CHAPTER 2

# Mathematical Fundamentals for ANN Study

*Unless you try to do something beyond what you have already mastered, you will never grow.*
Ralph Waldo Emerson

This chapter reviews a number of topics from linear algebra and geometry that will prove useful in developing, training, visualizing, and analyzing ANNs. Readers familiar with specific concepts may wish to skip relevant portions. In addition, readers (and instructors) may wish to refer back to specific sections as they progress through the subsequent chapters.

## 2.1 VECTOR AND MATRIX FUNDAMENTALS

In the study of ANNs matrices, vectors, and vector functions provide frameworks for both visualization and computation. Many concepts involve the visualization of points in $d$-dimensional space. We are comfortable when $d \leq 3$, since this corresponds to the physical world.

### 2.1.1 Elementary Matrices

The simplest characterization of an $n \times m$–dimensional matrix $A$ is a rectangular arrangement of $nm$ entities (real or complex) in an array of $n$ rows each with $m$ elements, or $m$ columns each with $n$ elements. This is denoted

$$A = [a_{ij}] \qquad i = 1, 2, \ldots, n \quad j = 1, 2, \ldots, m \tag{2.1}$$

where $a_{ij}$ is the $ij$th element, residing at the intersection of the $i$th row and the $j$th column. The following are special cases:

1. If $m = n$, the matrix is square.
2. If $m = 1$, the matrix is a column vector; if $n = 1$, the matrix is a row vector.[1] Vectors are denoted with an underbar; i.e., $\underline{x}$ is a vector. Often we are also careful to indicate the length of the vector, which is also the dimension of the vector space within which the vector resides.
3. If $m = n = 1$, the matrix is a scalar.
4. The transpose of matrix $A$, denoted $A^T$, is obtained by interchanging rows and columns, i.e.,

$$A^T = [a_{ji}] \qquad j = 1, 2, \ldots, m \quad i = 1, 2, \ldots, n \tag{2.2}$$

5. If $A = A^T$ (or $a_{ij} = a_{ji}$), the matrix is *symmetric*. $A$ must be square to be symmetric.
6. It is often convenient to *partition* a matrix, for both visualization and computation. For example, the $n \times m$ matrix $A$ may be partitioned as

$$A = \begin{pmatrix} \overset{p\times q}{A_1} & \overset{p\times(m-q)}{A_2} \\ \overset{(n-p)\times q}{A_3} & \overset{(n-p)\times(m-q)}{A_4} \end{pmatrix} \tag{2.3}$$

where $A_1$ is $p \times q$, $A_2$ is $p \times (m-q)$, $A_3$ is $(n-p) \times q$, and $A_4$ is $(n-p) \times (m-q)$, and $p \geq 1, q \geq 1$.

7. The most important partitioning of a matrix is into an array of *column vectors*. This is denoted

$$A = [\underline{a}_1 \quad \underline{a}_2 \quad \ldots \quad \underline{a}_m] \tag{2.4}$$

where $\underline{a}_i (i = 1, 2, \ldots, m)$ is an $n \times 1$–dimensional column vector. Columns of a matrix are easily accessed in MATLAB using "colon" notation. For example, `a(:,3)` denotes the third column of matrix `a`.

Elementary matrix operations include addition (subtraction), multiplication, and scaling. For example, for computer implementation the matrix product $\overset{m\times p}{C} = \overset{m\times n}{A}\,\overset{n\times p}{B}$ is formed via

$$C = [c_{ij}] \qquad i = 1, 2, \ldots, m \quad j = 1, 2, \ldots, p \tag{2.5}$$

where

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \tag{2.6}$$

Matrix dimensions must be *conformable* for the specific operation. For example, an $n \times m$ matrix postmultiplied by an $m \times p$ matrix is conformable under multiplication and yields an $n \times p$ product. This serves as a simplistic check on the validity of derivations using matrices and vectors.

---

[1] Throughout we will consider vectors, by default, to be column vectors; thus, a row vector $\underline{x}$ is denoted $\underline{x}^T$.

### 2.1.2 Vectors

For a positive integer $d$, let $R^d$ be the set of all ordered $n$-tuples of the form

$$\{x_1, x_2, \ldots, x_d\} \tag{2.7}$$

These could be viewed as the coordinates of a point $x$ in $d$-dimensional space. Of course, the coordinate system must be specified for this interpretation to be meaningful. The $x_i$ coordinates may be arranged in a vector, yielding

$$\underline{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} \tag{2.8}$$

### 2.1.3 Linearity

A mapping, $\underline{f}(\underline{x})$, is linear[2] if superposition holds, i.e.,

$$\underline{x} = \alpha \underline{x}_1 + \beta \underline{x}_2 \Rightarrow \underline{f}(\underline{x}) = \alpha \underline{f}(\underline{x}_1) + \beta \underline{f}(\underline{x}_2) \tag{2.9}$$

#### Vector-matrix equations

There are several interpretations of the vector-matrix equation[3]

$$\overset{n\times d}{A}\ \overset{d\times 1}{\underline{x}} = \overset{n\times 1}{\underline{y}} \tag{2.10}$$

Each may be viewed somewhat abstractly. Notice that we have not stated any relationship between $n$ and $d$. Three cases are possible:

$$n = d \tag{2.11}$$

$$n < d \tag{2.12}$$

$$n > d \tag{2.13}$$

One visualization of Equation (2.1) is the use of $A$ to map vectors from $R^d$ to $R^n$, as shown in Figure 2.1. However, Equation (2.10) has a particularly important connotation when combined with Equations (2.4) and (2.8), yielding

$$\underline{y} = \sum_{k=1}^{d} \underline{a}_i x_i \tag{2.14}$$

$\underline{y}$ is formed as a linear combination of the columns of $A$. The set of all linear combinations (i.e., using any $\underline{x}$) of the columns of $A$ is the *range of A*, denoted $R(A)$. Clearly, in

---

[2]Sometimes referred to as *linear in the input/output sense* to distinguish it from other connotations of linearity.

[3]Written in MATLAB simply as `y=A*x`

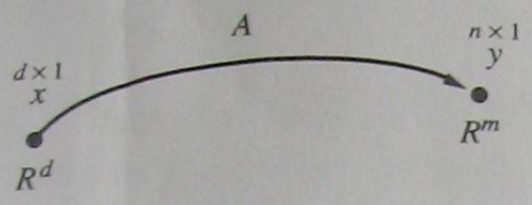



**FIGURE 2.1**
Matrices as vector space mappings.

matrix equations of the form

$$A\underline{x} = \underline{y} \tag{2.15}$$

if $\underline{y} \notin R(A)$, trying to find an $\underline{x}$ that satisfies Equation (2.15) *exactly* is futile.

Another viewpoint of Equation (2.10) when $n = d$ is that of a change of coordinate system (or basis vectors). $\underline{x}$ is the representation of a point in $R^d$ with respect to the original coordinate system, $\underline{y}$ is the representation with respect to a new coordinate system, and $A$ relates the original and new systems.

### Matrix rank

Equation (2.4) facilitates a discussion of the property of matrix rank. With $A$ viewed in "column form" as in Equation (2.4), the rank of $A$, denoted rank($A$), is defined as the number of linearly independent columns of $A$. For $\overset{n\times d}{A}$ and $n \geq d$, this number is at most $d$. A square matrix with full rank [i.e., rank($A$) $= n = d$] is invertible,[4] and its columns provide a basis set for $R^n = R^d$. Any vector in $R^n = R^d$ may be represented as a linear combination of a basis set.

## 2.1.4 Inner and Outer Products and Applications

### Inner products

If $\underline{x}$ and $\underline{y}$ are real $d \times 1$ vectors, their vector inner product is denoted using braces ($\langle\,\rangle$) and defined to be the scalar given by $\langle \underline{x}, \underline{y} \rangle = (\underline{x})^T \underline{y} = \underline{y}^T \underline{x}$. The inner product, since it is a scalar, is symmetric. Geometrically, $\langle \underline{x}, \underline{y} \rangle$ is visualized as the projection of $\underline{y}$ onto $\underline{x}$ (or vice versa), as shown in Figure 2.2. The inner product provides a measure of the closeness[5] of two vectors.

Notice

$$\langle \underline{x}, \underline{x} \rangle = \underline{x}^T \underline{x} = \sum_{k=1}^{d} x_k^2 \tag{2.16}$$

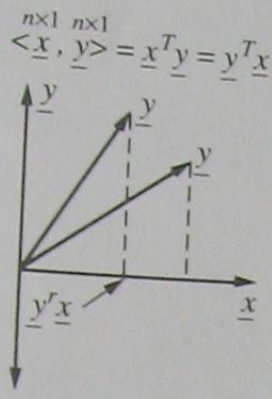
so the (Euclidean) length or *norm* of vector $\underline{x}$, denoted $\|\underline{x}\|$, is

$$\|\underline{x}\| = \sqrt{\sum_{k=1}^{d} x_k^2} = [\langle \underline{x}, \underline{x} \rangle]^{1/2} \tag{2.17}$$

---

[4]Invertibility is defined in Section 2.1.9.

[5]This must be further qualified; e.g., relative vector lengths must be considered before we use it as a measure of match.




Projection of $\underline{y}$ onto $\underline{x}$ or $\underline{x}$ onto $\underline{y}$

**FIGURE 2.2**
Inner product visualization as a projection.

or

$$\langle \underline{x}, \underline{x} \rangle = \|\underline{x}\|^2 \tag{2.18}$$

Another commonly encountered form is the inner product with respect to a matrix, i.e.,

$$\langle \underline{x}, R\underline{x} \rangle = \underline{x}^T R\underline{x} = \|\underline{x}\|_R^2 \tag{2.19}$$

together with

$$\langle \underline{x}, R\underline{y} \rangle = \underline{x}^T R\underline{y} = \underline{y}^T R\underline{x} \tag{2.20}$$

Often Equation (2.19) is used to denote a non-Euclidean vector norm (squared). It is also a quadratic form (to be discussed). When $R = I$, the Euclidean norm results. Refer to Section 2.1.9 for an example application.

