EE951: Linear Algebra **eMasters**

Chapter 5: Basic Matrix Operations

Author: Ketan Rajawat

Contents

5.1 Reshaping Matrices

In many programming languages, matrices are often stored and manipulated as vectors. This storage format is known as “column-major” or “row-major” ordering, depending on how the elements are arranged in memory.

In column-major ordering, the elements of a matrix are stored column by column as a single long vector. This means that the elements of the first column are stored consecutively, followed by the elements of the second column, and so on. This format is commonly used in languages like Fortran and MATLAB.

In row-major ordering, the elements of a matrix are stored row by row as a single long vector. The elements of the first row are stored consecutively, followed by the elements of the second row, and so on. This format is commonly used in languages like C, C++, and Python (with libraries like NumPy).

To convert a matrix into a vector in programming, we can use the $\text{vec}(\cdot)$ operator. The $\text{vec}(\cdot)$ operator reshapes the matrix into a column vector by concatenating the columns or rows of the matrix into a single vector.

For example, consider the matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

If we apply the vec() operator to $\mathbf{A}$ in column-major form (e.g. in MATLAB), we obtain:

$$\operatorname{vec}(\mathbf{A}) = \begin{bmatrix} 1 \\ 4 \\ 2 \\ 5 \\ 3 \\ 6 \end{bmatrix}$$

In this case, the vec $(\cdot)$ operator essentially "flattens" the matrix into a column vector by stacking the columns or rows on top of each other. The reader is warned however that different languages may define the vec $(\cdot)$ differently, depending on whether the matrix is stored in row-major of column-major ordering. In row-major ordering, we would have

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \tag{5.1}$$

More generally, we have the reshape command in most languages. The reshaping operation on matrices allows us to change the dimensions or structure of a matrix while preserving the total number of elements. Given a matrix with dimensions $m \times n$, we can reshape it into a new matrix with dimensions $p \times q$, as long as the total number of elements remains the same ($mn = pq$). For example, consider the matrix stored in row-major format (e.g. in C++ or Python):

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

We can reshape $\mathbf{A}$ into a new matrix with dimensions 3×2:

$$\mathbf{B} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

In this example, we changed the shape of $\mathbf{A}$ from 2×3 to 3×2 by rearranging the elements in row-major order. Again, the reshape operation is defined differently in different languages, and depends on the ordering in which a matrix is stored.

5.2 Transpose and Addition

5.2.1 Matrix Transpose

The transpose of a matrix is an operation that flips the matrix over its main diagonal, reflecting its elements across the diagonal. It is denoted by placing a superscript T or an apostrophe ($'$) after the matrix.

For example, consider the matrix A:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

The transpose of matrix $\mathbf{A}$, denoted as $\mathbf{A}^\mathsf{T}$ or $\mathbf{A}'$, is obtained by interchanging its rows and columns:

$$\mathbf{A}^\mathsf{T} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

The rows of the original matrix become the columns of the transpose, and the columns of the original matrix become the rows of the transpose. When a matrix is transposed twice, the result is the original matrix, i.e.,

$$(\mathbf{A}^\mathsf{T})^\mathsf{T} = \mathbf{A}$$

This property holds for any matrix, regardless of its size or elements.

The transpose notation can also be applied to vectors, and the transpose operation converts a row vector into a column vector and vice versa. For instance, the first column of $\mathbf{A}$ in the example above can be written as $\begin{bmatrix} 1 & 4 \end{bmatrix}^\mathsf{T}$.

Let us consider a block matrix $\mathbf{A}$ with submatrices $\mathbf{B}$ and $\mathbf{C}$ arranged as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \end{bmatrix}$$

The transpose of $\mathbf{A}$, denoted as $\mathbf{A}^\mathsf{T}$, is obtained by taking the transpose of the block matrix, with each element also transposed:

$$\mathbf{A}^\mathsf{T} = \begin{bmatrix} \mathbf{B}^\mathsf{T} \\ \mathbf{C}^\mathsf{T} \end{bmatrix}.$$

A **symmetric matrix** is a square matrix that is the same as its transpose. In other words, if we have a matrix $\mathbf{A}$ of size $n \times n$, it is symmetric if and only if $\mathbf{A} = \mathbf{A}^\mathsf{T}$. In other words, the entries above the main diagonal are mirror images of the entries below the main diagonal. Here is an example of a symmetric matrix:

$$\mathbf{A} = \begin{bmatrix} 2 & 5 & 7 \\ 5 & 1 & 9 \\ 7 & 9 & 4 \end{bmatrix}$$

In this matrix, you can observe that $A_{ij} = A_{ji}$ for all $1 \leq i, j \leq n$. A symmetric matrix may have at most $n(n+1)/2$ unique elements.

