Widrow-Hoff (α-LMS) Learning Rule

Another example of an error correcting rule with a quadratic criterion function is the Widrow-Hoff rule (Widrow and Hoff, 1960). This rule was originally used to train the linear unit, also known as the *adaptive linear combiner element* (ADALINE), shown in Figure 2-3. In this case, the output of the linear unit in response to the input $\mathbf{x}^k$ is simply $y^k = \left(\mathbf{x}^k\right)^{\mathrm{T}} \mathbf{w}$. The Widrow-Hoff rule was proposed originally as an ad hoc rule which embodies the so-called minimal disturbance principle. Later, it was discovered (Widrow and Stearns, 1985) that this rule converges in the mean square to the solution $\mathbf{w}^*$ that corresponds to the least- mean-square (LMS) output error if all

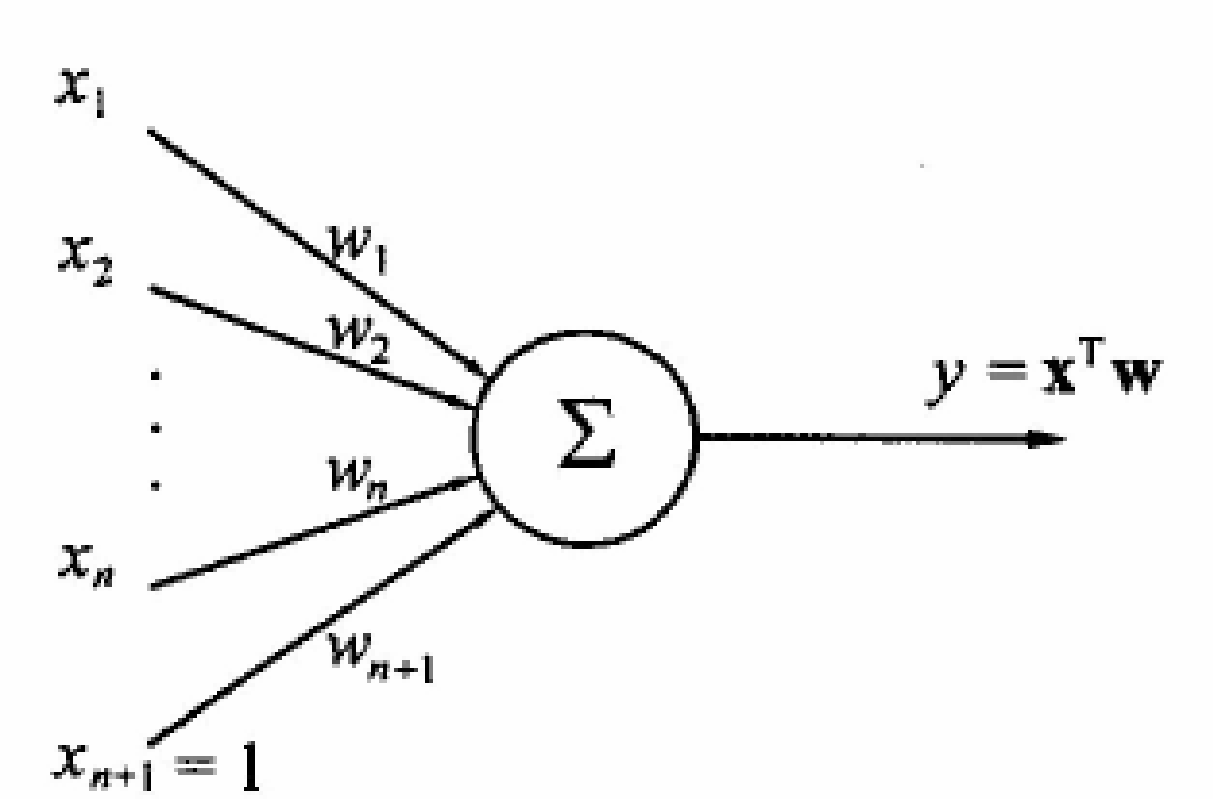

Figure 2-3 Adaptive linear combiner clement (ADALINE).

