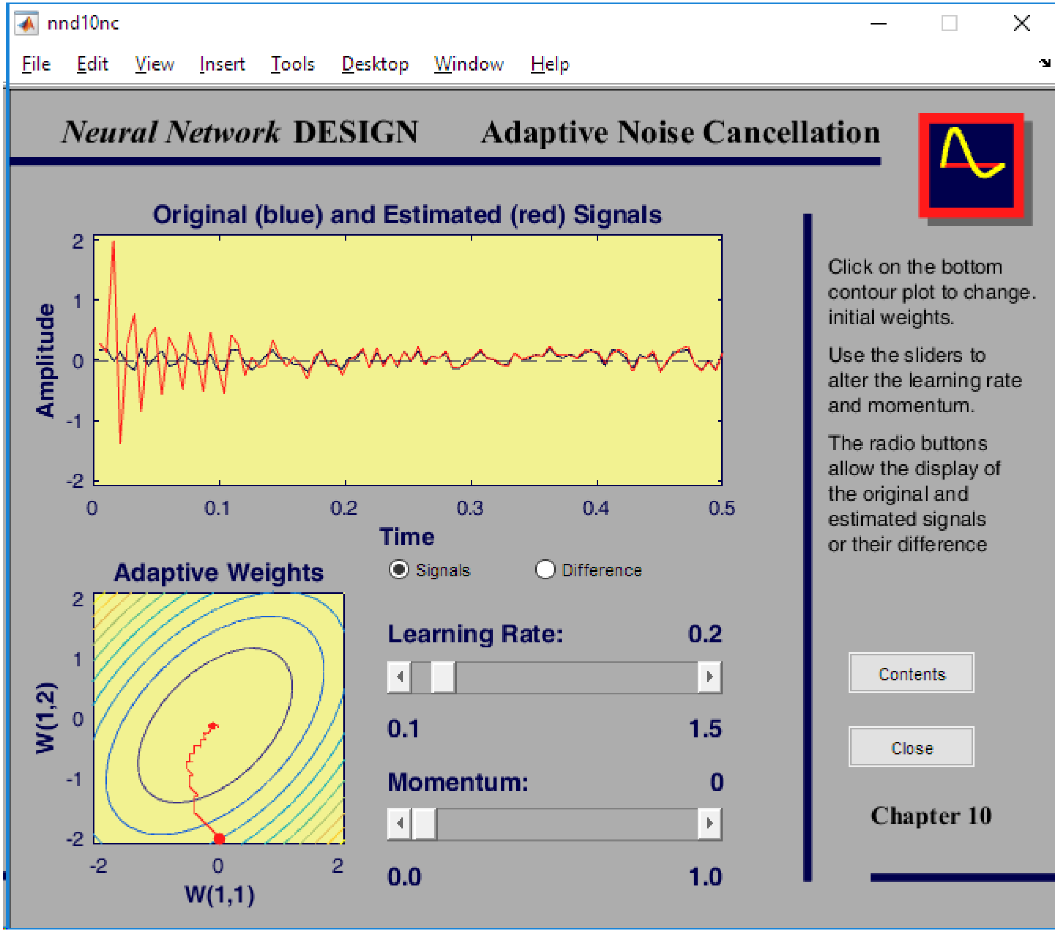
تراجکتوری الگوریتم LMS با $\alpha = 0.1$