A particularly useful form of Equation (2.19) involves the difference (vector) of two vectors and provides a scalar measure of the closeness of $\underline{x}$ and $\underline{y}$, i.e.,

$$\|\underline{x} - \underline{y}\|_R = \langle (\underline{x} - \underline{y}), R(\underline{x} - \underline{y}) \rangle = (\underline{x} - \underline{y})^T R(\underline{x} - \underline{y}) \tag{2.21}$$

$$= \|\underline{x}\|_R^2 + \|\underline{y}\|_R^2 - (\langle y, R\underline{x} \rangle + \langle \underline{x}, R\underline{y} \rangle) \tag{2.22}$$

Simplification of Equation (2.22) is possible for symmetric $R$, yielding

$$\|\underline{x} - \underline{y}\|_R = \langle (\underline{x} - \underline{y}), R(\underline{x} - \underline{y}) \rangle = \|\underline{x}\|_R^2 + \|y\|_R^2 - 2\langle x, R\underline{y} \rangle \tag{2.23}$$

Finally, the inner product is linear:

$$\langle \alpha \underline{x}_1 + \beta \underline{x}_2, \underline{y} \rangle = \alpha \langle \underline{x}_1, \underline{y} \rangle + \beta \langle \underline{x}_2, \underline{y} \rangle \tag{2.24}$$

If $\langle \underline{x}, \underline{y} \rangle = 0$, vectors $\underline{x}$ and $\underline{y}$ are said to be *orthogonal*.

A general vector $p$ norm[6] may be expressed as:

$$\|\underline{x}\|_p = (|x_1|^p + |x_2|^p + |x_d|^p)^{1/p} \tag{2.25}$$

Two particularly useful cases are

- $p = 2$, considered previously
- $p = \infty$, resulting in the *maximum norm:*

---

[6]Unfortunately, there is some risk of confusion between $p$ norms and $R$ norms (Section 2.1.9).

$$\|\underline{x}\|_\infty = \max_{1 \le i \le d} |x_i| \tag{2.26}$$

Vector norms must satisfy the following constraints, which are analogous to the usual notion of length:

1. $\|\underline{x}\| > 0$ if $\underline{x} \neq 0$
2. $\|\alpha \underline{x}\| = \alpha \|\underline{x}\|$
3. $\|\underline{x} + \underline{y}\| \le \|\underline{x}\| + \|\underline{y}\|$

### Outer products

The outer product of $\underline{x}$ and $\underline{y}$, denoted $\rangle \underline{x}, \underline{y} \langle$, is the rank 1 matrix $\underline{x}\underline{y}^T$. In contrast to the inner product, $\underline{x}$ and $\underline{y}$ may have unequal dimensions. Thus,

$$\rangle {}^{n\times 1}_{\ }\underline{x}, {}^{m\times 1}_{\ }\underline{y} \langle = \underline{x}\underline{y}^T = {}^{n\times m}_{\ }P \tag{2.27}$$

and $P$ is generally nonsquare. Expanding Equation (2.27), where $\underline{x} = (x_1, x_2, \ldots, x_n)^T$ and $\underline{y} = (y_1, y_2, \ldots, y_m)^T$, yields

$$\underline{x}\underline{y}^T = P = \begin{pmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_m \\ x_2y_1 & x_2y_2 & \cdots & x_2y_m \\ \vdots & \vdots & & \vdots \\ x_ny_1 & x_ny_2 & \cdots & x_ny_m \end{pmatrix} \tag{2.28}$$

When square, the outer product matrix $P$ is not necessarily symmetric.

## 2.1.5 Measures of Similarity in Vector Space

Distance is one measure of vector similarity. The Euclidean distance between vectors $\underline{x}$ and $\underline{y}$ is given by

$$d(\underline{x}, \underline{y}) = \|\underline{x} - \underline{y}\| = \sqrt{(\underline{x} - \underline{y})^T (\underline{x} - \underline{y})} \tag{2.29}$$

$$= + \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \tag{2.30}$$

A related and more general metric is

$$d_p(\underline{x}, \underline{y}) = \left( \sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p} \tag{2.31}$$

Equation (2.31) reduces to Equation (2.30) for $p = 2$.

Often, weighted distance measures are used. An example is

$$d_w^2(\underline{x}, \underline{y}) = (\underline{x} - \underline{y})^T R (\underline{x} - \underline{y}) = \|\underline{x} - \underline{y}\|_R^2 \tag{2.32}$$

Equation (2.32) implements on a *weighted inner product* or weighted $R$ norm. The matrix $R$ is often required to be positive definite and symmetric. In this case, $R$ may be factored. Equation (2.32) represents the transformation of a vector space; i.e., the linear transformations

$$\underline{\tilde{x}} = T\underline{x} \tag{2.33}$$

$$\underline{\tilde{y}} = T\underline{y} \tag{2.34}$$

yield

$$d^2(\underline{\tilde{x}}, \underline{\tilde{y}}) = (T\underline{x} - T\underline{y})^T(T\underline{x} - T\underline{y}) \tag{2.35}$$

$$= (\underline{x} - \underline{y})^T T^T T(\underline{x} - \underline{y}) \tag{2.36}$$

$$= d_w^2(\underline{x}, \underline{y}) \tag{2.37}$$

When $\underline{x}$ and $\underline{y}$ are binary, measures such as the Hamming distance (Chapter 8) are useful.

### Correlation and matching

Correlation is a simple and extremely popular matching approach that is applicable to signals, vectors, strings, and sets. A set of reference patterns, called *templates*, are used together with an unknown pattern. The unknown pattern is achieved by shifting the template over all possible relative locations and, using a suitable matching metric, computing a *correlation function*. This leads to more generalized approaches, including matched filtering.

The following designations are used:

$g$: The (input) pattern. For example, this may simply be a $d \times 1$ vector $\underline{g}$.
$f$: The reference pattern or template for a particular class. For example, this may be a $d \times 1$ mean vector $\underline{f} = \mu_i$ corresponding to class $w_i$.
$R_m$: The extent of $g$ over which the match occurs. In some applications, this includes all of $g$, e.g., all $d$ components of vector $\underline{g}$. However, the match may also be over a smaller extent, for example, finding a subpattern in a larger pattern.

Consider the discrete formulation of the following two candidate metrics indicating *mismatch* (indices are omitted for simplicity):

$$m_1 = \sum_{R_m} |f - g| \tag{2.38}$$

$$m_2 = \sum_{R_m} (f - g)^2 \tag{2.39}$$

Intuitively, $m_1$ and $m_2$ will be small (ideally zero) when $f$ and $g$ are identical, and large when they are significantly different. Whereas the first metric is easy to compute, a closer examination of Equation (2.39) leads us to some interesting results. Expanding the second-order term in Equation (2.39) yields

$$m_2 = \sum f^2 - 2\sum fg + \sum g^2 \tag{2.40}$$

In the vector case, where

$$\sum f g = \langle \underline{f}, \underline{g} \rangle \tag{2.41}$$

since the coefficient of this term in Equation (2.40) is negative, when this term is large $m_2$ will be small. Therefore, $m_2$ provides a good measure of *mismatch*, and $\sum f g$ provides a reasonable measure of match. This operation is denoted the *nonnormalized correlation* of $f$ and $g$ (over $R$) and amounts to an element-by-element multiplication followed by a summation.

**Matrix norms**

Various matrix norms exist [Dah74]. One of the most useful is the matrix-bound norm

$$\|A\| = \max_{\underline{x} \neq \underline{0}} \frac{\|A\underline{x}\|}{\|\underline{x}\|} \tag{2.42}$$

which has the property

$$\|A\underline{x}\| \leq \|A\| \, \|\underline{x}\| \tag{2.43}$$

where $\|\underline{x}\|$ is a vector norm.

## 2.1.6 Differentiation of Matrices and Vectors

Let $f(\underline{x})$ be a scalar-valued function of $n$ variables $x_i$, written as an $n \times 1$ vector $\underline{x}$. The derivative of $f(\underline{x})$ with respect to $\underline{x}$ is an $n \times 1$ vector defined as

$$\frac{df(\underline{x})}{d\underline{x}} = \begin{pmatrix} \frac{\partial f(\underline{x})}{\partial x_1} \\ \frac{\partial f(\underline{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\underline{x})}{\partial x_n} \end{pmatrix} \tag{2.44}$$

Equation (2.44) defines the gradient (vector) of $f$, denoted as $\nabla_x \underline{f}$ or $\text{grad}_x \underline{f}$, which is *the direction of maximum increase of the function f*.

The differentiation of a vector function, i.e., $\underline{f}(\underline{x})$ where $\underline{f}$ is $m \times 1$ and $\underline{x}$ is $n \times 1$, results in a $m \times n$ matrix of the form

$$\frac{d\underline{f}(\underline{x})}{d\underline{x}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ & \ddots & \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} \tag{2.45}$$

where the $ij$th element of this matrix is $\partial f_i / \partial x_j$, $f_i$ is the $i$th element of $\underline{f}$, and $x_j$ is the $j$th element of $\underline{x}$. This matrix is the Jacobian of $\underline{f}(\underline{x})$, denoted $J_{\underline{x}}$. The differentiation of

a matrix with respect to a vector requires a three-dimensional representation and thus employs tensor notation.

Examples of properties using the preceding definitions may be easily derived and are summarized here. For a matrix $A$ and vectors $\underline{x}$ and $\underline{y}$,

$$\frac{d}{d\underline{x}}(A\underline{x}) = A \tag{2.46}$$

$$\frac{d}{d\underline{x}}(\underline{y}^T A\underline{x}) = A^T\underline{y} \tag{2.47}$$

$$\frac{d}{d\underline{x}}(\underline{x}^T A\underline{x}) = (A + A^T)\underline{x} \tag{2.48}$$

### 2.1.7 The Chain Rule

To rigorously derive one of the feedforward network training algorithms, we need to consider the *chain rule* and composite (error) functions. Observe the following:

- A differentiable function of a differentiable function is itself differentiable.
- If $\varsigma = \phi(x, y, \ldots)$, $\eta = \phi(x, y, \ldots), \ldots$ are differentiable functions of $x, y, \ldots$ and $f(\varsigma, \eta, \ldots)$ is a differentiable function of $\varsigma, \eta, \ldots$, then $f(\phi(x, y, \ldots), \psi(x, y, \ldots), \ldots)$ is a differentiable function of $x, y, \ldots$,[7] with partial derivatives given by

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial \phi}\frac{\partial \phi}{\partial x} + \frac{\partial f}{\partial \psi}\frac{\partial \psi}{\partial x} + \cdots \tag{2.49}$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial \phi}\frac{\partial \phi}{\partial y} + \frac{\partial f}{\partial \psi}\frac{\partial \psi}{\partial y} + \cdots \tag{2.50}$$

$$\vdots \tag{2.51}$$

This result is independent of the number of independent variables $x, y, \ldots$.