input patterns are of the same length (i.e., $\|\mathbf{x}^k\|$ is the same for all *k)*. Therefore, this rule is sometimes referred to as the α-LMS *rule* (the α is used here to distinguish this rule from another very similar rule that is discussed in next section). The α-LMS rule is given by

$$\begin{cases} \mathbf{w}^1 = \mathbf{0} \text{ or arbitrary} \\ \mathbf{w}^{k+1} = \mathbf{w}^k + \alpha\left(d^k - y^k\right)\dfrac{\mathbf{x}^k}{\|\mathbf{x}^k\|^2} \end{cases} \qquad (2.30)$$

where $d^k \in R$ is the desired response, and $\alpha > 0$. Equation (2.30) is similar to the perceptron rule if one sets ρ in Equation (2.2) as

$$\rho = \rho^k = \frac{\alpha}{\|\mathbf{x}^k\|^2} \qquad (2.31)$$

However, the error in Equation (2.30) is measured at the linear output, not after the nonlinearity, as in the perceptron. The constant α controls the stability and speed of convergence (Widrow

and Stearns, 1985; Widrow and Lehr, 1990). If the input vectors are independent over time, stability is ensured for most practical purposes if $0 < \alpha < 2$.

As for Mays' rule, this rule is self-normalizing in the sense that the choice of α does not depend on the magnitude of the input vectors. Since the α-LMS rule selects $\Delta\mathbf{w}^k$ to be collinear with $\mathbf{x}^k$, the desired error correction is achieved with a weight change of the smallest possible magnitude. Thus, when adapting to learn a new training sample, the responses to previous training samples are, on average, minimally disturbed. This is the basic idea behind the minimal disturbance principle on which the α-LMS is founded. Alternatively, one can show that the α-LMS learning rule is a gradient descent minimizer of an appropriate quadratic criterion function

Other Gradient-Descent-Based Learning Rules

In the following, additional learning rules for single-unit training are derived. These rules are derived systematically by first defining an appropriate criterion function and then optimizing such a function by an iterative gradient search procedure.

μ-LMS Learning Rule

The μ-LMS learning rule (Widrow and Hoff, 1960) represents the most analyzed and most applied simple learning rule. It is also of special importance due to its possible extension to learning in multiple unit neural nets. Therefore, special attention is given to this rule in this chapter. In the following, the μ-LMS rule is described in the context of the linear unit in Figure 2-3. Let

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{m}\left(d^i - y^i\right)^2 \qquad (2.32)$$

be the sum of squared error (SSE) criterion function, where

$$y^i = \left(\mathbf{x}^i\right)^{\mathrm{T}}\mathbf{w} \qquad (2.33)$$

Now, using steepest gradient-descent search to minimize $J(\mathbf{w})$ in Equation (2.32) gives

$$\begin{aligned}\mathbf{w}^{k+1} &= \mathbf{w}^{k} - \mu \nabla J\left(\mathbf{w}\right) \\ &= \mathbf{w}^{k} + \mu \sum_{i=1}^{m} \left(d^{i} - y^{i}\right)\mathbf{x}^{i}\end{aligned} \tag{2.34}$$

The criterion function $J(\mathbf{w})$ in Equation (2.32) is quadratic in the weights because of the linear relation between y^i and $\mathbf{w}$. In fact, $J(\mathbf{w})$ defines a convex[1] hyperparaboloidal surface with a single minimum $\mathbf{w}^*$ (the global minimum). Therefore, if the positive constant μ is chosen sufficiently small, the gradient-descent search implemented by Equation (2.34) will asymptotically converge toward the solution $\mathbf{w}^*$ regardless of the setting of the initial search point $\mathbf{w}^1$. The learning rule in Equation (2.34) is sometimes referred to as the *batch LMS rule.*

The incremental version of Equation (2.34), known as the *μ-LMS* or LMS *rule*, is given by

1 A function of the form $f : R^n \rightarrow R$ is said to be convex if the following condition is satisfied:

$$(1-\lambda) f(\mathbf{u}) + \lambda f(\mathbf{v}) \geq f\left[(1-\lambda)\mathbf{u} + \lambda \mathbf{v}\right]$$

for any pair of vectors **u** and **v** in R^n and any real number λ in the closed interval [0,1].

$$\begin{cases} \mathbf{w}^1 = 0 \ \text{ or arbitrary} \\ \mathbf{w}^{k+1} = \mathbf{w}^k + \mu\left(d^k - y^k\right)\mathbf{x}^k \end{cases} \qquad (2.35)$$