حالت مشابه کاهش گرادیان، اما نویزی

چرا خطا صفر نمی‌شود؟

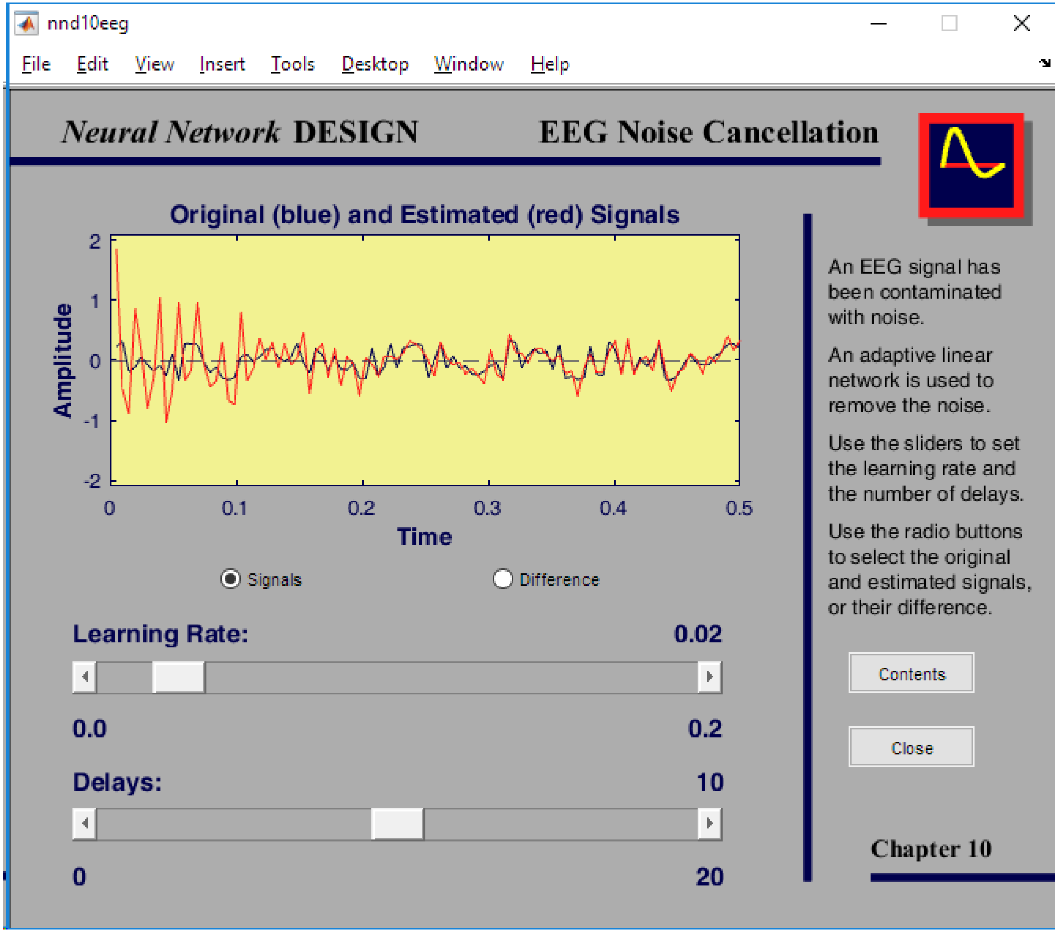
زیرا LMS از تقریب کاهش گرادیان بهره می‌برد،

نه مقدار دقیق آن.

