EE951: Linear Algebra **eMasters**

Chapter 6: Matrix Multiplication

Author: Ketan Rajawat

Contents

6.1 Introduction

Matrix multiplication is an operation performed between two matrices, resulting in a new matrix. To multiply two matrices, the number of columns in the first matrix must be equal to the number of rows in the second matrix. Suppose we have two matrices, $\mathbf{A}$ of size $m \times n$ and $\mathbf{B}$ of size $n \times p$, their product $\mathbf{C} = \mathbf{AB}$ is a matrix of size $m \times p$.

To compute the entry at the ith row and jth column of the product matrix $\mathbf{C}$, denoted as C_{ij}, we use the calculation:

$$C_{ij} = \sum_{k=1}^{n} A_{ik} \cdot B_{kj}$$

In words, to calculate C_{ij}, we multiply the corresponding entries of the ith row of $\mathbf{A}$ with the jth column of $\mathbf{B}$, and then sum the results. One way to remember this rule is to think of it as a "row-by-column" process.

As an example, let $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 7 & 8 \\ 9 & 10 \end{bmatrix}$. To find the product $\mathbf{C} = \mathbf{AB}$, we compute:

$$c_{11} = (1 \cdot 7) + (2 \cdot 9) = 25$$

$$c_{12} = (1 \cdot 8) + (2 \cdot 10) = 28$$

$$c_{21} = (3 \cdot 7) + (4 \cdot 9) = 57$$

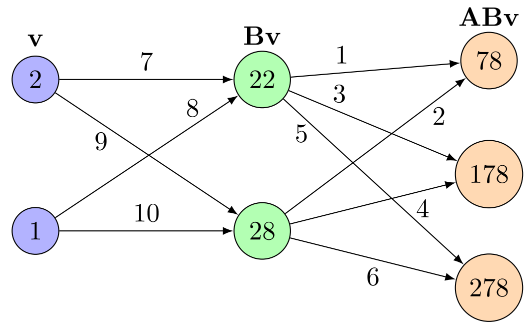

Figure 6.1: Matrix-matrix-vector product represented as a neural network.

$$c_{22} = (3 \cdot 8) + (4 \cdot 10) = 64$$
$$c_{31} = (5 \cdot 7) + (6 \cdot 9) = 89$$
$$c_{32} = (5 \cdot 8) + (6 \cdot 10) = 100$$

Therefore, the resulting matrix $\mathbf{C}$ is $\begin{bmatrix} 25 & 28 \\ 57 & 64 \\ 89 & 100 \end{bmatrix}$. The matrix multiplication subsumes matrix-vector product as a special case but not generally the scalar-vector or scalar-matrix products. This is because if we consider a scalar as a 1×1 matrix, then its size is not compatible for multiplication with an $m \times n$ matrix for $m, n > 1$. However, it would be compatible to left-multiply a scalar with a row vector or right-multiply a scalar with a column vector, in which case it becomes a special case of matrix multiplication.

We note that a matrix-matrix-vector product can again be represented as a neural network, as shown in Fig. 6.1, where we calculate $\mathbf{u} = \mathbf{ABv}$.

The vector **outer product** is a special case of matrix multiplication that involves multiplying a column vector with a row vector to produce a matrix. Given a column vector $\mathbf{u}$ of size $m \times 1$ and a row vector $\mathbf{v}^\mathsf{T}$ of size $1 \times n$, their outer product $\mathbf{uv}^\mathsf{T}$ yields a matrix of size $m \times n$. The (i, j)-th entry of the outer product matrix $\mathbf{uv}^\mathsf{T}$ can be computed using the formula:

$$[\mathbf{uv}^\mathsf{T}]_{ij} = u_i v_j$$

where u_i is the i-th element of vector $\mathbf{u}$ and v_j is the j-th element of the vector $\mathbf{v}$. It's important to note that the outer product matrix is not symmetric in general, meaning that $\mathbf{uv}^\mathsf{T}$ is not always equal to $\mathbf{vu}^\mathsf{T}$. Furthermore, if $\mathbf{u}$ and $\mathbf{v}$ have different lengths (i.e., $m \neq n$), the resulting matrix will be rectangular.

The identity matrix $\mathbf{I}$ is the multiplicative identity for matrix multiplication, so that $\mathbf{AI} = \mathbf{A}$ and $\mathbf{IA} = \mathbf{A}$. Observe that the size of $\mathbf{I}$ when left/right multiplying a rectangular matrix has to be different.

