Chapter 6

Matrix multiplication

Multiplying matrices is considerably more complicated than multiplying regular numbers. The normal rules you know about multiplication (e.g., $a \times b = b \times a$) often don't apply to matrices, and there are extra rules you will need to learn. Furthermore, there are several ways of multiplying matrices. And to make matters more complicated, not all pairs of matrices can be multiplied. So, take a deep breath, because this chapter is your first deep-dive into linear algebra!

6.1 "Standard" matrix multiplication

For lack of a better term, this method will be called the "standard" method. Unless otherwise explicitly stated, you can assume (in this book and elsewhere) that two matrices next to each other (like this: $\mathbf{AB}$) indicates "standard" matrix multiplication.

Terminology The first thing to know about matrix multiplication is that it is not commutative, so $\mathbf{AB} \neq \mathbf{BA}$. There are exceptions where this is the case (for example, $\mathbf{AI} = \mathbf{IA}$), but those exceptions are rare. For this reason, even the terminology of matrix multiplication is complicated.

The following five statements are ways to say the operation $\mathbf{AB}$ out loud (e.g., when trying to show off to your math-inclined friends and family):

"$\mathbf{A}$ times $\mathbf{B}$"
"$\mathbf{A}$ left-multiplies $\mathbf{B}$"
"$\mathbf{A}$ pre-multiplies $\mathbf{B}$"
"$\mathbf{B}$ right-multiplies $\mathbf{A}$"
"$\mathbf{B}$ post-multiplies $\mathbf{A}$"

Validity Before learning how standard matrix multiplication works, you need to learn when matrix multiplication *is valid*. The rule for multiplication validity is simple and visual, and you need to memorize this rule before learning the mechanics of multiplication.

If you write the matrix sizes underneath the matrices, then matrix multiplication is valid only when the two "inner dimensions" match, and the size of the resulting product matrix is given by the "outer dimensions." By "inner" and "outer" I'm referring to the spatial organization of the matrix sizes, as in Figure 6.1.

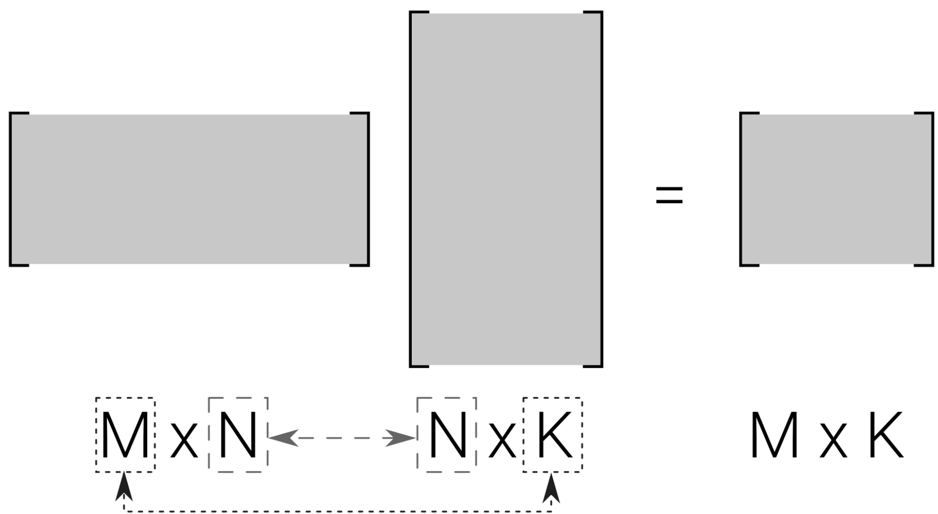

Figure 6.1: Visualization of the rule for matrix multiplication validity. Note the reference to "inner dimensions" (N) and "outer dimensions" (M and K).

Consider the following matrices; are these multiplications valid?

$$\underset{5\text{x}2}{\mathbf{A}}\,\underset{2\text{x}7}{\mathbf{B}} \qquad \underset{2\text{x}7}{\mathbf{B}}\,\underset{5\text{x}2}{\mathbf{A}} \qquad \underset{5\text{x}7}{\mathbf{C}^{\mathbf{T}}}\,\underset{5\text{x}2}{\mathbf{A}}$$

The first pair (**AB**) is valid because the "inner" dimensions match (both 2). The resulting product matrix will be of size 5×7. The second pair shows the lack of commutativity in matrix multiplication: The "inner" dimensions (7 and 5) do not match, and thus the multiplication is not valid. The third pair is an interesting case. You might be tempted to call this an invalid operation; however, when transposing **C**, the rows and columns swap, and so the "inner" dimensions become consistent (both 5). So this multiplication is valid.

Here's something exciting: you are now armed with the knowledge to understand the notation for the dot product and outer product. In particular, you can now appreciate why the order of transposition ($\mathbf{v}^{\mathrm{T}}\mathbf{w}$ or $\mathbf{v}\mathbf{w}^{\mathrm{T}}$) determines whether the multiplication is the dot product or the outer product (Figure 6.2).

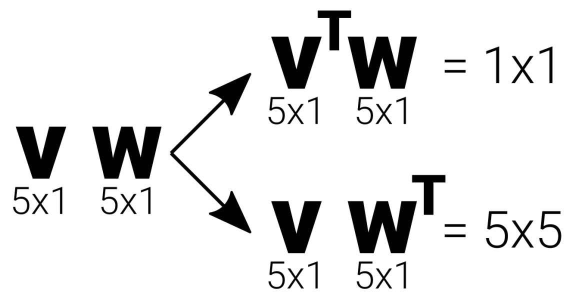

Figure 6.2: For two adjacent column vectors, transposing the first vs. the second vector is the difference between the dot product vs. the outer product.