Note that this rule becomes identical to the α-LMS learning rule in Equation (2.30) upon setting μ as

$$\mu = \mu^k = \frac{\alpha}{\left\|\mathbf{x}^k\right\|^2} \qquad (2.36)$$

Also, when the input vectors have the same length, as would be the case when $\mathbf{x} \in \{-1,+1\}^n$, then the α-LMS rule becomes identical to the μ-LMS rule. Since the α-LMS learning algorithm converges when $0 < \alpha < 2$, we can start from Equation (2.36) and calculate the required range on μ for ensuring the convergence of the μ-LMS rule for "most practical purposes":

$$0 < \mu < \frac{2}{\max_i \left\|\mathbf{x}^i\right\|^2} \qquad (2.37)$$

For input patterns independent over time and generated by a stationary process, convergence of the mean of the weight vector $\langle \mathbf{w}^k \rangle$ is ensured if the fixed learning rate μ is chosen to be smaller than $2/\langle \|\mathbf{x}\|^2 \rangle$ (Widrow and Stearns, 1985). Here, $\langle \bullet \rangle$ signifies the "mean" or expected value. In this case, $\langle \mathbf{w}^k \rangle$ approaches a solution $\mathbf{w}^*$ as $k \to \infty$. Note that the bound $2/\langle \|\mathbf{x}\|^2 \rangle$ is less restrictive than the one in Equation (2.37). The bound $2/\left(3\langle \|\mathbf{x}\|^2 \rangle\right)$ on μ guarantees the convergence of $\mathbf{w}$ in the mean square (i.e., $\langle \|\mathbf{w}^k - \mathbf{w}^*\|^2 \rangle \to 0$ as $k \to \infty$) for input patterns generated by a zero-mean Gaussian process independent over time. It should be noted that convergence in the mean square implies convergence in the mean; however, the converse is not necessarily true. The assumptions of decorrelated patterns and stationarity are not necessary conditions for the convergence of μ-LMS. For example, Macchi and Eweda (1983) have a much stronger result regarding convergence of the μ-LMS rule which is even valid when a finite number of successive training patterns are strongly correlated.

In practical problems, $m > n+1$; hence it becomes impossible to satisfy all requirements $\left(\mathbf{x}^k\right)^{\mathrm{T}} \mathbf{w} = d^k, k = 1, 2, ..., m$. Therefore, Equation (2.35) never converges. Thus, for convergence, μ is set to $\mu_0 / k > 0$, where μ_0 is a small positive constant. In applications such as linear filtering, though, the decreasing step size is not very valuable, because it cannot accommodate nonstationarity in the input signal. Indeed, $\mathbf{w}^k$ will essentially stop changing for large *k*, which precludes the tracking of time variations. Thus the fixed-increment (constant μ) LMS learning rule has the advantage of limited memory, which enables it to track time fluctuations in the input data.

When the learning rate μ is sufficiently small, the μ-LMS rule becomes a "good" approximation to the gradient-descent rule in Equation (2.34). This means that the weight vector $\mathbf{w}^k$ will tend to move toward the global minimum $\mathbf{w}^*$ of the convex SSE criterion function. Next, we show that $\mathbf{w}^*$ is given by

$$\mathbf{w}^* = \mathbf{X}^\dagger \mathbf{d} \qquad (2.38)$$

where $\mathbf{X} = \left[\mathbf{x}^1 \;\; \mathbf{x}^2 \;\; ... \;\; \mathbf{x}^m \right]$, $\mathbf{d} = \left[d^1 \;\; d^2 \;\; ... \;\; d^m \right]^{\mathrm{T}}$, and $\mathbf{X}^{\dagger} = \left(\mathbf{X}\mathbf{X}^{\mathrm{T}} \right)^{-1} \mathbf{X}$ is the generalized inverse or pseudoinverse (Penrose, 1955) of $\mathbf{X}$ for $m > n+1$.

The extreme points (minima and maxima) of the function $J(\mathbf{w})$ are solutions to the equation

$$\nabla J\left(\mathbf{w}\right) = \mathbf{0} \qquad (2.39)$$

Therefore, any minimum of the SSE criterion function in Equation (2.32) must satisfy

$$\nabla J\left(\mathbf{w}\right) = -\sum_{i=1}^{m} \left[d^i - \left(\mathbf{x}^i\right)^{\mathrm{T}} \mathbf{w} \right] \mathbf{x}^i = \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{w} - \mathbf{d}\right) = \mathbf{0} \qquad (2.40)$$

Equation (2.40) can be rewritten as

$$\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{w} = \mathbf{X}\mathbf{d} \qquad (2.41)$$

which for a nonsingular matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ gives the solution in Equation (2.38), or explicitly

$$\mathbf{w}^* = \left(\mathbf{X}\mathbf{X}^{\mathrm{T}}\right)^{-1}\mathbf{X}\mathbf{d} \qquad (2.42)$$