6.1.1 Non-Commutativity

Matrix multiplication is not commutative in general, which means that the order of multiplication matters. Consider matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times p}$, so that $\mathbf{AB} \in \mathbb{R}^{m \times p}$ can be calculated. It can be seen that in this case $\mathbf{BA}$ is not defined since the sizes of $\mathbf{B}$ and $\mathbf{A}$ are incompatible.

In the special case when $m = p$, it is possible to calculate both $\mathbf{AB}$ and $\mathbf{BA}$, but even then, these may not necessarily be equal. For instance, consider

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

$$\mathbf{B} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

Calculating $\mathbf{AB}$, we have:

$$\mathbf{AB} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

and calculating $\mathbf{BA}$, we have:

$$\mathbf{BA} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 23 & 34 \\ 31 & 46 \end{bmatrix}$$

As we can see, $\mathbf{AB}$ and $\mathbf{BA}$ are not equal, demonstrating that matrix multiplication is not commutative in general.

6.1.2 Properties of Matrix Multiplication

Some of the basic properties of matrix multiplication are as follows

1. *Associativity:* Matrix multiplication is associative, which means that the order in which we multiply matrices does not matter. For any three matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ of compatible sizes, we have
$$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$$
2. *Associativity with scalar multiplication:* We can freely move scalar multiplication around when multiplying matrices. For any matrix $\mathbf{A}$ and scalar c, we have $(c\mathbf{A})\mathbf{B} = c(\mathbf{AB}) = \mathbf{A}(c\mathbf{B})$.
3. *Distributivity with addition:* Matrix multiplication distributes over matrix addition. For any matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ of compatible sizes, we have $\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB}+\mathbf{AC}$.

An implication of the distributivity property is the following equality:

$$\begin{aligned}(\mathbf{A}+\mathbf{B})(\mathbf{C}+\mathbf{D}) &= \mathbf{A}(\mathbf{C}+\mathbf{D}) + \mathbf{B}(\mathbf{C}+\mathbf{D}) \\ &= \mathbf{AC}+\mathbf{AD}+\mathbf{BC}+\mathbf{BD}\end{aligned}$$

Let us establish another result related to the transpose of a product of matrices.

Lemma 6.1. The transpose of a matrix product is the product of the transposes in reverse order. For any matrices $\mathbf{A}$ and $\mathbf{B}$, we have $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$.

This property shows that taking the transpose of a product is equivalent to taking the product of the transposes in reverse order.

Proof: We begin with assuming that $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is an $n \times p$ matrix. This implies that the product $\mathbf{AB}$ is an $m \times p$ matrix.

Next, let us calculate the transpose of $(\mathbf{AB})^T$:

$$[(\mathbf{AB})^{\mathsf{T}}]_{ij} = [\mathbf{AB}]_{ji} = \sum_{k=1}^{n} A_{jk}B_{ki}$$

for $1 \leq j \leq m$ and $1 \leq i \leq p$. In the same way, we have that

$$[\mathbf{B}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}]_{ij} = \sum_{k=1}^{n} [\mathbf{B}^{\mathsf{T}}]_{ik}[\mathbf{A}^{\mathsf{T}}]_{kj} = \sum_{k=1}^{n} B_{ki}A_{jk} = \sum_{k=1}^{n} A_{jk}B_{ki}$$

which is the same as $[(\mathbf{AB})^{\mathsf{T}}]_{ij}$. Here, note that the first equality follows from the definition of the matrix product, the second from the definition of the transpose, and third from the commutativity of scalar multiplication. Since the (i, j) element of $(\mathbf{AB})^T$ is equal to the (i, j) element of $\mathbf{B}^T\mathbf{A}^T$ for all i and j, we can conclude that $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$. ■

6.1.3 Multiplication of Block Matrices

Let us consider two block matrices:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

and

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}$$

where $\mathbf{A}_{ij}$ and $\mathbf{B}_{ij}$ represent the individual blocks. To compute the product $\mathbf{AB}$, we multiply the corresponding blocks and perform the necessary additions. The resulting block matrix will have the same block structure:

$$\mathbf{AB} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_{11} + \mathbf{A}_{12}\mathbf{B}_{21} & \mathbf{A}_{11}\mathbf{B}_{12} + \mathbf{A}_{12}\mathbf{B}_{22} \\ \mathbf{A}_{21}\mathbf{B}_{11} + \mathbf{A}_{22}\mathbf{B}_{21} & \mathbf{A}_{21}\mathbf{B}_{12} + \mathbf{A}_{22}\mathbf{B}_{22} \end{bmatrix}$$

It is important to note that for this multiplication to be defined, the sizes of the individual blocks must be compatible. Specifically, the number of columns in $\mathbf{A}_{ij}$ must be equal to the number of rows in $\mathbf{B}_{ij}$. If this condition is satisfied, we can perform the block matrix multiplication as described above. This approach can be extended to block matrices of larger sizes by applying the same principles and considering the compatibility of block sizes.