### 2.1.8 Multidimensional Taylor Series Expansions

The Taylor series expansion for a scalar function of a vector variable $f(\underline{x})$ about point $\underline{x}_o$ is written, using the results of the previous section, as

$$f(\underline{x}) = f(\underline{x}_o) + \left[\frac{df(\underline{x}_o)}{d\underline{x}}\right]^T (\underline{x} - \underline{x}_o) + \frac{1}{2}(\underline{x} - \underline{x}_o)^T \left[\frac{d^2 f(\underline{x}_o)}{d\underline{x}^2}\right](\underline{x} - \underline{x}_o) + \text{higher-order terms} \tag{2.52}$$

[7] Specification of the region $R$ over which this holds is also necessary.

Expanding Equation (2.52) to get $\underline{x} + \Delta\underline{x}$ yields

$$f(\underline{x} + \Delta\underline{x}) \approx f(\underline{x}) + \frac{df(\underline{x})^T}{d\underline{x}} \Delta\underline{x} \tag{2.53}$$

Noting the inner product operation in Equation (2.53), we see why the gradient is the direction of maximum increase in $\underline{f}(\ )$.

Similarly, a vector function expansion is

$$\underline{f}(\underline{x}) = f(\underline{x}_o) + \left[\frac{d\underline{f}(\underline{x}_o)}{d\underline{x}}\right](\underline{x} - \underline{x}_o) + \text{higher-order terms} \tag{2.54}$$

### 2.1.9 The Pseudoinverse of a Matrix and Least Squares Techniques (Deterministic)

Referring to the three quantities involved in Equation (2.10),

$$\overset{n\times m}{A}\ \overset{m\times 1}{\underline{x}} = \overset{n\times 1}{\underline{y}} \tag{2.55}$$

our previous viewpoint was in "producing" $\underline{y}$, given $A$ and $\underline{x}$. Consider now two alternative viewpoints:

1. Given $A$ and $y$, can an $\underline{x}$ be found that satisfies this equation? If not, can we come close? How close?
2. Given $\underline{x}$ and $\underline{y}$ [or perhaps several pairs $(\underline{x}_i, y_i)$], can an $A$ be found that produces this mapping?

Both cases are of significant interest in ANN design and analysis. For example, linear pattern associators may be trained using least squares approaches (if the training data yield an overdetermined case). Alternatively, minimum-length solution vectors, also given by a special form of the pseudoinverse, are often sought.

Square matrices with full rank have a unique inverse. When the rank is less than the dimension, or when the matrix is not square, a number of possible cases may arise. The problem of inverting an arbitrary rectangular matrix $A$ has been studied for some time [RM71]. The *pseudoinverse* of an $m \times n$ real matrix, $A$, is an $n \times m$ matrix denoted by $A^\dagger$. Desirable properties of $A^\dagger$ are

$$AA^\dagger A = A \tag{2.56}$$

$$A^\dagger AA^\dagger = A^\dagger \tag{2.57}$$

$$(AA^\dagger)^T = AA^\dagger \tag{2.58}$$

If $A$ has full column rank, an inverse of considerable interest is the *least squares* inverse, denoted by

$$A^\dagger = (A^T A)^{-1} A^T \tag{2.59}$$

**Algebraic least squares formulation**

Given an overdetermined linear equation of the form

$$\underline{y} = A\underline{x} \tag{2.60}$$

where $\underline{y}$ is $m \times 1$, $\underline{x}$ is $n \times 1$, $m > n$, and $A$ is an $m \times n$ matrix of rank $n$, there clearly is no way to exactly satisfy this equation for arbitrary $\underline{y}$. We define an $m \times 1$ error vector corresponding to some approximate solution, $\hat{\underline{x}}$ as

$$\underline{e} = \underline{y} - A\hat{\underline{x}} \tag{2.61}$$

and then determine a procedure to minimize some function of this error. Often, in unweighted least squares, this function, denoted $J$, is chosen to be

$$J = \underline{e}^T\underline{e} \tag{2.62}$$

To find the minimum of this function, we set

$$\frac{dJ}{d\underline{x}} = \underline{0} \tag{2.63}$$

and use Equation (2.44) to develop the so-called normal equations, i.e.,

$$A^T A\hat{\underline{x}} = A^T\underline{y} \tag{2.64}$$

from which $\underline{y}$ may be determined. Note that, in theory, $A^T A$ may be inverted to yield $\hat{\underline{x}}$.

**The geometrical approach (overdetermined system)**

The modern approach to Equation (2.60) proceeds from a geometric view of vector-matrix relationships in $m$- and $n$-dimensional spaces. For example, $A\underline{x} = \underline{y}$ may be thought of as a way to map the $n$-dimensional vector $\underline{x}$ to the $m$-dimensional vector $\underline{y}$. The problem concerns inverting this mapping.

In the overdetermined case, $\underline{y}$ does not lie in the column space of $A$ (the *range of A*), or $R(A)$. Thus, we desire a solution $\underline{x}$ such that the orthogonal distance between $A\underline{x}$ and $\underline{y}$ is minimum. As in the previous approach, we characterize this distance as the length of the error vector $\underline{e}$, defined in Equation (2.61). This situation is shown in Figure 2.3.

The geometric approach makes use of the fact that the length of $\underline{e}$ is minimum when $\underline{e}$ lies in a vector space orthogonal to $R(A)$. This space is known as the *null space of* $A^T$, denoted $N(A^T)$. Any vector $\underline{\zeta}$ in this null space is characterized by

$$A^T\underline{\zeta} = \underline{0} \tag{2.65}$$

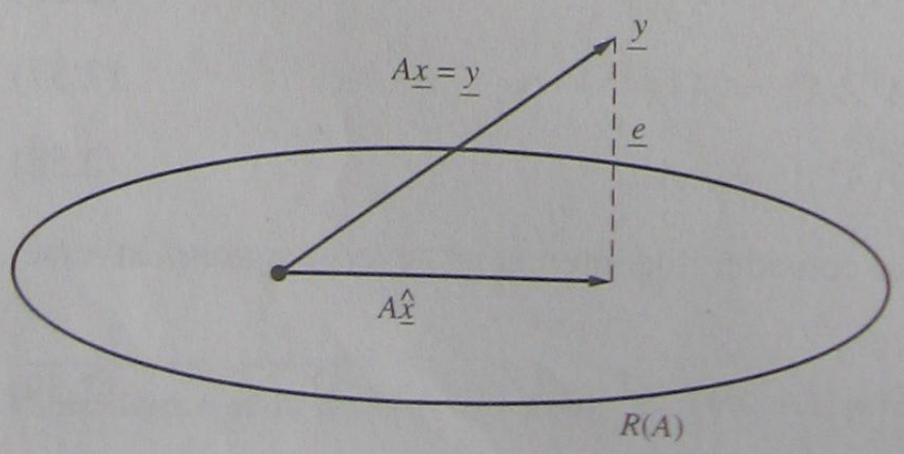



**FIGURE 2.3**
Geometric viewpoint of least squares.

The geometric approach thus constrains $\underline{e}$ to satisfy

$$A^T \underline{e} = \underline{0} \tag{2.66}$$

from which the so-called normal equations, identical to the previous solution given by Equation (2.64), arise.

The geometric solution may also be characterized as finding the *projection* of $\underline{y}$ onto $R(A)$ (call this $\underline{y}^P$) and then finding the vector $\underline{\bar{x}}$ that exactly satisfies $A\underline{\bar{x}} = \underline{y}^P$. This approach also yields the normal equations, as shown previously. Once the normal equations have been formed, if $(A^T A)$ is invertible, we may form the solution for $\underline{\bar{x}}$ as

$$\underline{\hat{x}} = (A^T A)^{-1} A^T \underline{y} \tag{2.67}$$

The quantity $(A^T A)^{-1} A^T$ is known as the pseudoinverse of $A$.

**Practical concerns regarding least squares solutions (QR decomposition)**

Although the quantity $(A^T A)^{-1}$ exists in theory,[8] there may be significant numerical difficulties in computing this inverse in cases where $A^T A$ is nearly singular. In addition, the process of forming $A^T A$ for large $A$ is computationally expensive. The numerical errors incurred in forming $A^T A$ and then forming the inverse spawn a need for alternative approaches not plagued by numerical sensitivities. One solution is to avoid forming the normal equations explicitly but still follow the geometric approach. This solution is known as QR decomposition [DA74].

***The underdetermined case.*** Although situations yielding the least squares formulation for overdetermined systems are quite prevalent, another situation that arises from Equation (2.60) is where multiple solutions for $x$ exist. In other words, $\underline{y} \in R(A)$, but the columns of $A$ are not linearly independent. Nonzero vectors from $N(A)$ may be added to any solution for $\underline{x}$ to yield different, but valid, solutions. Consequently, the pseudoinverse must be defined for this case. Perhaps the easiest constraint to add is that the optimal solution, among all solutions to

$$A\underline{\hat{x}} = \underline{y}^P \tag{2.68}$$

is the one with minimum length. The row space of $A$ and the null space of $A$ $[N(A)]$ are orthogonal complements. The addition of any vector in $N(A)$ increases the length of $\underline{\bar{x}}$; therefore, the null space component of the solution must be zero, and $\underline{\bar{x}}$ must lie in the row space of $A$. When the row space of $A$ has full rank,[9] the pseudoinverse becomes

$$\underline{\hat{x}} = (AA^T)^{-1} A\underline{y} \tag{2.69}$$

Equation (2.69) is another commonly encountered form.

***A unifying approach: Singular value decomposition.*** [Str76] presents this derivation in a particularly lucid way. Here we enforce both of the preceding constraints:

- $A\underline{\hat{x}} = \underline{y}^P$
- $\underline{\hat{x}}$ lies in the row space of $A$.