Recall that just because $\mathbf{w}^*$ in Equation (2.42) satisfies the condition $\nabla J\left(\mathbf{w}^*\right) = \mathbf{0}$, this does not guarantee that $\mathbf{w}^*$ is a local minimum of the criterion function *J*. It does, however, considerably narrow the choices in that such $\mathbf{w}^*$ represents (in a local sense) either a point of minimum, maximum, or saddle point of *J*. To verify that $\mathbf{w}^*$ is actually a minimum of .*J*(**w**), we may evaluate the second derivative or Hessian matrix

$$\nabla\nabla J = \left[\frac{\partial^2 J}{\partial w_i \partial w_j}\right]$$

of *J* at $\mathbf{w}^*$ and show that it is positive definite[2]. But this result follows immediately after noting that $\nabla\nabla J$ is equal to the positive-definite matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$. Thus $\mathbf{w}^*$ is a minimum of *J*.[3]

---

[2] An $n \times n$ real symmetric matrix **A** is positive-definite if the quadratic form $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}$ is strictly positive for all nonzero column vectors **x** in R^n.

[3] Of course, the same result could have been achieved by noting that the convex, unconstrained quadratic nature of *J*(**w**) admits one extreme point $\mathbf{w}^*$, which must be the

The LMS rule also may be applied to synthesize the weight vector **w** of a perceptron for solving two-class classification problems. Here, one starts by training the linear unit in Figure 2- with the given training pairs $\{\mathbf{x}^k, d^k\}, k = 1, 2, ..., m$, using the LMS rule. During training, the desired target d^k is set to +1 for one class and to -1 for the other class. (In fact, any positive constant can be used as the target for one class, and any negative constant can be used as the target for the other class.) After convergence of the learning process, the solution vector obtained may now be used in the perceptron for classification. Because of the thresholding nonlinearity in the perceptron, the output of the classifier will now be properly restricted to the set $\{-1, +1\}$.

When used as a perceptron weight vector, the minimum SSE solution in Equation (2.42) does not generally minimize the perceptron classification error rate. This should not be surprising, since the SSE criterion function is not designed to constrain its minimum inside the linearly separable solution region. Therefore, this solution does not necessarily represent a linear separable solution, even when the training set is linearly separable (this is further explored

---

global minimum of $J(\mathbf{w})$.

above). However, when the training set is nonlinearly separable, the solution arrived at may still be a useful approximation. Therefore, by employing the LMS rule for perceptron training, linear separability is sacrificed for good compromise performance on both separable and nonseparable problems.

Example 2.1 This example presents the results of a set of simulations that should help give some insight into the dynamics of the batch and incremental LMS learning rules. Specifically, we are interested in comparing the convergence behavior of the discrete-time dynamical systems in Equations (2.34) and (2.35). Consider the training set depicted in Figure 2-4 for a simple mapping problem. The 10 squares and 10 filled circles in this figure are positioned at the points whose coordinates (x_1, x_2) specify the two components of the input vectors. The squares and circles are to be mapped to the targets +1 and −1, respectively. For example, the left-most square in the figure represents the training pair $\{[1,0]^T, 1\}$. Similarly, the right most circle represents the training pair $\{[2,2]^T, -1\}$.