6.1.4 Interpretations of Matrix Multiplication

Column Interpretation: Matrix-matrix multiplication can be interpreted in terms of matrix-vector products between the first matrix and the columns of the second matrix. Given a matrices $\mathbf{A}$ and $\mathbf{B}$, where the columns of $\mathbf{B}$ are denoted by $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_p$, the product $\mathbf{AB}$ is given by

$$\mathbf{AB} = \mathbf{A}\begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \ldots & \mathbf{b}_m \end{bmatrix} = \begin{bmatrix} \mathbf{Ab}_1 & \mathbf{Ab}_2 & \ldots & \mathbf{Ab}_p \end{bmatrix}$$

In other words, the product $\mathbf{AB}$ can be obtained by applying the transformation $\mathbf{A}$ to each column of $\mathbf{B}$. This interpretation is often used for combining multiple linear systems of equations $\{\mathbf{Ax}_k = \mathbf{y}_k\}_{k=1}^{p}$ together as

$$\mathbf{A}\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_p \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \ldots & \mathbf{y}_p \end{bmatrix} \tag{6.1}$$

which can be compactly written as

$$\mathbf{AX} = \mathbf{Y}. \tag{6.2}$$

Row Interpretation: In a similar way, we can interpret matrix multiplication in terms of matrix-vector product between the transpose of the second matrix and the vectors formed by the rows of the first matrix. That is, given a matrix $\mathbf{A}$ with rows $\mathbf{a}_{1\cdot}, \mathbf{a}_{2\cdot}, \ldots, \mathbf{a}_{m\cdot}$, we have that

$$\mathbf{AB} = \begin{bmatrix} \mathbf{a}_{1\cdot} \\ \mathbf{a}_{2\cdot} \\ \vdots \\ \mathbf{a}_{m\cdot} \end{bmatrix} \mathbf{B} = \begin{bmatrix} \mathbf{a}_{1\cdot}\mathbf{B} \\ \mathbf{a}_{2\cdot}\mathbf{B} \\ \vdots \\ \mathbf{a}_{m\cdot}\mathbf{B} \end{bmatrix} = \begin{bmatrix} (\mathbf{B}^\mathsf{T}\mathbf{a}_{1\cdot}^\mathsf{T})^\mathsf{T} \\ (\mathbf{B}^\mathsf{T}\mathbf{a}_{2\cdot}^\mathsf{T})^\mathsf{T} \\ \vdots \\ (\mathbf{B}^\mathsf{T}\mathbf{a}_{m\cdot}^\mathsf{T})^\mathsf{T} \end{bmatrix}. \tag{6.3}$$

where we first wrote $\mathbf{A}$ as a stack of its rows, multiply with $\mathbf{B}$, and take transpose by observing that the transpose of transpose is the matrix itself.

Inner Product Interpretation: The inner product of $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ is defined as

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^\mathsf{T}\mathbf{v} \tag{6.4}$$

The product of two matrices, $\mathbf{A}$ and $\mathbf{B}$, can be expressed in terms of the inner product of the rows of $\mathbf{A}$ with the columns of $\mathbf{B}$.

Let

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_{1\cdot} \\ \mathbf{a}_{2\cdot} \\ \vdots \\ \mathbf{a}_{m\cdot} \end{bmatrix} \tag{6.5}$$

where $\mathbf{a}_i^\mathsf{T} \in \mathbb{R}^n$ and

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \ldots & \mathbf{b}_p \end{bmatrix} \tag{6.6}$$

where $\mathbf{b}_i \in \mathbb{R}^n$. The (i, j)-th entry of the product matrix $\mathbf{AB}$ can be calculated as:

$$(\mathbf{AB})_{ij} = \mathbf{a}_{i\cdot}\mathbf{b}_j = \langle \mathbf{a}_{i\cdot}^\mathsf{T}, \mathbf{b}_j \rangle$$

This interpretation means that to compute each entry of the product matrix, we take the inner product of the corresponding row of $\mathbf{A}$ (written as a column vector) with the corresponding column of $\mathbf{B}$. The resulting inner product is then placed in the corresponding entry of the product matrix.