---

[8] Since we required that the rank of $A = n$.

[9] This situation occurs frequently.

The solution is based on factoring the $m \times n$ matrix $A$ into

$$A = Q_1 \Sigma Q_2^T \tag{2.70}$$

where $Q_1$ is an $m \times m$ orthogonal matrix, $Q_2$ is an $n \times n$ orthogonal matrix, and $\Sigma$ is an $m \times n$ diagonal matrix with the first $r$ diagonal entries, $\mu_1, \ldots, \mu_r$, nonzero. The $\mu_i$ are called the singular values of $A$. In general,

$$A^\dagger = Q_2 \Sigma^\dagger Q_1^T \tag{2.71}$$

where $\Sigma^\dagger$ is $n \times m$ with diagonal entries $\mu_1^{-1}, \mu_2^{-1}, \ldots, \mu_r^{-1}$. Numerical routines to compute the singular value decomposition exist [DA74].

### Weighted least squares

For many reasons, minimization of residuals based on $\underline{e}^T \underline{e}$, or $\|\underline{e}\|^2$, as used in Equation (2.62), is not advisable. In that formulation all elements of $\underline{e}$ are squared and added. This does not allow giving special attention to some errors vis-á-vis others. Using non-Euclidean error norms is a way to overcome this and leads to weighted least squares (WLS).

The WLS formulation begins by redefining Equation (2.62):

$$J = \underline{e}^T R \underline{e} = \|\underline{e}\|_R^2 \tag{2.72}$$

which represents an error measure based on an $R$ norm. In most reasonable solutions, must be positive definite; it is usually symmetric and often diagonal. Therefore, anoth way to view Equation (2.62) is as Equation (2.72) with $R = I$. The reader is encourag to repeat the steps of that section to arrive at the modified normal equations:

$$A^T R A \hat{\underline{x}} = A^T R \underline{y} \tag{2.7}$$

## 2.1.10 Eigenvalues and Eigenvectors

### Formulation

Consider an $n \times n$ (square) matrix $A$. A scalar $\lambda$ and a vector $\hat{\underline{x}}$[10] are sought s that

$$A \hat{\underline{x}} = \lambda \hat{\underline{x}} \tag{2}$$

i.e., $A\hat{\underline{x}}$ has the same direction in $R^n$ as $\hat{\underline{x}}$ and is scaled by a factor of $\lambda$. $\hat{\underline{x}} = \underline{0}$ is a tr solution. Equation (2.74) may be rewritten

$$(A - \lambda I)\hat{\underline{x}} = \underline{0} \tag{}$$

$(A - \lambda I)^{-1}$ exists only if $|A - \lambda I| \neq 0$; therefore, for "interesting" (nontrivial) solu for $\hat{\underline{x}}$, we require $|A - \lambda I| = 0$. For an $n \times n$ matrix, this yields a scalar polynomi $\lambda$ of order $n$ of the form

$$|A - \lambda I| = \lambda^n + a_1 \lambda^{n-1} + \cdots + a_{n-1}\lambda + a_n = 0$$

[10] This $\hat{\underline{x}}$ should not be confused with the least squares estimator considered previously.

Equation (2.76) is called the *characteristic equation* or *characteristic polynomial of the matrix A*, often denoted as $P(\lambda)$. There are $n$ solutions for the $\lambda$ constrained by Equation (2.76) (although they are not all necessarily unique and are possibly complex). These $n$ solutions for $\lambda$ are termed the *eigenvalues* or *e-values* of $A$, and the corresponding vectors $\underline{\hat{x}}$ are the *eigenvectors* or *e-vectors*. Only under certain conditions do we get $n$ linearly independent eigenvectors. The *direction* of the e-vector is specified, because Equation (2.75) is homogeneous.

One important result is that the summation of the eigenvalues of a matrix equals the trace[11] of the matrix. This is an especially useful property in the analysis of recurrent networks using the Hopfield storage prescription.

### The modal matrix

The matrix formed by the column vectors $\underline{\hat{x}}_i$ (the e-vectors of $A$) is called the *modal* matrix, denoted $M$:

$$M = [\underline{\hat{x}}_i] \quad i = 1, 2, \ldots, n \tag{2.77}$$

For the case of nonrepeated $\lambda_i$, $M$ will have $n$ *linearly independent* eigenvectors.[12] Thus, any matrix with distinct e-values yields a model matrix $M$ that is invertible. Recall that $M$ is not unique because of a possible scaling of the $\underline{\hat{x}}_i$. Given $n$ solutions for $\lambda_i$ and $\underline{\hat{x}}_i$ in the equation $A\underline{\hat{x}}_i = \lambda_i \underline{\hat{x}}_i$, these equations may be written as

$$AM = M\Lambda \tag{2.78}$$

where

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \ldots & 0 & 0 \\ 0 & \lambda_2 & \ldots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \ldots & 0 & \lambda_n \end{pmatrix} \tag{2.79}$$

Since $M$ is invertible,

$$A = M\Lambda M^{-1} \tag{2.80}$$

or

$$\Lambda = M^{-1}AM \tag{2.81}$$

Equations (2.80) and (2.81) provide a means for diagonalization of the matrix $A$ or the representation of $A$ with respect to its eigenvectors. This is a *coordinate transformation.*

### Application to symmetric matrices

The uniqueness of the $\lambda_i$ and the existence of a set of linearly independent e-vectors is guaranteed in the case of (real) symmetric matrices. A matrix satisfying $A = A^T$ has the following properties:

---

[11] The sum of the diagonal elements.

[12] Note that matrices with repeated e-values also may have a linearly independent set of e-vectors, but this is not guaranteed. The identity matrix, $I$, is a good example.

1. The matrix has real e-values and a diagonal $\Lambda$.
2. The e-vectors compose a set of *orthogonal* vectors. Thus, if the e-vectors are normalized such that they are orthonormal (have unity length),

$$M^{-1} = M^T \tag{2.82}$$

## 2.2 GEOMETRY FOR STATE-SPACE VISUALIZATION

In this section we present several results useful for analyzing decision regions and visualizing feature vectors $R^d$.

### 2.2.1 Geometric Interpretation of ANN Mappings

It is often useful to develop a geometric viewpoint of ANN input/output mappings. ANN inputs are arranged in a $d$-dimensional vector, denoted $\underline{x}$, which yields a multidimensional *input space*. If each input is an unconstrained real number, the input space is $R^d$. In other cases it is convenient to restrict the input space to a subspace of $R^d$. Often a desired mapping partitions the input space into geometrically distinguishable regions.

### 2.2.2 Hypercubes

The outputs of many ANN units are "squashed" into discrete values, e.g., $\{-1, 1\}$ (which we hereafter refer to as *bipolar*) or $\{0, 1\}$ (denoted *binary*). Alternatively, continuous outputs may lie in the intervals $[-1, 1]$, $[0, 1]$ or $(-1, 1)$, $(0, 1)$. For a set of $d$ units, the output of the units may be collectively shown as a vector in $R^d$. This allows visualization of the unit outputs as either points on $d$-dimensional hypercubes or as points in the interior of such hypercubes.

For example, if individual neuron outputs and network inputs are restricted to the range $[0, 1]$, for a $d$-dimensional input vector, we have an input space that is a unit volume hypercube in $R^d$. This is denoted by the *unit cube:*

$$[0, 1]^d = \{\underline{x} = (x_1, x_2, \ldots, x_d) \in R^d | 0 \leq x_i \leq 1 \; \forall i = 1, d\} \tag{2.83}$$

Discrete binary cubes are defined analogously. Suppose

$$\underline{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} \tag{2.84}$$

where $x_i \in \{0, 1\}$. For example, vector $\underline{x}$ could represent the state of a $d$-unit neural network. A convenient visualization of this vector in $I^d$ is as the "corner" or vertex of a $d$-dimensional cube. Other vectors that differ from $\underline{x}$ by a Hamming distance (HD) of 1 are directly connected to $\underline{x}$ via a vertex, whereas those that differ from $\underline{x}$ by HD > 1

require a traversal in $I^d$ of at least one other vertex. Formally, the discrete unit binary cube is defined by

$$\{0, 1\}^d = \{\underline{x} = (x_1, x_2, \ldots, x_d) \in R^d | x_i \in \{0, 1\} \, \forall i = 1, d\} \tag{2.85}$$

Finally, the bipolar equivalents to the cubes defined in Equations (2.83) and (2.85) are

$$[-1, 1]^d = \{\underline{x} = (x_1, x_2, \ldots, x_d) \in R^d | -1 \leq x_i \leq 1 \, \forall i = 1, d\} \tag{2.86}$$

and

$$\{-1, 1\}^d = \{\underline{x} = (x_1, x_2, \ldots, x_d) \in R^d | x_i \in \{-1, 1\} \, \forall i = 1, d\} \tag{2.87}$$

### 2.2.3 ANN Mappings, Decision Regions and Boundaries, and Discriminant Functions

#### Discriminant functions

In the $c$-class mapping case, discriminant functions, denoted $g_i(\underline{x})$ $(i = 1, 2, \ldots, c)$ are used to partition $R^d$ as follows.

*Decision rule:* Map $\underline{x}$ to decision or output $w_m$ (region $R_m$), where $g_m(\underline{x}) > g_i(\underline{x}) \; \forall i = 1, 2, \ldots, c$ and $i \neq m$. Note that the case where $g_k(\underline{x}) = g_l(\underline{x})$ defines a decision boundary.

A particularly important discriminant function form is the *linear discriminant function*

$$g_i(\underline{x}) = \underline{w}_i^T \underline{x} + w_{oi} \tag{2.88}$$

where $\underline{w}_i$ is a $d \times 1$ vector of weights used for class $i$. This function yields decision boundaries that are hyperplanes.

#### Decision regions

The concept of *decision regions* is familiar in pattern recognition (PR). This concept has significant utility in the analysis of ANN mappings. A *classifier* partitions input space into decision regions. In order to use decision regions for a possible and unique mapping, these regions must cover $R^d$ and be disjoint (nonoverlapping). An exception to the latter constraint is the notion of fuzzy sets [Zad75]. The border of each decision region is a *decision boundary*.