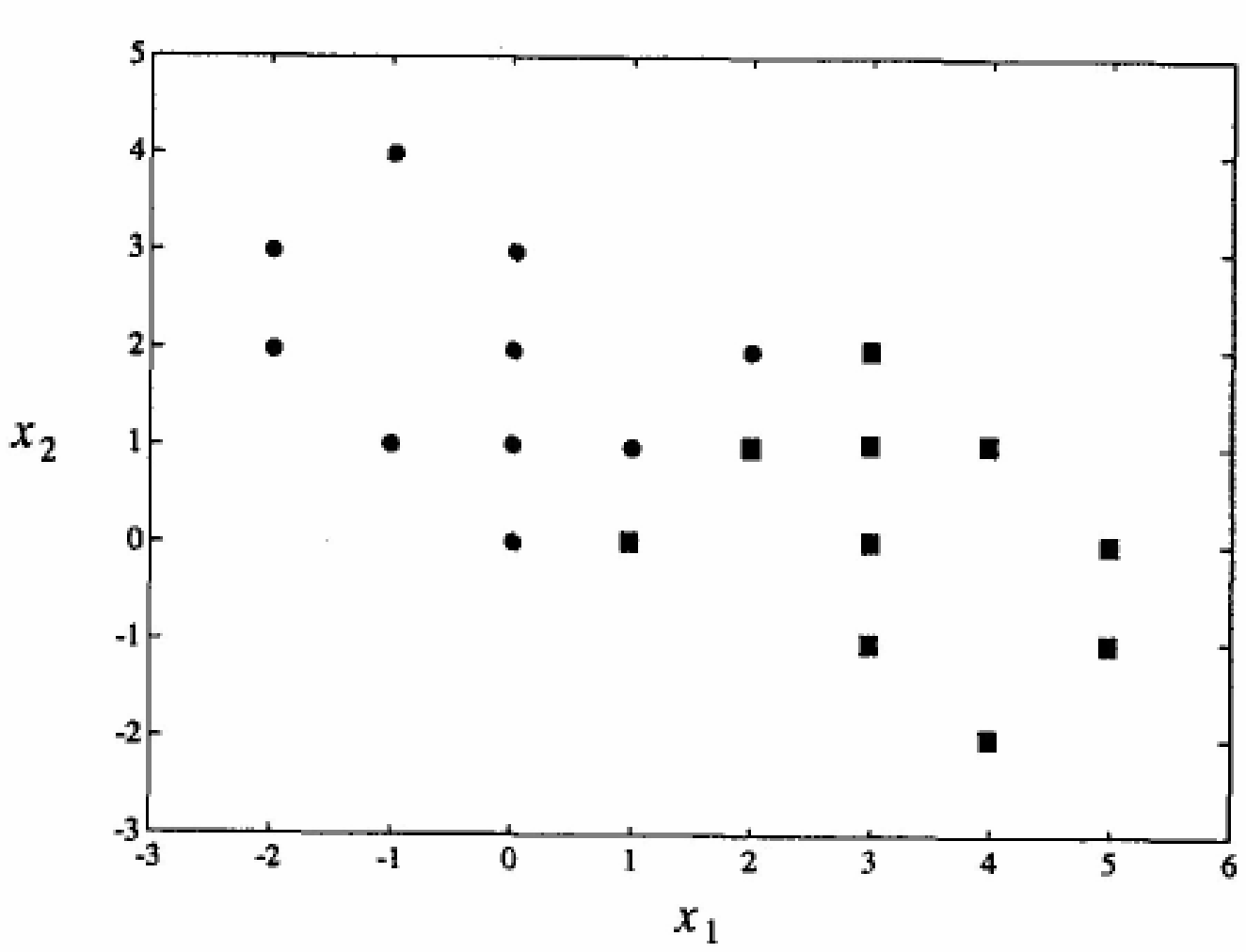

Figure 2-4 A 20-sample training set used in the simulations associated with Example 2.1. Points signified by a square and a filled circle should map into +1 and -1, respectively.

Figure 2-5 shows plots for the evolution of the square of the distance between the vector $\mathbf{w}^k$ and the (computed) minimum SSE solution $\mathbf{w}^*$ for batch LMS (dashed line) and incremental LMS (solid line). In both simulations, the learning rate (step size) μ was set to 0.005. The initial search point $\mathbf{w}^1$ was set to $[0,0]^T$. For the incremental LMS rule, the training examples are selected randomly from the training set. The batch LMS rule converges to the optimal solution $\mathbf{w}^*$ in less than 100 steps. Incremental LMS requires more learning steps, on the order of 2000 steps, to converge to a small neighborhood of $\mathbf{w}^*$.