Gram Matrix: For an $m \times n$ matrix $\mathbf{A}$ with columns $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$, the Gram matrix is given by $\mathbf{G} = \mathbf{A}^\mathsf{T}\mathbf{A}$. Here, we see that the rows of $\mathbf{A}^\mathsf{T}$, expressed as columns vectors, are indeed $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$. In other words,

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \ldots & \mathbf{a}_n \end{bmatrix} \tag{6.7}$$

and

$$\mathbf{A}^\mathsf{T} = \begin{bmatrix} \mathbf{a}_1^\mathsf{T} \\ \mathbf{a}_2^\mathsf{T} \\ \vdots \\ \mathbf{a}_n^\mathsf{T} \end{bmatrix} \tag{6.8}$$

Hence, using the inner product interpretation, we can see that

$$(\mathbf{G})_{ij} = \mathbf{a}_i^\top \mathbf{a}_j = \langle \mathbf{a}_i, \mathbf{a}_j \rangle \tag{6.9}$$

That is, the (i, j)-th entry of $\mathbf{G}$ gives the inner product between the i-th and j-th columns of $\mathbf{A}$. We can see that the Gram matrix $\mathbf{G}$ is symmetric since

$$\mathbf{G}^\top = (\mathbf{A}^\top \mathbf{A})^\top = \mathbf{A}^\top (\mathbf{A}^\top)^\top = \mathbf{A}^\top \mathbf{A} = \mathbf{G}.$$

The name "Gram matrix" is derived from the mathematician Jørgen Pedersen Gram, who made significant contributions to the field of linear algebra and matrix theory. The Gram matrix itself is closely related to the concept of the Gram determinant, which was introduced by Jørgen Pedersen Gram in the late 19th century.

Outer Product: Consider the matrices

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix} \tag{6.10}$$

and

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_{1\cdot} \\ \mathbf{b}_{2\cdot} \\ \vdots \\ \mathbf{b}_{n\cdot} \end{bmatrix} \tag{6.11}$$

Then it can be seen that the product $\mathbf{AB}$ can be expressed as sum of outer products of columns of $\mathbf{A}$ and rows of $\mathbf{B}$, i.e.,

$$\mathbf{AB} = \mathbf{a}_1 \mathbf{b}_{1\cdot} + \mathbf{a}_2 \mathbf{b}_{2\cdot} + \dots + \mathbf{a}_n \mathbf{b}_{n\cdot}$$

where each of the summands are matrices of the size $m \times p$.

6.2 Other Matrix Operations

6.2.1 Trace

The matrix trace, denoted as $\mathrm{tr}(\mathbf{A})$, is the sum of the diagonal elements of a square matrix. In other words, it represents the sum of the elements on the main diagonal of the matrix. Mathematically, the trace is expressed as:

$$\mathrm{tr}(\mathbf{A}) = \sum_{i=1}^{n} A_{ii}$$

For example, consider the matrix: $\mathbf{A} = \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix}$, whose trace is given by $\mathrm{tr}(\mathbf{A}) = 2 + 3 = 5$. The trace is linear, since $\mathrm{tr}(\alpha\mathbf{A}) = \alpha\mathrm{tr}(\mathbf{A})$ and $\mathrm{tr}(\mathbf{A} + \mathbf{B}) = \mathrm{tr}(\mathbf{A}) + \mathrm{tr}(\mathbf{B})$. We have the following important property of trace.

Lemma 6.2. For matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times m}$, we have that

$$\mathrm{tr}\left(\mathbf{AB}\right) = \mathrm{tr}\left(\mathbf{BA}\right). \tag{6.12}$$

Proof: We need to expand the expression for trace as follows.

$$[\mathbf{AB}]_{ii} = \sum_{j=1}^{n} A_{ij} B_{ji} \tag{6.13}$$

so that

$$\operatorname{tr}(\mathbf{AB}) = \sum_{i=1}^{m} [\mathbf{AB}]_{ii} = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} B_{ji} \tag{6.14}$$

$$= \sum_{j=1}^{n} \sum_{i=1}^{m} B_{ji} A_{ij} = \sum_{j=1}^{n} [\mathbf{BA}]_{jj} = \operatorname{tr}(\mathbf{BA}) \tag{6.15}$$

■

From this property, we can see that

$$\operatorname{tr}(\mathbf{ABC}) = \operatorname{tr}(\mathbf{A}(\mathbf{BC})) = \operatorname{tr}(\mathbf{BCA}) = \operatorname{tr}(\mathbf{B}(\mathbf{CA})) = \operatorname{tr}(\mathbf{CAB}) \tag{6.16}$$

which can be extended for arbitrary number of matrices in the same way.