5.2.2 Matrix Addition and Subtraction

Matrix addition is an operation performed on two matrices of the same size, where corresponding entries in the matrices are added together to create a new matrix. The resulting matrix has the same dimensions as the original matrices. The addition of matrices follows the element-wise addition rule. Consider the following matrices:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

To add these matrices, we simply add the corresponding entries together:

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} 1+5 & 2+6 \\ 3+7 & 4+8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

Matrix subtraction is similar to matrix addition, but instead of adding the corresponding entries, we subtract them. Like matrix addition, matrix subtraction is performed on matrices of the same size, and the resulting matrix also has the same dimensions. Using the same matrices $\mathbf{A}$ and $\mathbf{B}$ as in the previous example, we can subtract them as follows:

$$\mathbf{A} - \mathbf{B} = \begin{bmatrix} 1-5 & 2-6 \\ 3-7 & 4-8 \end{bmatrix} = \begin{bmatrix} -4 & -4 \\ -4 & -4 \end{bmatrix}$$

In matrix subtraction, we subtract the corresponding entries of the matrices to obtain the resulting matrix. We can list the following properties of matrix addition.

1. *Commutativity:* changing the order of the matrices being added does not affect the result, i.e., $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$;
2. *Associativity:* when adding more than two matrices together, the grouping of addition does not affect the result, i.e., $(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$;
3. *Additive Identity:* adding a zero matrix leaves the matrix unchanged, i.e., $\mathbf{A} + \mathbf{0} = \mathbf{A}$; and
4. *Transpose of Sum:* the transpose of the sum of two matrices is the same as the sum of their transposes, i.e., $(\mathbf{A} + \mathbf{B})^\mathsf{T} = \mathbf{A}^\mathsf{T} + \mathbf{B}^\mathsf{T}$.

5.3 Scaling a Matrix

Scalar multiplication with a matrix or scaling is an operation where each entry of the matrix is multiplied by a scalar. It is similar to scalar multiplication with vectors, where each component of the vector is multiplied by a scalar. Given a scalar α and an $m \times n$ matrix $\mathbf{A}$, the scalar multiplication is denoted as $\alpha\mathbf{A}$. We also have that

$$\alpha\mathbf{A} = \mathbf{A}\alpha \tag{5.2}$$

When multiplying a scalar with a matrix, each entry of the matrix is multiplied by the scalar. This operation is distributive over matrix addition, meaning that for scalars α and β and an $m \times n$ matrix $\mathbf{A}$, we have $\alpha(\mathbf{A} + \mathbf{B}) = \alpha\mathbf{A} + \alpha\mathbf{B}$ and $(\alpha + \beta)\mathbf{A} = \alpha\mathbf{A} + \beta\mathbf{A}$. This property is analogous to scalar multiplication with vectors.

Matrix multiplication with zero scalar has a special property. For any matrix $\mathbf{A}$, we have $0\mathbf{A} = \mathbf{0}$, where $\mathbf{0}$ represents a matrix of the same size as $\mathbf{A}$ filled with zeros. Multiplying any matrix by zero yields a matrix consisting entirely of zeros.

Other properties include associativity with scalar multiplication, where $(\alpha\beta)\mathbf{A} = \alpha(\beta\mathbf{A})$ for scalars α and β and matrix $\mathbf{A}$, and compatibility with scalar addition, where $(\alpha + \beta)\mathbf{A} = \alpha\mathbf{A} + \beta\mathbf{A}$ for scalars α and β and matrix $\mathbf{A}$.