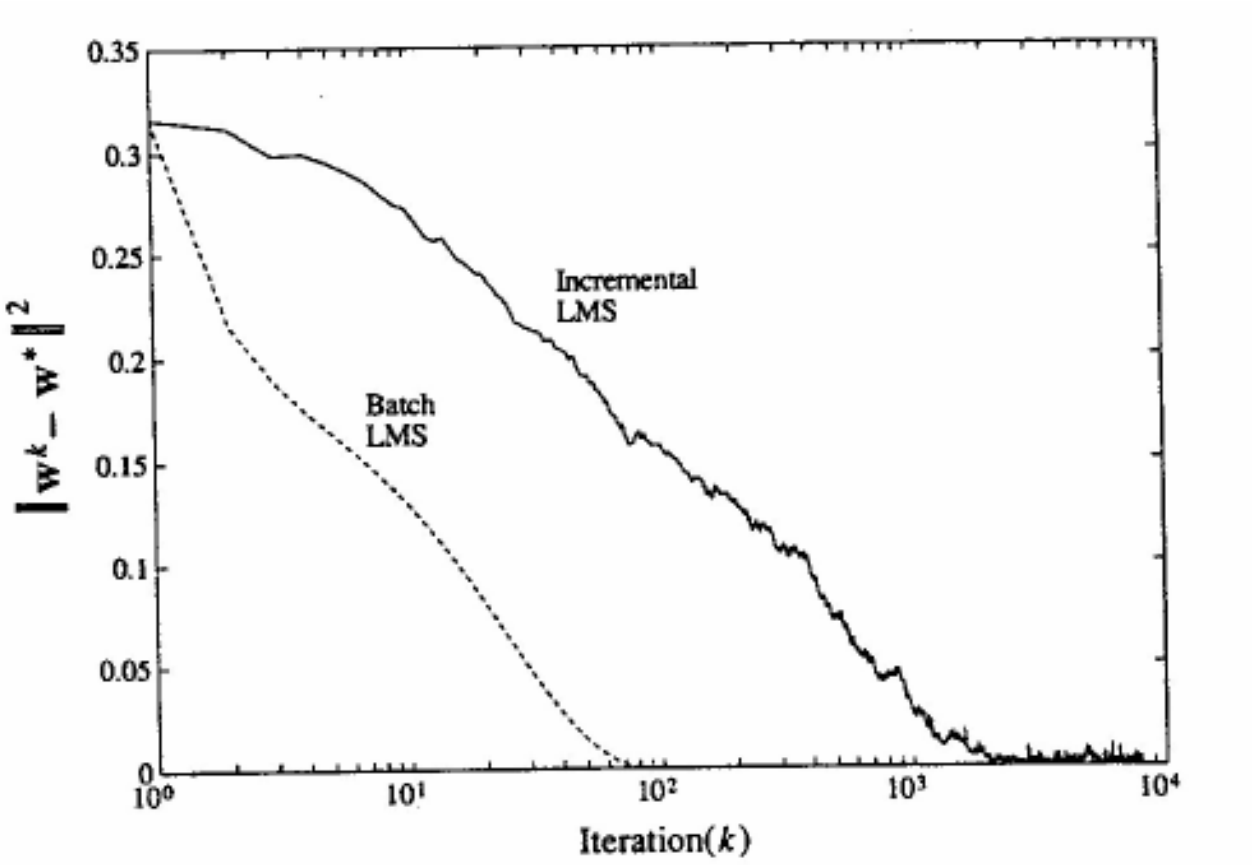

Figure 2-5 Plots (learning curves) for the square of the distance between the search point $\mathbf{w}^k$ and the minimum SSE solution $\mathbf{w}^*$ generated using two versions of the LMS learning rule. The dashed line corresponds to the batch LMS rule in Equation (2.34). The solid line corresponds to the incremental LMS rule in Equation (2.35) with a random order of presentation of the training patterns. In both cases, $w^1 = 0$ and $\mu = 0.005$ are used. Note the logarithmic scale for the iteration number *k***.**

The fluctuations in $\left\|\mathbf{w}^k - \mathbf{w}^*\right\|^2$ in this neighborhood are less than 0.02, as can be seen from Figure 2-5. The effect of a deterministic order of presentation of the training examples on the incremental LMS rule is shown by the solid line in Figure 2-6. Here, the training examples are presented in a predefined order, which did not change during training. The same initialization and step size are used as before. In order to allow for a more meaningful comparison between the two LMS rule versions, one learning step of incremental LMS is taken to mean a full cycle through the 20 samples. For comparison, the simulation result with batch LMS learning is plotted in the figure (see dashed line). These results indicate a very similar behavior in the convergence characteristics of incremental and batch LMS learning. This is so because of the small step size used. Both cases show asymptotic convergence toward the optimal solution $\mathbf{w}^*$, but with a relatively faster convergence of the batch LMS rule near $\mathbf{w}^*$. This is attributed to the use of more accurate gradient information.