6.2.2 Hadamard Product

Another important operation involving matrices is the Hadamard product, also known as the element-wise or entry-wise product.

Given two matrices, let's say $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$, the Hadamard product denoted as $\mathbf{A} \odot \mathbf{B}$, is obtained by multiplying corresponding entries of the matrices together. In other words, each entry of the resulting matrix is the product of the corresponding entries from $\mathbf{A}$ and $\mathbf{B}$. This operation is defined as follows:

$$(\mathbf{A} \odot \mathbf{B})_{ij} = \mathbf{A}_{ij} \mathbf{B}_{ij}$$

For example, consider the following matrices:

$$\mathbf{A} = \begin{bmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}. \tag{6.17}$$

Their Hadamard product, $\mathbf{A} \odot \mathbf{B}$ is given by

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} 2 \cdot 1 & 4 \cdot 2 & 6 \cdot 3 \\ 1 \cdot 4 & 3 \cdot 5 & 5 \cdot 6 \end{bmatrix} = \begin{bmatrix} 2 & 8 & 18 \\ 4 & 15 & 30 \end{bmatrix} \tag{6.18}$$

The Hadamard product possesses several notable properties:

1. *Commutativity:* Unlike matrix multiplication, the order of the matrices does not affect the result, so that $\mathbf{A} \odot \mathbf{B} = \mathbf{B} \odot \mathbf{A}$.
2. *Distributivity:* The Hadamard product distributes over matrix addition, i.e.,

   $$\mathbf{A} \odot (\mathbf{B} + \mathbf{C}) = (\mathbf{A} \odot \mathbf{B}) + (\mathbf{A} \odot \mathbf{C})$$

   where $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ are matrices of the same size.
3. *Scaling:* We have that $k \cdot (\mathbf{A} \odot \mathbf{B}) = (k \cdot \mathbf{A}) \odot \mathbf{B} = \mathbf{A} \odot (k \cdot \mathbf{B})$, where k is a scalar.

The Hadamard product is useful in various applications, such as element-wise operations in image processing, component-wise multiplication of vectors, and certain types of matrix transformations. The notation for regular matrix multiplication and Hadamard products varies across programming languages. In MATLAB, regular matrix multiplication is denoted by the `*` operator, while element-wise (Hadamard) product is denoted by the `.*` operator. In Python (when using NumPy), regular matrix multiplication is denoted by the `@` operator or the `dot()` function, while element-wise product is denoted by the `*` operator or using functions like `multiply()`.

6.3 Complexity of Matrix Operations

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

1. Since there are n diagonal elements, the flop count for computing the trace is $n - 1$ floating-point operations.
2. The Hadamard product of two $n \times n$ matrices involves multiplying each corresponding element. Since there are n^2 elements in total, the flop count for the element-wise product is n^2 floating-point operations.
3. The naive algorithm for matrix multiplication involves multiplying each element of a row from the first matrix with the corresponding element of a column from the second matrix and summing the products. Since the result is an $n \times n$ matrix, there are n^2 elements to be computed. For each element, we perform n multiplications and $n-1$ additions, resulting in a total of $2n-1$ floating-point operations. Therefore, the total flop count for matrix multiplication is approximately $n^2 \times (2n-1) = 2n^3 - n^2 \approx 2n^3$ floating-point operations. It is worth noting that more efficient algorithms exist for matrix multiplication. An example is the widely used Strassen algorithm which performs matrix multiplication in less than $n^{2.8}$ flops when n is sufficiently large. These algorithms take advantage of various mathematical properties to achieve faster multiplication, but they involve more complex computations and may not be practical for small matrices due to increased overhead.

Observe here that the matrix-matrix multiplication is the most computationally expensive operation. The flop counts for operations on mutli-dimensional arrays and various sparse matrix operations can be calculated similarly.

Table 6.1: Flop counts for matrix operations

Operation	**Flop count**
Trace	n
Hadamard Product	n^2
Matrix-Matrix Multiplication	$2n^3$