The determination of decision regions is a challenge. It is sometimes convenient, yet not always necessary (or possible), to visualize decision regions and boundaries. Moreover, computational and geometric aspects of certain decision boundaries (e.g., linear classifiers that generate hyperplanar decision boundaries) are noteworthy.

#### Hyperplanes

The general equation of a plane in $d$ dimensions is

$$\langle \underline{w}, \underline{x} \rangle = k \tag{2.89}$$

where $\underline{x}$ is a $d \times 1$ vector, $\underline{w}$ is the normal to the hyperplane, and $k$ is a (scalar) constant. Equation (2.89) may alternatively be viewed as a constraint on the locus of all vectors $\underline{x}$

in $R^d$. A general result (proved below) is that the minimum distance to the origin from any point on the plane is $d_{\min} = |k|/\|\underline{w}\|$, and the point is given by $\underline{x}_{\min} = k\underline{w}/\|\underline{w}\|^2$.

***3-D case: Through origin.*** In 3-D or $(x_1, x_2, x_3)$ space, *a plane through the origin*[13] is determined by three parameters $(w_1, w_2, w_3)$ via

$$(w_1\ w_2\ w_3)\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0 \tag{2.90}$$

or simply

$$\underline{w}^T \underline{x} = 0 \tag{2.91}$$

If $\underline{x}$ represents the position vector of a point $X$ in the plane measured with respect to the (assumed Cartesian) coordinate system origin, then Equation (2.91) indicates that the plane parameter vector $\underline{w}$ and $\underline{x}$ are *orthogonal*. Parameter vector $\underline{w}$ is therefore the normal to the plane, but since Equation (2.91) is homogeneous, only the direction of $\underline{w}$ is constrained.

***3-D case: Not necessarily through origin.*** A plane through any other point $X_o$, represented by position vector $\underline{x}_o$, may be written as

$$\underline{w}^T(\underline{x} - \underline{x}_o) = 0 \tag{2.92}$$

or

$$\underline{w}^T \underline{x} - d = 0 \tag{2.93}$$

where $d = \underline{w}^T \underline{x}_o$. This is shown in Figure 2.4 for a 2-D case.

The reformulation of Equation (2.90) into Equation (2.92) is equivalent to a coordinate system transformation where we shift the origin to $\underline{x}_o$ and therefore measure vectors $\underline{x}' = \underline{x} - \underline{x}_o$, where $\underline{x}'$ and $\underline{x}$ are the shifted and unshifted coordinate locations, respectively. Notice from Equation (2.93) that $\underline{w}$ is orthogonal to any vector $\underline{x} - \underline{x}_o$

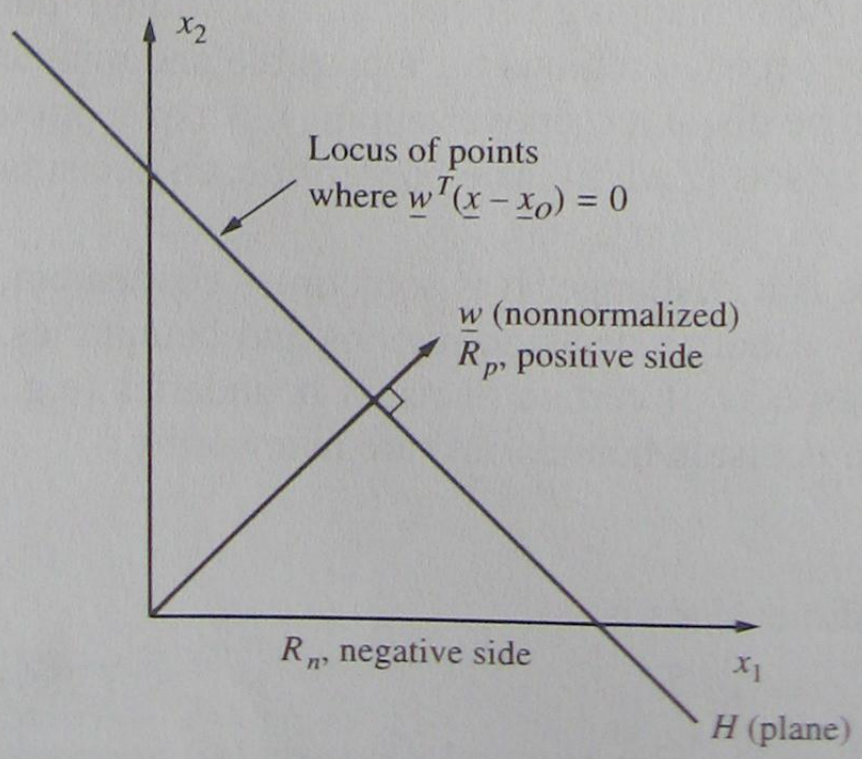



**FIGURE 2.4**
2-D "plane" representation.

[13] Or that contains the origin.

lying in the plane. The plane determined by Equation (2.93) is denoted $H$ in Figure 2.4. Interestingly, Equation (2.93) may be used to determine the "distance" of $H$ from the origin and, for a given $\underline{x}$, the "side" of $H$ in $R^3$ that $\underline{x}$ is on, as well as the distance of $\underline{x}$ from $H$. $\underline{w}$ is normalized to unit length by forming $\underline{w}' = \underline{w}/\|\underline{w}\|$. Vector $\underline{x}_d$ may be written as

$$\underline{x}_d = \alpha \underline{w}' \tag{2.94}$$

so $\|\underline{x}_d\| = |\alpha|$. Using Equation (2.92), since $\underline{x}_o$ represents a point on $H$,

$$\underline{w}^T(\underline{x}_d - \underline{x}_o) = \underline{w}^T\left(\alpha \frac{\underline{w}}{\|\underline{w}\|} - \underline{x}_o\right) = 0 \tag{2.95}$$

or

$$\alpha \frac{\underline{w}^T \underline{w}}{\|\underline{w}\|} - d = 0 \tag{2.96}$$

yielding

$$\alpha \|\underline{w}\| = d \tag{2.97}$$

or

$$\alpha = \frac{d}{\|\underline{w}\|} \tag{2.98}$$

Therefore, the distance from the origin to $H$ is given by $|\alpha| = |d|/\|\underline{w}\| = |\underline{w}^T \underline{x}_o|/\|\underline{w}\|$.

***Extension to linear decision region boundaries.*** Plane $H$, characterized by Equation (2.92), *partitions* $R^3$ (in general, $R^d$, as described below) into two mutually exclusive regions, denoted $R_p$ and $R_n$ in Figure 2.4. The assignment of vector $\underline{x}$ to either the "positive" side, the "negative" side, or along $H$ can be implemented by

$$\underline{w}^T \underline{x} - d \begin{cases} > 0 & \text{if } \underline{x} \in R_p \\ = 0 & \text{if } \underline{x} \in H \\ < 0 & \text{if } \underline{x} \in R_n \end{cases} \tag{2.99}$$

This suggests a *linear discriminant function*, $g(\underline{x})$ to implement Equation (2.99):

$$g(\underline{x}) = \underline{w}^T \underline{x} - d \tag{2.100}$$

Although we have considered only $R^3$ (and $R^2$) in the previous analysis, the results are easily extendible to $R^d$ by simply choosing $d > 3$. This allows linear classification of $d$-dimensional feature vectors, $\underline{x}$. Visualization, however, is more difficult. The surface $H$ in this context is referred to as a *hyperplane*.

### Quadratic forms

From the definition of inner products, for a $d \times 1$ vector $\underline{x} = (x_1, x_2, \ldots, x_d)^T$, the scalar

$$Q = \langle \underline{x}, R\underline{x} \rangle = \underline{x}^T R \underline{x} \tag{2.101}$$

$$= \sum_{j=1}^{d} \sum_{i=1}^{d} r_{ij} x_i x_j \tag{2.102}$$

is a polynomial that consists of terms in $\underline{x}_i$ such as $\underline{x}_i^2$, $x_i x_j$, and $x_i$. This is a special version of a quadratic form. Only the symmetric part of $R$ contributes to Equation (2.102).

**Hyperspheres**

A *hypersphere* of radius $r$ is defined by the set of points in $R^n$ satisfying

$$\langle \underline{x}, \underline{x} \rangle \leq r^2 \tag{2.103}$$

Equality in Equation (2.103) describes the points composing the surface of the hypersphere.

For $n = 2$ or 3, ramifications of Equation (2.89) are easily visualized, as shown below.

### 2.2.4 Quadric Surfaces and Boundaries

In $R^d$, with $\underline{x} = (x_1, x_2, \ldots, x_d)^T$, consider the constraint in $R^d$

$$\sum_{i=1}^{d} w_{ii} x_i^2 + \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} w_{ij} x_i x_j + \sum_{i=1}^{d} w_i x_i + w_o = 0 \tag{2.104}$$

Equation (2.104) defines a *quadric surface*, defined by *quadric discriminant functions*. Notice that when $d = 2$, Equation (2.104) reduces to $\underline{x} = (x_1\ x_2)^T$ and

$$w_{11} x_1^2 + w_{22} x_2^2 + w_{12} x_1 x_2 + w_1 x_1 + w_2 x_2 + w_o = 0 \tag{2.105}$$

When $w_{11} = w_{22} = w_{12} = 0$, Equation (2.105) defines a line. If $w_{11} = w_{22} = 1$ and $w_{12} = w_1 = w_2 = 0$, a circle with center at the origin results. When $w_{11} = w_{22} = 0$, a *bilinear constraint* between $x_1$ and $x_2$ results. When $w_{11} = w_{12} = w_2 = 0$, a parabola with a specific orientation results. When $w_{11} \neq 0$, $w_{22} \neq 0$, $w_{11} \neq w_{22}$, and $w_{12} = w_1 = w_2 = 0$, a simple ellipse results.