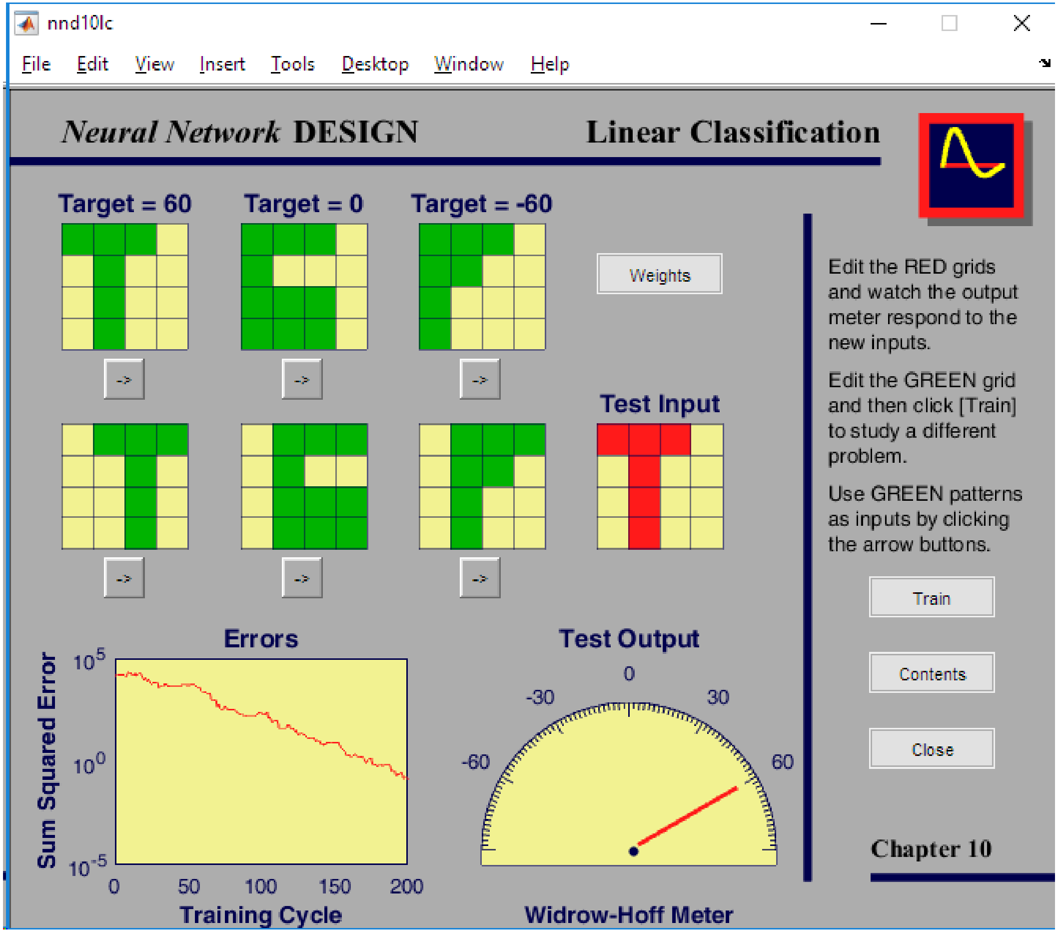


MATLAB

```
>> nnd10nc
```

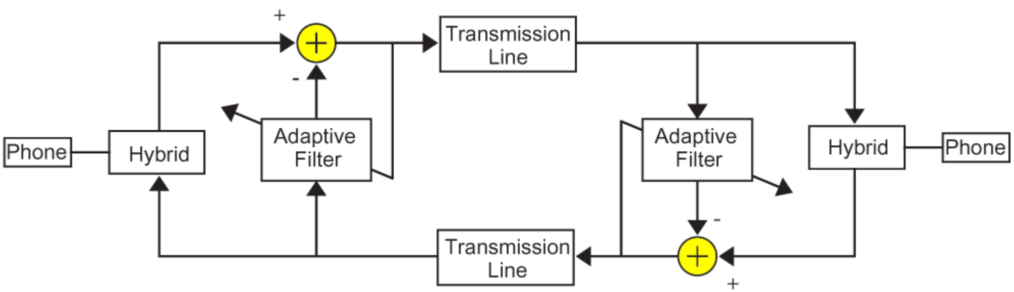

nnd10eeg
File Edit View Insert Tools Desktop Window Help
Neural Network DESIGN
EEG Noise Cancellation
Original (blue) and Estimated (red) Signals
Amplitude
Time
Signals
Difference
Learning Rate:
0.02
0.0
0.2
Delays:
10
0
20
An EEG signal has been contaminated with noise.
An adaptive linear network is used to remove the noise.
Use the sliders to set the learning rate and the number of delays.
Use the radio buttons to select the original and estimated signals, or their difference.
Contents
Close
Chapter 10

```
>> nnd10eeg
```


nnd10lc
File Edit View Insert Tools Desktop Window Help
Neural Network DESIGN
Linear Classification
Target = 60
Target = 0
Target = -60
Weights
Test Input
Edit the RED grids and watch the output meter respond to the new inputs.
Edit the GREEN grid and then click [Train] to study a different problem.
Use GREEN patterns as inputs by clicking the arrow buttons.
Train
Contents
Close
Errors
Sum Squared Error
Training Cycle
Test Output
Widrow-Hoff Meter
Chapter 10

>> nnd10lc

Echo Cancellation

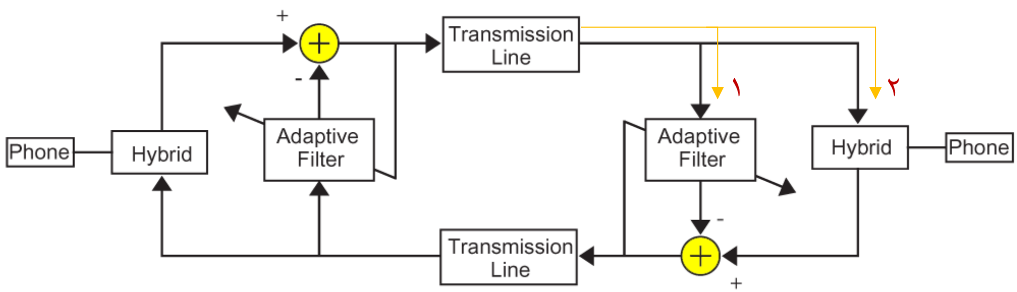


فیلتر کردن وفقی

مثال: لغو پژواک (لغوِ اکو)