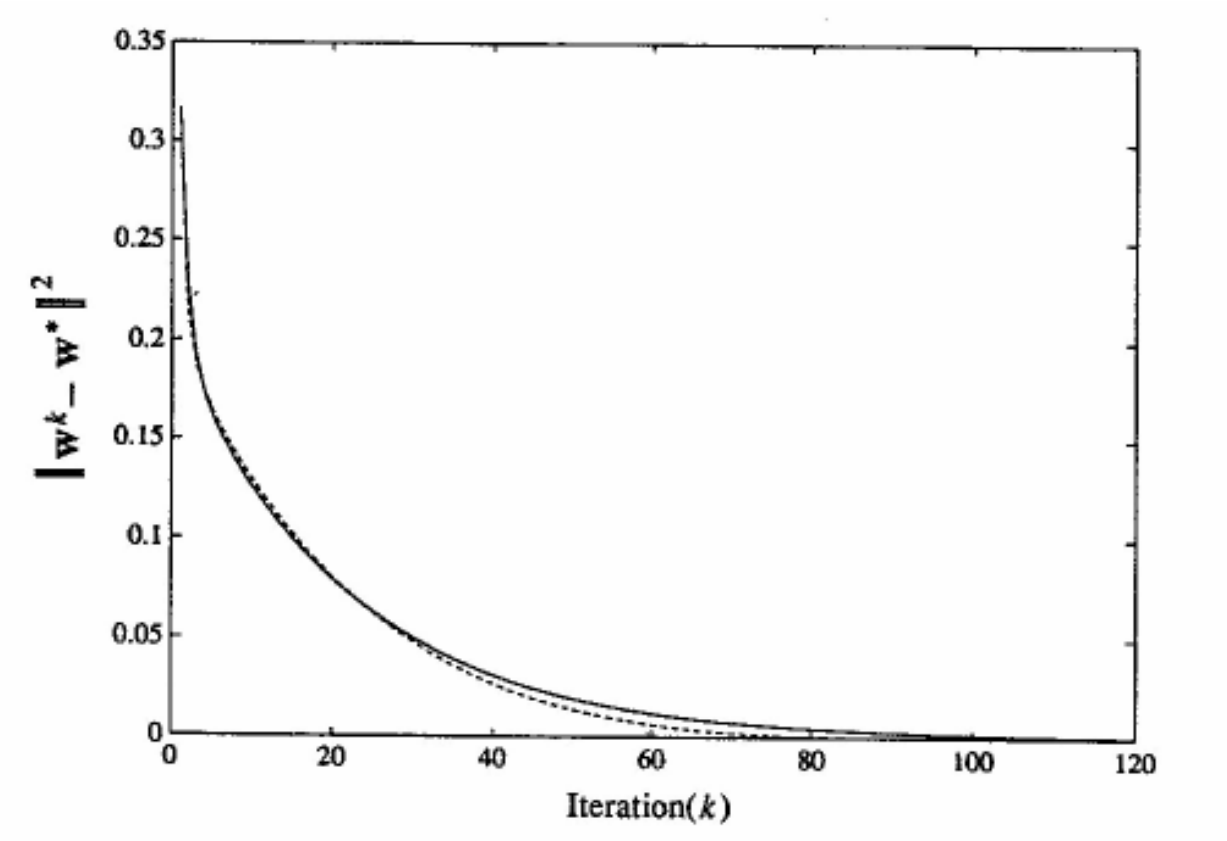

Figure 2-6 Learning curves for the batch LMS (dashed line) and incremental LMS (solid line) learning rules for the data in Figure 2-5. The result for the batch LMS rule shown here is identical to the one shown in Figure 2-5 (this result looks different only because of the present use of a linear scale for the horizontal axis). The incremental LMS rule results shown assume a deterministic, fixed order of presentation of the training patterns. Also, for the incremental LMS case, $\mathbf{w}^k$ represents the weight vector after the completion of the kth learning "cycle." Here, one cycle corresponds to 20 consecutive learning iterations.

The μ-LMS as a Stochastic Process

Stochastic approximation theory may be employed as an alternative to the deterministic gradient-descent analysis presented thus far. It has the advantage of naturally arriving at a learning-rate schedule ρ^k for asymptotic convergence in the mean square. Here, one starts with the mean-square error (MSE) criterion function:

$$J(\mathbf{w}) = \frac{1}{2}\left\langle \left(\mathbf{x}^{\mathrm{T}}\mathbf{w} - d\right)^2 \right\rangle \qquad (2.43)$$

where again $\langle \bullet \rangle$ denotes the mean (expectation) over all training vectors. Now one may compute the gradient of J as

$$\nabla J(\mathbf{w}) = \left\langle \left(\mathbf{x}^{\mathrm{T}}\mathbf{w} - d\right)\mathbf{x} \right\rangle \qquad (2.44)$$

which upon setting to zero allows us to find the minimum $\mathbf{w}^*$ of J in Equation (2.43) as the solution of

which gives $\left\langle \mathbf{x}\mathbf{x}^{\mathrm{T}} \right\rangle \mathbf{w}^{*} = \left\langle d\mathbf{x} \right\rangle$

$$\mathbf{w}^{*} = \mathbf{C}^{-1}\mathbf{P} \qquad (2.45)$$

where $\mathbf{C} \triangleq \left\langle \mathbf{x}\mathbf{x}^{\mathrm{T}} \right\rangle$ and $\mathbf{P} \triangleq \left\langle d\mathbf{x} \right\rangle$. Note that the expected value of a vector or a matrix is found by taking the expected values of its components. We refer to **C** as the *auto-correlation matrix* of the input vectors and to **P** as the *cross-correlation vector* between the input vector **x** and its associated desired target *d*. In Equation (2.45), the determinant of **C**, $|\mathbf{C}|$, is assumed different from zero. The solution $\mathbf{w}^{*}$ in Equation (2.45) is sometimes called the *Wiener weight vector* (Widrow and Stearns, 1985). It represents the minimum MSE solution, also known as the *least-mean-square (LMS) solution.*