Extrapolation from the $d = 2$ case in Equation (2.105) suggests that Equation (2.104) defines another family of "hyper" surfaces in $R^d$. First, Equation (2.104) is cast in a more compact and tractable form. There are $[(d + 1)(d + 2)]/2$ parameters in Equation (2.104), which may be organized as the $d \times d$ matrix $W$,

$$W = [\bar{w}_{ij}] \tag{2.106}$$

where

$$\bar{w}_{ij} = \begin{cases} w_{ii} & \text{if } i = j \\ \frac{1}{2} w_{ij} & \text{if } i \neq j \end{cases} \tag{2.107}$$

and the vector $\underline{w}$,

$$\underline{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix} \tag{2.108}$$

which yields the equivalent representation

$$\underline{x}^T W \underline{x} + \underline{w}^T \underline{x} + w_o = 0 \tag{2.109}$$

**Types of quadric surfaces**

1. If $W$ is positive definite, Equation (2.109) defines a hyperellipsoid surface whose axes are in the directions of the e-vectors of $W$.
2. If $W = kI$, where $k > 0$, Equation (2.109) defines a *hypersphere* [see Equation (2.103)].
3. If $W$ is positive semidefinite, Equation (2.109) defines a *hyperellipsoidal cylinder*.
4. If none of the above cases holds, Equation (2.109) defines a surface referred to as a *hyperhyperboloid*.

**Analysis of the quadratic term**

In Equation (2.109) the analysis of *quadratic* term $\underline{x}^T W \underline{x}$ is particularly useful. Recall that only the symmetric part of $W$ contributes to the value of the quadratic. If $\underline{x}^T W \underline{x} > 0 \; \forall \underline{x} \neq \underline{0}$, then the matrix $W$ is said to be *positive definite*. An e-vector–based transformation of coordinates thus requires all e-values of $W$ to be positive. Similarly, if $\underline{x}^T W \underline{x} \geq 0 \; \forall \underline{x} \neq \underline{0}$, then all e-values of $W$ are required to be nonnegative. In this case we refer to $W$ as *positive semidefinite*.

## 2.3 OPTIMIZATION

### 2.3.1 Gradient Descent–Based Procedures

Gradient approaches are optimization procedures used extensively for the training of certain classes of ANNs. It is important to become comfortable with the underlying concept.

Since $\nabla_x f$ in Equation (2.44) defines the direction of maximum increase in the function, we may maximize (or minimize) a scalar function $f(\underline{x})$ by recursively calculating $\nabla_x f$ and adjusting $\underline{x}$ until we reach a minimum (or maximum). The algorithm for the minimization of a function, termed *steepest descent*, is:

1. Make an initial guess, $\underline{x}^0$.
2. Compute $\nabla_x f$, i.e.,

$$\frac{df(\underline{x}^0)}{d\underline{x}} \tag{2.110}$$

3. Adjust $\underline{x}^0$ to obtain $\underline{x}^1$ by moving in a direction *opposite* to the gradient, i.e.,

$$\underline{x}^1 = \underline{x}^0 - K\left[\frac{df(\underline{x}^0)}{d\underline{x}}\right] \tag{2.111}$$

4. Stop when $\underline{x}^{n+1} - \underline{x}^n$ is sufficiently small.

As an example, consider the equation introduced in Section 2.1.9:

$$A\underline{x} = \underline{y} \tag{2.112}$$

where $A$ is an $n \times n$ matrix and $\underline{x}$ and $\underline{y}$ are $n \times 1$ vectors. In this formulation, consider $A$ and $\underline{y}$ as given, with $\underline{x}$ unknown. Equation (2.111) may be thought of as

1. A matrix equation,
2. A set of $n$ linear constraints of the form

$$\underline{a}_i^T \underline{x} = y_i \tag{2.113}$$

where $\underline{a}_i^T$ is the $i$th row of $A$, or

3. A set of I/O specifications for a neural net, where row $i$ of $A$ and element $y_i$ of $\underline{y}$ are the desired input and output patterns, respectively, and $\underline{x}$ is a set of weights to be determined (see Chapters 4, 5, and 6).

One solution to Equation (2.112) is to (attempt to) compute the "batch" solution

$$\underline{x} = A^{-1}\underline{y} \tag{2.114}$$

However, we instead explore the ramifications of more general and extendible formulations. Assume that there is *at least* one solution to Equation (2.112). Defining

$$\underline{e} = A\underline{x} - \underline{y} \tag{2.115}$$

we note that $\underline{e} = \underline{0}$ when a solution to Equation (2.112) is found.

Instead of dealing with $\underline{e}$ directly, consider

$$E = \|\underline{e}\|^2 = e_1^2 + e_2^2 + \cdots + e_d^2 \tag{2.116}$$

where $e_i (i = 1, 2, \ldots, d)$ is an element of vector $\underline{e}$. With this choice of *error function*, when $E = 0$ a solution is found. $E = 0$ is therefore the minimum error. From Equation (2.115),

$$E = \|\underline{e}\|^2 = \langle \underline{e}, \underline{e} \rangle = \underline{e}^T\underline{e} = (A\underline{x} - \underline{y})^T(A\underline{x} - \underline{y}) = (\underline{x}^T A^T - \underline{y}^T)(A\underline{x} - \underline{y}) \tag{2.117}$$

$$= \underline{x}^T A^T A\underline{x} - \underline{x}^T A^T \underline{y} - \underline{y}^T A\underline{x} + \underline{y}^T\underline{y} \tag{2.118}$$

Computing the gradient of $E(\underline{x})$ with respect to $\underline{x}$ in Equation (2.118) yields

$$\nabla_{\underline{x}} E(\underline{x}) = 2(A^T A)\underline{x} - 2A^T\underline{y} = 2A^T(A\underline{x} - \underline{y}) = 2A^T\underline{e} \tag{2.119}$$

Since the gradient of $E$ defines the direction of *maximum increase in E*, Equation (2.119) is used to form an iterative minimization procedure. Consider a procedure to find $\hat{\underline{x}}$, i.e., the solution vector that minimizes Equation (2.118), of the form

$$\hat{\underline{x}}^{n+1} = \hat{\underline{x}}^n - \mu(n)\nabla_{\underline{x}} E(\hat{\underline{x}}^n) \tag{2.120}$$

We show this via a simple 2-D example. Consider a specific example[14] of Equation (2.112) for $d = 2$:

$$\begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 4 \end{pmatrix} \tag{2.122}$$

---

[14]The "batch" (inverse) solution yields

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \tag{2.121}$$

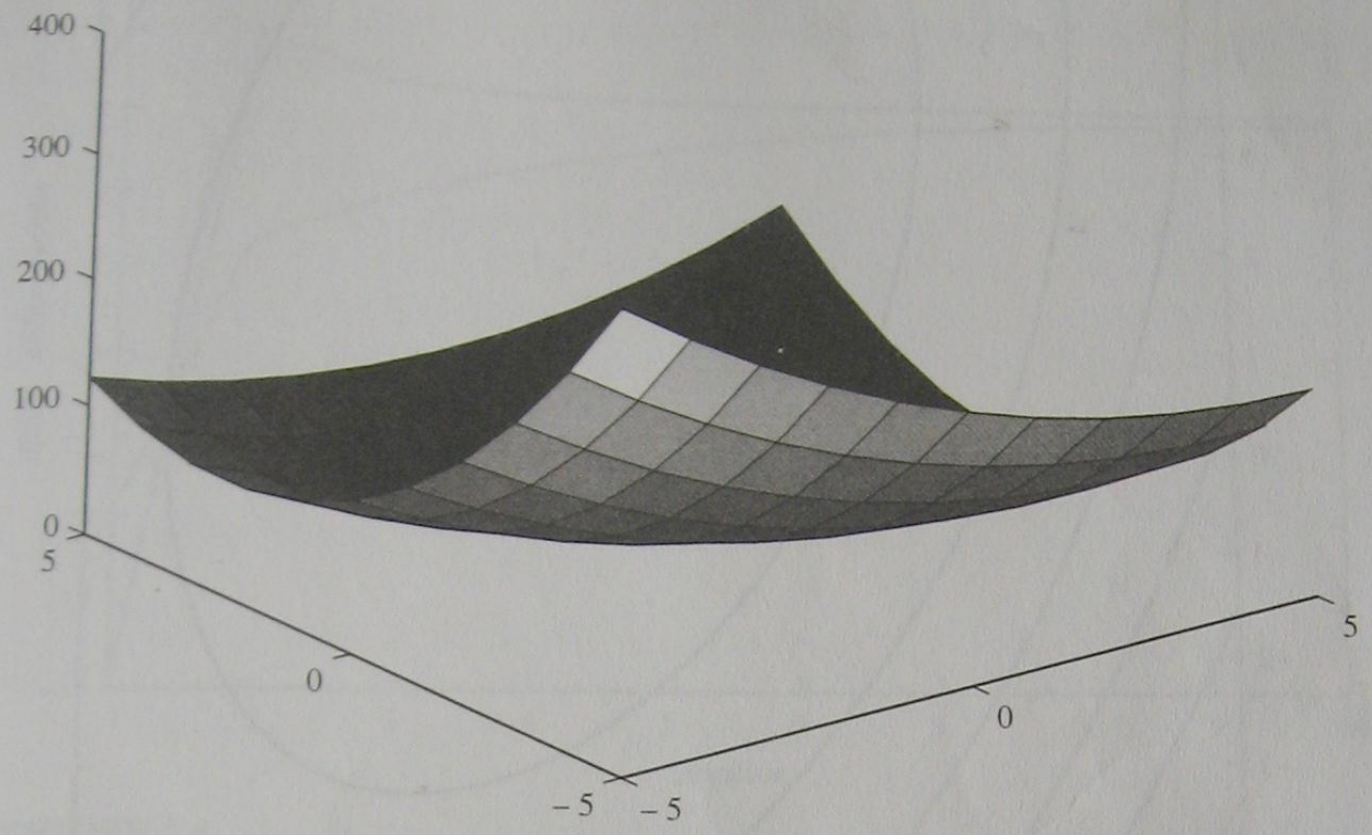



**FIGURE 2.5**
Error function plot for text example.

Formulating the error measure of Equation (2.118) yields

$$E(\underline{x}) = (x_1 \quad x_2)\begin{pmatrix}5 & 1\\ 1 & 2\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + (-14 \quad -10)\begin{pmatrix}x_1\\ x_2\end{pmatrix} + 17 \tag{2.123}$$

or

$$E(\underline{x}) = 5x_1^2 + 2x_1x_2 + 2x_2^2 - 14x_1 - 10x_2 + 17 \tag{2.124}$$

Therefore, $E(\underline{x})$ is quadratic in $x_1$ and $x_2$. This is shown in Figure 2.5.