Echo Cancellation

اکو در خطوط تلفن راه دور زیاد اتفاق می‌افتد.
(به‌دلیل عدم تطابق امپدانس در دستگاه‌های هیبرید که تقاطع میان خطوط راه دور و خطوط محلی را ایجاد می‌کنند.)

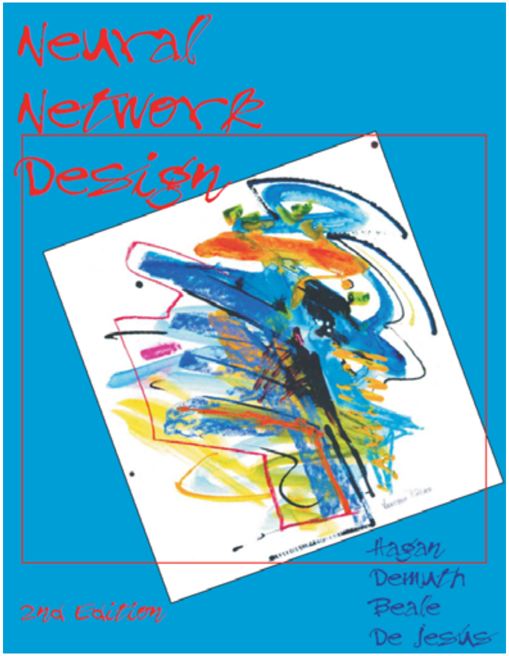


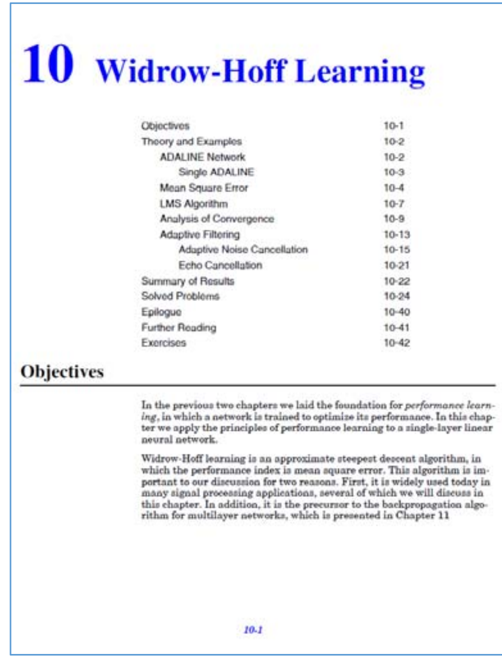
در انتهای خط راه دور، سیگنال وارده به یک فیلتر وفقی و همچنین به دستگاه هیبرید وارد می‌شود.
خروجی تارگت فیلتر، خروجی هیبرید است.
پس، فیلتر سعی می‌کند بخشی از خروجی هیبرید که همبسته با سیگنال ورودی است (یعنی اکو) را لغو کند.

۶

منابع

منبع اصلی





10 Widrow-Hoff Learning

Objectives

In the previous two chapters we laid the foundation for *performance learning*, in which a network is trained to optimize its performance. In this chapter we apply the principles of performance learning to a single-layer linear neural network.

Widrow-Hoff learning is an approximate steepest descent algorithm, in which the performance index is mean square error. This algorithm is important to our discussion for two reasons. First, it is widely used today in many signal processing applications, several of which we will discuss in this chapter. In addition, it is the precursor to the backpropagation algorithm for multilayer networks, which is presented in Chapter 11

10-1

Martin T. Hagan, Howard B. Demuth, Mark H. Beale, Orlando De Jesus,
Neural Network Design,
2nd Edition, Martin Hagan, 2014.
Chapter 10

Online version can be downloaded from: http://hagan.okstate.edu/nnd.html