It is interesting to note here the close relation between the minimum SSE solution in Equation (2.42) and the LMS or minimum MSE solution in Equation (2.45). In fact, one can show that when the size of the training set *m* is large, the minimum SSE solution converges to the minimum MSE solution.

First, let us express $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ as the sum of vector outer products $\sum_{k=1}^{m}\mathbf{x}^{k}\left(\mathbf{x}^{k}\right)^{\mathrm{T}}$. We can also rewrite $\mathbf{X}\mathbf{d}$ as $\sum_{k=1}^{m}d^{k}\mathbf{x}^{k}$. This representation allows us to express Equation (2.42) as

$$\mathbf{w}^{*}=\left[\sum_{k=1}^{m}\mathbf{x}^{k}\left(\mathbf{x}^{k}\right)^{\mathrm{T}}\right]^{-1}\left(\sum_{k=1}^{m}d^{k}\mathbf{x}^{k}\right)$$

Now, multiplying the right-hand side of the preceding equation by *m/m* allows us to express it as

$$\mathbf{w}^{*}=\left[\frac{1}{m}\sum_{k=1}^{m}\mathbf{x}^{k}\left(\mathbf{x}^{k}\right)^{\mathrm{T}}\right]^{-1}\left(\frac{1}{m}\sum_{k=1}^{m}d^{k}\mathbf{x}^{k}\right)$$

Finally, if *m* is large, the averages

$$\frac{1}{m}\sum_{k=1}^{m}\mathbf{x}^{k}\left(\mathbf{x}^{k}\right)^{\mathrm{T}} \qquad \text{and} \quad \frac{1}{m}\sum_{k=1}^{m}d^{k}\mathbf{x}^{k}$$

become very good approximations of the expectations $\mathbf{C} = \left\langle \mathbf{x}\mathbf{x}^{\mathrm{T}} \right\rangle$ and $\mathbf{P} = \left\langle d\mathbf{x} \right\rangle$, respectively. Thus we have established the equivalence of the minimum SSE and minimum MSE for a large training set.

Next, in order to minimize the MSE criterion, one may employ a gradient-descent procedure where, instead of the expected gradient in Equation (2.44), the instantaneous gradient $\left[\left(\mathbf{x}^k \right)^{\mathrm{T}} \mathbf{w}^k - d^k \right] \mathbf{x}^k$ is used. Here, at each learning step the input vector $\mathbf{x}$ is drawn at random. This leads to the stochastic process

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \rho^k \left[d^k - \left(\mathbf{x}^k \right)^{\mathrm{T}} \mathbf{w}^k \right] \mathbf{x}^k \qquad (2.46)$$

which is the same as the μ-LMS rule in Equation (2.35) except for a variable learning rate ρ^k. It can be shown that if $|\mathbf{C}| \neq 0$ and ρ^k satisfies the three conditions

1. $\rho^k \geq 0$ (2.47a)

2. $\lim_{m\to\infty} \sum_{k=1}^{m} \rho^k = +\infty$ (2.47b)

3. $\lim_{m\to\infty} \sum_{k=1}^{m} \left(\rho^k\right)^2 < \infty$ (2.47c)

then $\mathbf{w}^k$ converges to $\mathbf{w}^*$ in Equation (2.45) asymptotically in the mean square; i.e.,

$$\lim_{k\to\infty} \left\langle \left\| \mathbf{w}^k - \mathbf{w}^* \right\|^2 \right\rangle = 0 \qquad (2.48)$$

The criterion function in Equation (2.43) is of the form $\left\langle g\left(\mathbf{w},\mathbf{x}\right)\right\rangle$ and is known as a *regression function.* The iterative algorithm in Equation (2.46) is also known as a *stochastic approximation procedure* (or Kiefer-Wolfowitz or Robbins-Monro procedure). For a thorough discussion of stochastic approximation theory, the reader is referred to Wasan (1969).