The loci of constant $E$ are ellipses, given by

$$5x_1^2 + 2x_1x_2 + 2x_2^2 - 14x_1 - 10x_2 = k \tag{2.125}$$

This is shown in Figure 2.6.

A plot of $x_1$ and $x_2$, starting with

$$\underline{x}(0) = \begin{pmatrix}0\\ 0\end{pmatrix}$$

as a function of iteration is shown in Figure 2.7. Also note the behavior of the gradient in the iterative procedure, shown in Figure 2.8.

Gradient descent procedures are considered for ANN applications in depth in Chapters 4, 5, and 6. In addition, Chapter 7 considers the use of conjugate gradient and second-order (derivative) approaches, which may lead to significant improvements in convergence properties.

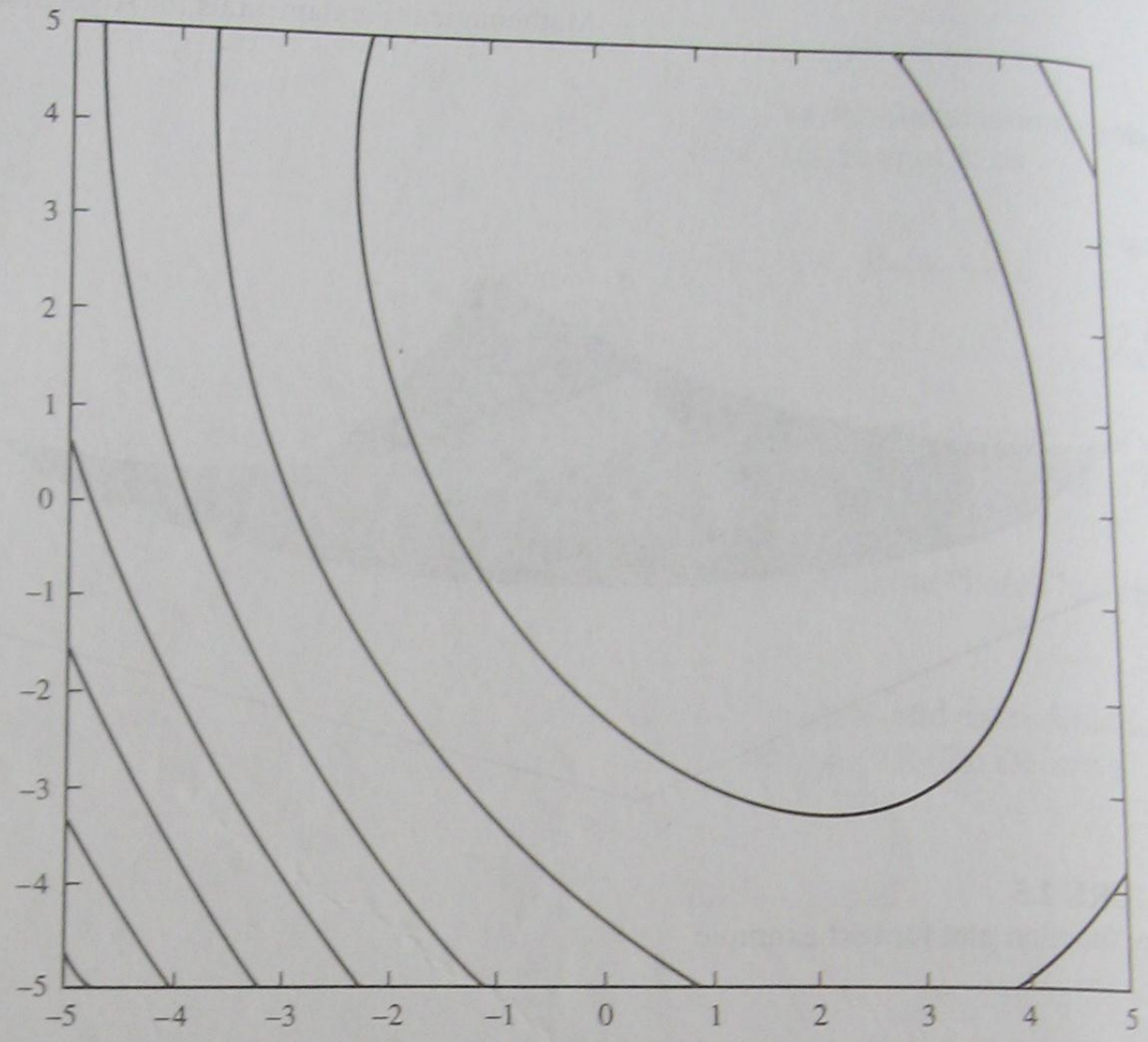



**FIGURE 2.6**
Error function contours for text example.

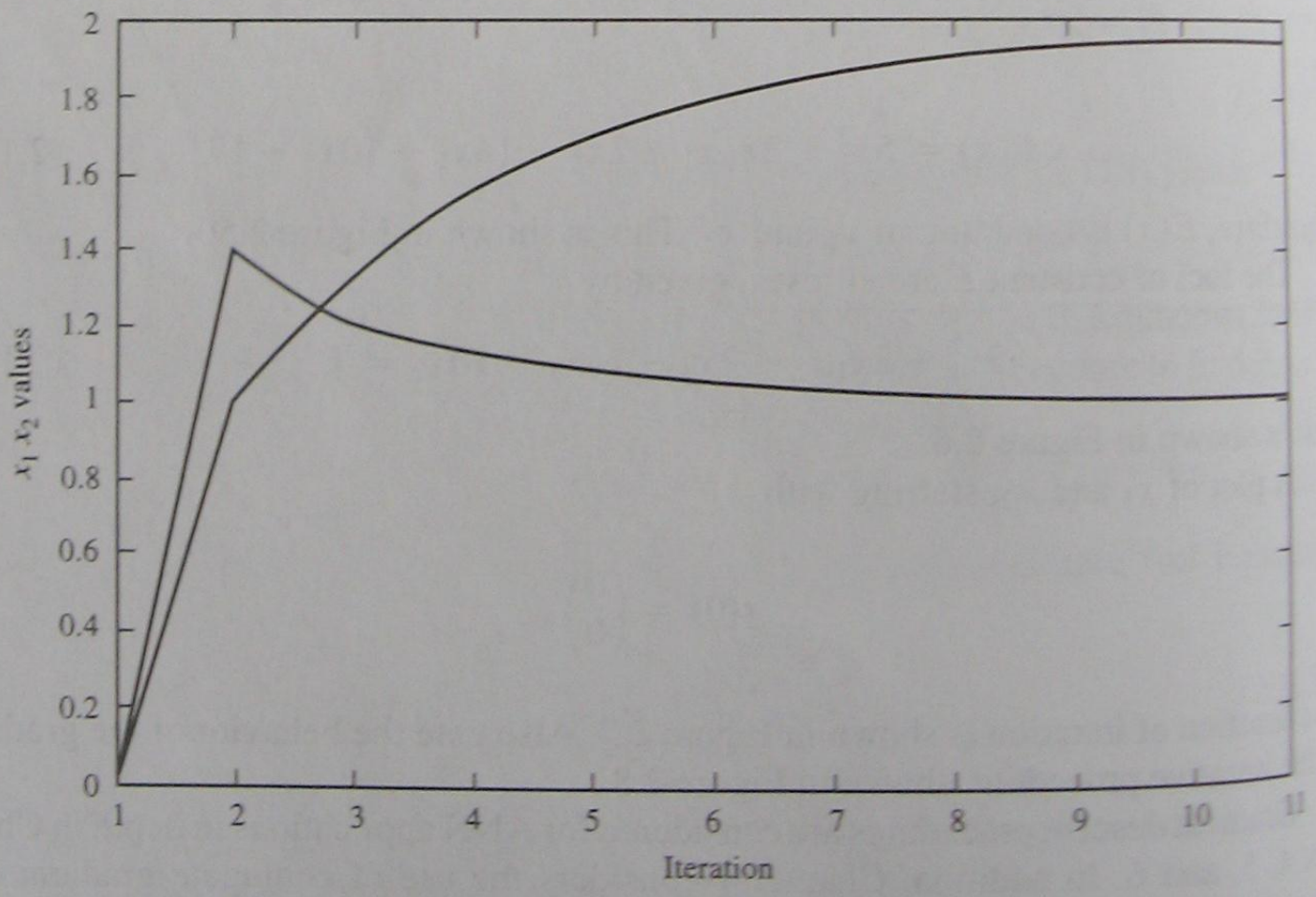



**FIGURE 2.7**
Gradient-based computation of $\underline{x}$.

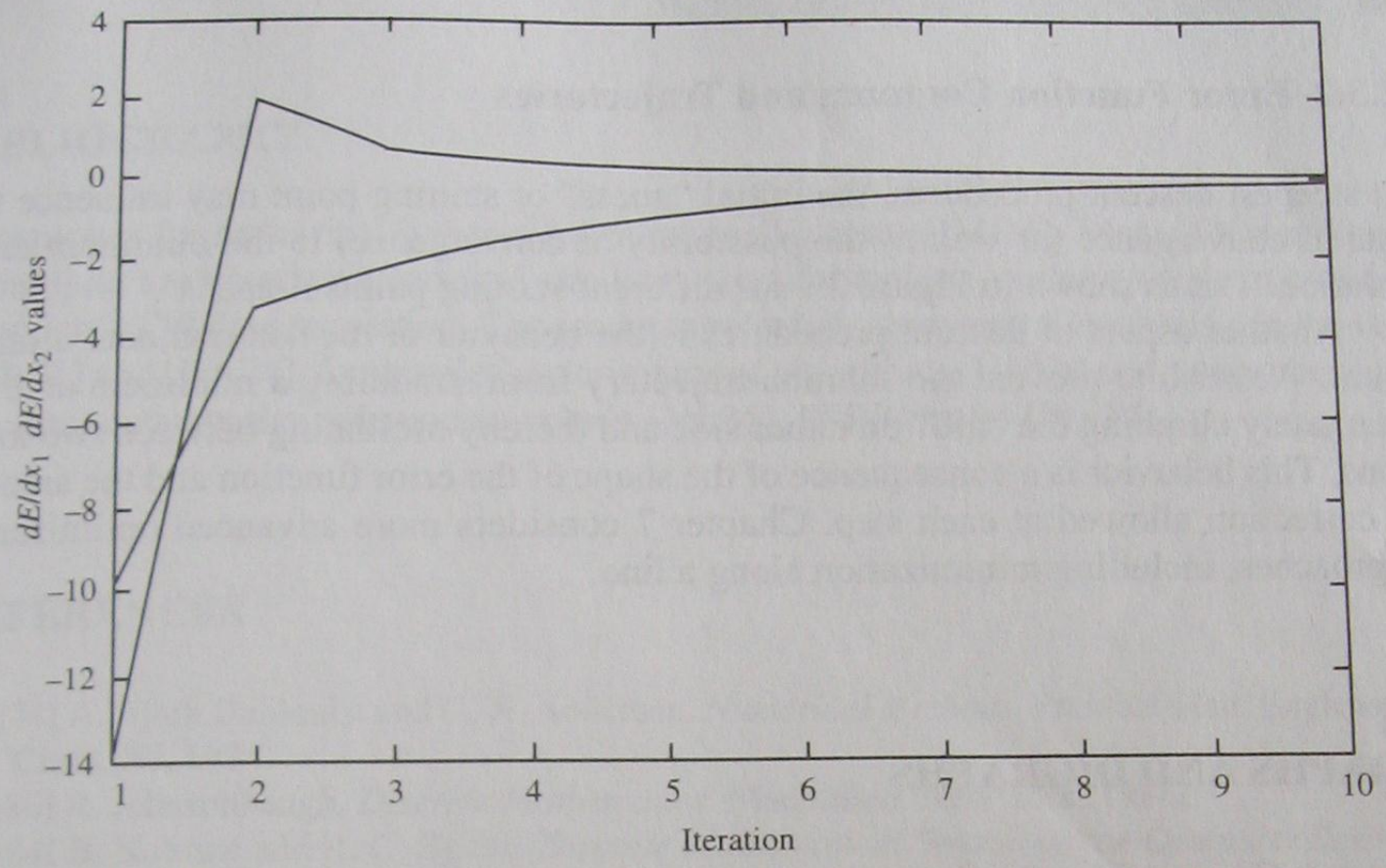



**FIGURE 2.8**
Behavior of gradient during solution to sample problem.

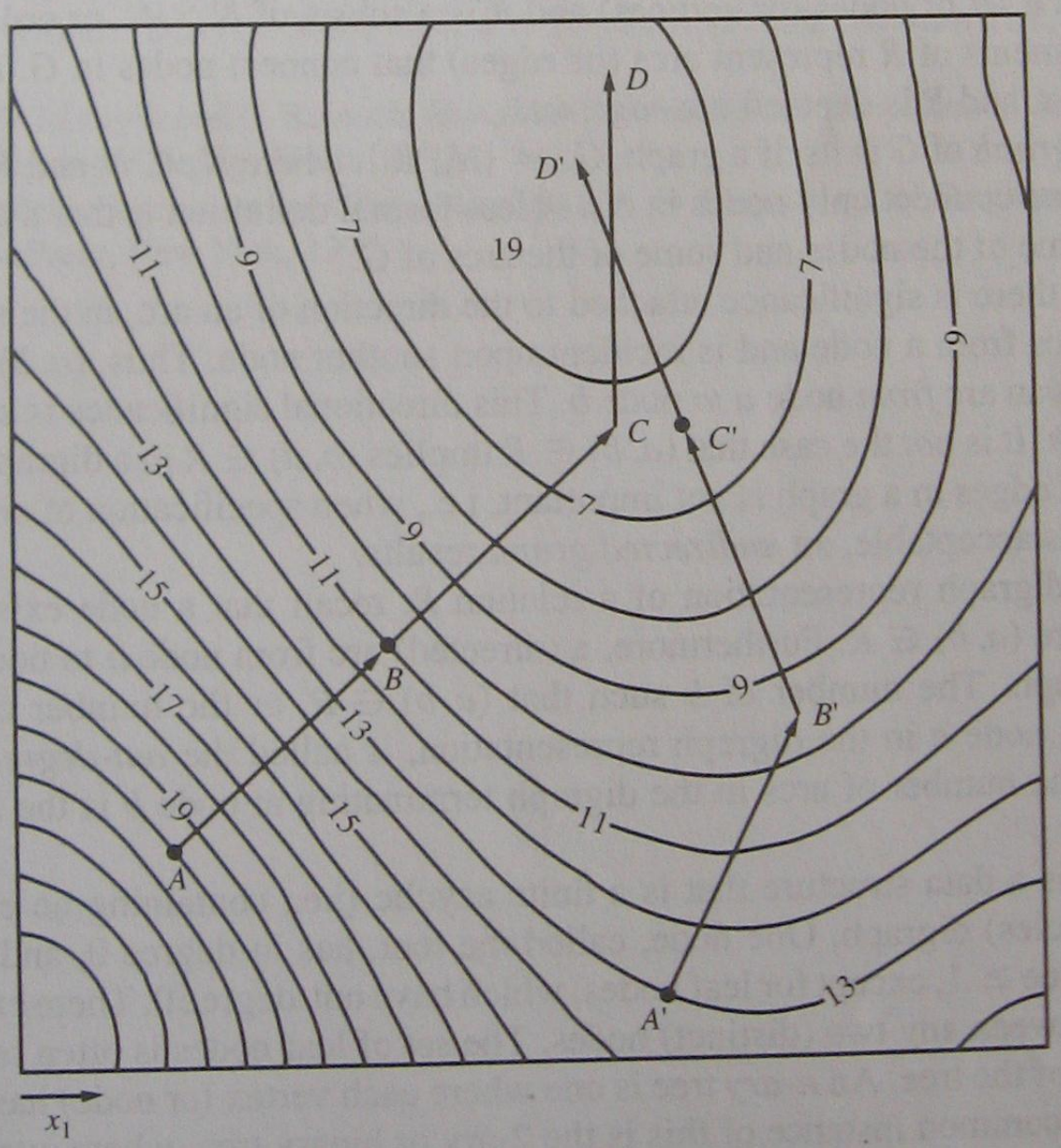



**FIGURE 2.9**
Error contours and trajectories.

### 2.3.2 Error Function Contours and Trajectories

In steepest descent procedures, the initial "guess" or starting point may influence the rate of convergence (as well as the possibility of convergence) to the minimum error solution. This is shown in Figure 2.9 for different starting points $A$ and $A'$.

Another aspect of descent procedures is the behavior of the solution near a minimum. We wish to prevent the solution trajectory from straddling a minimum of $E$ by alternately climbing the "hill" on either side and thereby oscillating between two locations. This behavior is a consequence of the shape of the error function and the amount of correction allowed at each step. Chapter 7 considers more advanced optimization approaches, including minimization along a line.

## 2.4 GRAPHS AND DIGRAPHS

A *graph* $G$ is an ordered pair

$$G = \{N, R\} \tag{2.126}$$

where $N$ is a set of nodes (or vertices) and $R$ is a subset of $N \times N$, or ordered pairs of nodes. Elements of $R$ represent arcs (or edges) that connect nodes in $G$. $N$ is denoted the *node set*, and $R$ is denoted the *edge set*.

A *subgraph* of $G$ is itself a graph, $G_s = \{N_s, R_s\}$, where $N_s \subset N$ and $R_s$ consists of arcs in $R$ that connect only nodes in $N_s$. A less formal definition is that a $G_s$ is a graph that has some of the nodes and some of the arcs of $G$.