5.4 Vector Space of Matrices

The concept of linear combinations can be applied to matrices, where we multiply each element of a matrix by a scalar and sum the scaled matrices together. Consider two matrices, $\mathbf{A}$ and $\mathbf{B}$ of the same size, and two scalars, c and d. The linear combination of $\mathbf{A}$ and $\mathbf{B}$ is given by the expression $c\mathbf{A} + d\mathbf{B}$, where c and d are scalars. For example, let $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} -1 & 0 \\ 2 & -3 \end{bmatrix}$. Then, a linear combination of $\mathbf{A}$ and $\mathbf{B}$ could be $2\mathbf{A} + 3\mathbf{B}$. This would result in:

$$2\mathbf{A} + 3\mathbf{B} = 2\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + 3\begin{bmatrix} -1 & 0 \\ 2 & -3 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 8 & 10 \end{bmatrix}$$

.

The space of all $m \times n$ matrices with entries from a field $\mathbb{F}$ is denoted by $\mathbb{F}^{m\times n}$. For instance, the space of all real-valued $m \times n$ matrices is denoted by $\mathbb{R}^{m\times n}$. It is straightforward to verify that $\mathbb{R}^{m\times n}$, for instance, contains the zero element and is closed under linear combinations.

5.5 Matrix-Vector Multiplication

Matrix-vector multiplication is an important operation in linear algebra that combines the elements of a matrix and a vector to produce a new vector. This operation is denoted using both compact and expanded notations.

In compact notation, matrix-vector multiplication is expressed as $\mathbf{Av}$, where $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{v}$ is an $n \times 1$ column vector. The resulting vector, denoted as $\mathbf{u}$, is an $m \times 1$ column vector.

Expanded notation provides a clearer representation of the calculation. Let us consider an example to illustrate this. Suppose we have a matrix $\mathbf{A}$ and a vector $\mathbf{v}$:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

To compute the matrix-vector multiplication, we multiply each row of the matrix $\mathbf{A}$ by the corresponding element of the vector $\mathbf{v}$, and then sum the products. This can be represented as follows:

$$\mathbf{u} = \mathbf{Av} = \begin{bmatrix} a_{11}v_1 + a_{12}v_2 + \ldots + a_{1n}v_n \\ a_{21}v_1 + a_{22}v_2 + \ldots + a_{2n}v_n \\ \vdots \\ a_{m1}v_1 + a_{m2}v_2 + \ldots + a_{mn}v_n \end{bmatrix}$$

In this computation, each element of the resulting vector $\mathbf{u}$ is obtained by taking a weighted sum of the corresponding elements of the matrix rows, where the weights are given by the elements of the vector $\mathbf{v}$. The size of the resulting vector $\mathbf{u}$ is determined by the number of rows in the matrix $\mathbf{A}$. We can interpret $\mathbf{u}$ as a linear combination of the columns of $\mathbf{A}$, since

$$\mathbf{u} = v_1\mathbf{a}_1 + v_2\mathbf{a}_2 + \ldots + v_n\mathbf{a}_n \tag{5.3}$$

Here the coefficients of the linear combination are given by the elements of the vector $\mathbf{v}$. Each entry of the resulting vector is obtained by taking a weighted sum of the columns of $\mathbf{A}$, where the weights are determined by the corresponding elements of $\mathbf{v}$.

Example 5.1. As an example, consider the matrix $\mathbf{A}$ and vector $\mathbf{v}$:

$$\mathbf{A} = \begin{bmatrix} 2 & 4 \\ 1 & 3 \\ 5 & 2 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

To compute the matrix-vector multiplication $\mathbf{u} = \mathbf{Av}$, we multiply each row of the matrix $\mathbf{A}$ by the

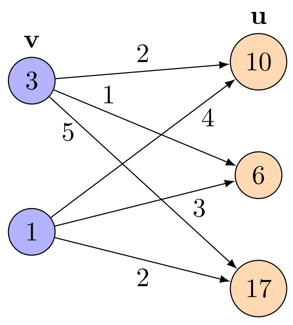

Figure 5.1: Matrix-vector product represented as a neural network

corresponding element of the vector **v**, and then sum the products.

$$\mathbf{u} = \mathbf{A}\mathbf{v} = \begin{bmatrix} 2(3) + 4(1) \\ 1(3) + 3(1) \\ 5(3) + 2(1) \end{bmatrix} = \begin{bmatrix} 10 \\ 6 \\ 17 \end{bmatrix}$$

In this example, the resulting vector **u** is a 3×1 column vector, obtained by multiplying each row of **A** by the corresponding element of **v** and summing the products.