Often, there is significance attached to the direction of an arc, in the sense that an arc emanates from a node and is incident upon another node. Thus, $(a, b) \in R$ means that there is an arc *from* node $a$ *to* node $b$. This directional significance is characterized as a *digraph*. It is *not* the case that $(a, b) \in R$ implies $(b, a) \in R$ in a digraph. When the direction of edges in a graph is not important, i.e., when specification of either $(a, b)$ or $(b, a) \in R$ is acceptable, an *undirected graph* results.

In the digraph representation of a relation $R$, recall that a node exists for every $a \in A$ where $(a, b) \in R$. Furthermore, a (directed) arc from node $a$ to node $b$ appears in the digraph. The number of $b$ such that $(a, b) \in R$, or the number of arcs emanating from node $a$ in the digraph representation, is called the *out-degree* of node $a$. Similarly, the number of arcs in the digraph terminating at node $b$ is the *in-degree* of node $b$.

A *tree* is a data structure that is a finite acyclic (i.e., containing no closed loops, paths, or cycles) digraph. One node, called the root, has in-degree 0, and every node has out-degree $\geq 1$, except for leaf nodes, which have out-degree 0. There exists exactly one path between any two (distinct) nodes. The set of leaf nodes is often referred to as the *frontier* of the tree. An *n-ary* tree is one where each vertex (or node) has out-degree $n$ or less. A common instance of this is the 2-ary or binary tree, where every node has either 0 or 2 descendants.

## 2.5 BIBLIOGRAPHY

Techniques for numerical linear algebra are well known [DA74]. Most of the modern techniques are based on a series of similarity transformations, perhaps the most popular of which is QR decomposition. The reader is referred to numerical methods handbooks, [WR71] and [DA74], for detailed descriptions of algorithms. Useful and comprehensive discrete mathematics references include [Joh86], [KB84], and [Wii87].

## REFERENCES

[DA74] A. Bjork Dahlquist and G. N. Anderson. *Numerical Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1974.

[Joh86] R. Johnsonbaugh. *Discrete Mathematics*. Macmillan, New York, 1986.

[KB84] B. Kolman and R. C. Busby. *Discrete Mathematical Structures for Computer Science*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

[RM71] C. Rao and S. Mitra. *Generalized Inverse of Matrices and Its Applications*. John Wiley & Sons, New York, 1971.

[Str76] G. Strang. *Linear Algebra and Its Applications*. Academic Press, New York, 1976.

[Wii87] S. A. Wiitala. *Discrete Mathematics: A Unified Approach*. McGraw-Hill, New York, 1987.

[WR71] J. H. Wilkenson and C. Reinsch. *Handbook for Automatic Computation. Vol. 2: Linear Algebra*. Springer-Verlag, Berlin, 1971.

[Zad75] L. Zadeh. *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*. Academic Press, New York, 1975.