More recently, it is also common to visualize the matrix-vector product as a neural network; see Fig. 5.1.

5.5.1 Applications

Let us have a look at some simple examples of matrix-vector products.

1. We can obtain the i-th column of **A** by multiplying it with the i-th standard unit vector, i.e., $\mathbf{A}\mathbf{e}_i = \mathbf{a}_i$. Likewise, we can obtain the rows as $\mathbf{A}^\top \mathbf{e}_j = \mathbf{a}_{j\cdot}^\top$.
2. The vector containing the sum of columns of **A** is given by **A1** where **1** is the all-one vector of appropriate size. Likewise, the sum of rows can be obtained by calculating $\mathbf{A}^\top \mathbf{1}$.
3. The difference matrix is given by

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

When multiplied by the vector **v**, we obtain the vector of forward differences

$$\mathbf{D} \cdot \mathbf{v} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix} = \begin{bmatrix} v_2 - v_1 \\ v_3 - v_2 \\ v_4 - v_3 \\ v_5 - v_4 \end{bmatrix}$$

4. The cumulative sum matrix $\mathbf{S}$ is defined as:

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

To calculate the result of multiplying the cumulative sum matrix $\mathbf{S}$ with the vector $\mathbf{v}$, we perform matrix-vector multiplication:

$$\mathbf{S} \cdot \mathbf{v} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_1 + v_2 \\ v_1 + v_2 + v_3 \\ v_1 + v_2 + v_3 + v_4 \\ v_1 + v_2 + v_3 + v_4 + v_5 \end{bmatrix}$$

As shown in the example, multiplying the cumulative sum matrix with the vector $\mathbf{v}$ yields a new vector that represents the cumulative sum of the elements in $\mathbf{v}$. Each element in the resulting vector represents the sum of all elements from the beginning of the vector up to that particular index.

5.5.2 Properties

The matrix-vector product has several important properties:

1. *Distributive over vector addition:* For matrices $\mathbf{A}$ and $\mathbf{B}$ of compatible dimensions, and a vector $\mathbf{v}$, we have $\mathbf{A}(\mathbf{v} + \mathbf{w}) = \mathbf{A}\mathbf{v} + \mathbf{A}\mathbf{w}$. Here the matrix-vector multiplication has higher precedence than vector addition.
2. *Distributive over matrix addition:* For matrices $\mathbf{A}$ and $\mathbf{B}$ of compatible dimensions, and a vector $\mathbf{v}$, we have $(\mathbf{A} + \mathbf{B})\mathbf{v} = \mathbf{A}\mathbf{v} + \mathbf{B}\mathbf{v}$.
3. *Multiplication with scalar:* For a matrix $\mathbf{A}$ and a scalar α, we have $(\alpha\mathbf{A})\mathbf{v} = \alpha(\mathbf{A}\mathbf{v}) = \mathbf{A}(\alpha\mathbf{v})$.

5.6 Complexity of Matrix Operations

Flop counts for various matrix operations can be calculated in a similar way.

1. Transposing an $n \times n$ matrix involves exchanging the elements across the main diagonal. Each element in the original matrix needs to be moved to its corresponding position in the transposed matrix. The flop count of matrix transpose is zero, but there is still time required to copy the entries from one location to another.
2. Scaling an $n \times n$ matrix by a scalar α requires multiplying each element of the matrix by α. Since there are n^2 elements in the matrix, the flop count for matrix scaling is n^2.
3. Adding two $n \times n$ matrices involves adding the corresponding elements of the matrices. Since there are n^2 elements to be added, the flop count for matrix addition is n^2.
4. In the matrix-vector product, each entry of the resulting n-dimensional vector is computed by taking the product of a row from the matrix with the vector. Since each such product takes $2n - 1$ flops and there are a total of n such products required to be taken, the total flop count for the matrix-vector product is $2n^2 - n$ floating-point operations.

Table 5.1: Flop counts for matrix operations

Operation	**Flop count**
Transpose	0
Scaling	n^2
Addition	n^2
Matrix-Vector Multiplication	$2n^2$