Practice problems For the following matrices, determine whether matrix multiplication is valid, and, if so, the size of the product matrix.

$$\mathbf{A} \in \mathbb{R}^{2\times 4}, \quad \mathbf{B} \in \mathbb{R}^{3\times 4}, \quad \mathbf{C} \in \mathbb{R}^{4\times 4}$$

a) $\mathbf{AB}$ **b)** $\mathbf{AC}$ **c)** $\mathbf{AB}^{\mathrm{T}}$ **d)** $\mathbf{BCA}^{\mathrm{T}}$ **e)** $\mathbf{BB}^{\mathrm{T}}\mathbf{A}$

Answers

a) no **b)** 2×4 **c)** 2×3 **d)** 3×2 **e)** no

Code Unfortunately, matrix multiplication is confusingly different between MATLAB and Python. Pay close attention to the subtle but important differences (`@` vs. `*`).

Code block 6.1: Python

```
M1 = np.random.randn(4,3)
M2 = np.random.randn(3,5)
C = M1 @ M2
```

Code block 6.2: MATLAB

```
M1 = randn(4,3);
M2 = randn(3,5);
C = M1 * M2
```

It is now time to learn how to multiply matrices. There are four ways to think about and implement matrix multiplication. All four methods give the same result, but provide different perspectives on what matrix multiplication means. It's useful to understand all of these perspectives, because they provide insights into matrix computations in different contexts and for different problems. It is unfortunate that many linear algebra textbooks teach only the dot-product method (what I call the "element perspective").

(1) The "element perspective" Each element $c_{i,j}$ in $\mathbf{AB} = \mathbf{C}$ is the dot product between the i^{th} row in $\mathbf{A}$ and the j^{th} column in $\mathbf{B}$. The equation below shows how you would create the top-left element in the product matrix.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1a + 2c & \\ & \end{bmatrix} \tag{6.1}$$

For convenience, I will label all matrices in the following examples as $\mathbf{AB} = \mathbf{C}$.

This makes it clear that element $c_{i,j}$ comes from the dot product between row $\mathbf{a}_i$ and column $\mathbf{b}_j$.

Here is another example; make sure you see how each element in the product matrix is the dot product between the corresponding row in the left matrix and column in the right matrix.

$$\begin{bmatrix} 3 & 4 \\ -1 & 2 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 5 & 1 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 3{\cdot}5 + 4{\cdot}3 & 3{\cdot}1 + 4{\cdot}1 \\ -1{\cdot}5 + 2{\cdot}3 & -1{\cdot}1 + 2{\cdot}1 \\ 0{\cdot}5 + 4{\cdot}3 & 0{\cdot}1 + 4{\cdot}1 \end{bmatrix} = \begin{bmatrix} 27 & 7 \\ 1 & 1 \\ 12 & 4 \end{bmatrix}$$

There is a hand gesture that you can apply to remember this rule: extend your pointer fingers of both hands; simultaneously move your left hand from left to right (across the row of the left matrix) while moving your right hand down towards you (down the column of the right matrix) (Figure 6.3).

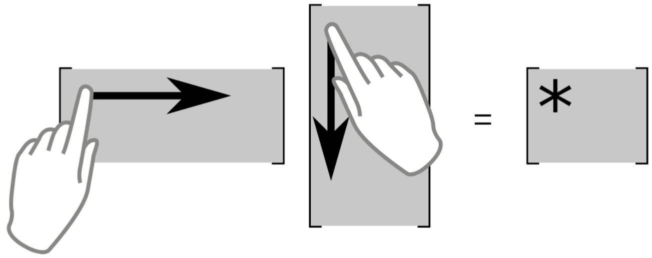

Figure 6.3: Visual representation of the mechanism of computing each element of the matrix multiplication product as the vector dot product between each row of the left matrix (from left-to-right and each column of the right matrix (from top-to-bottom).

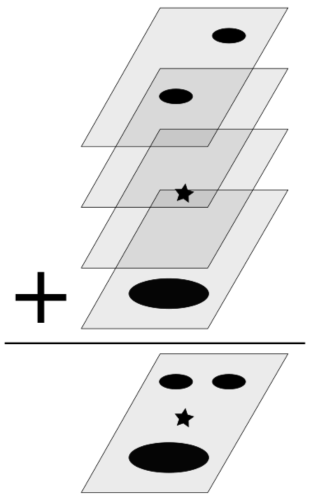

Figure 6.4: A simulacrum of building up a matrix one layer at a time. Each layer is the same size as the product yet provides only partial information.

Below you can see another visualization of matrix multiplication from the element perspective. This visualization facilitates three important features of matrix multiplication.

$$\begin{bmatrix} — & a_1 & — \\ — & a_2 & — \\ & \vdots & \\ — & a_n & — \end{bmatrix} \begin{bmatrix} | & | & & | \\ b_1 & b_2 & \cdots & b_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & \cdots & a_1 \cdot b_n \\ a_2 \cdot b_1 & a_2 \cdot b_2 & & a_2 \cdot b_n \\ \vdots & & \ddots & \vdots \\ a_n \cdot b_1 & a_n \cdot b_2 & \cdots & a_n \cdot b_n \end{bmatrix}$$

- The diagonal of the product matrix **C** contains dot products between rows and columns of the same ordinal position (row i in **A** and column i in **B**).
- The lower-triangle of **C** contains dot products between *later* rows in **A** and *earlier* columns in **B** (row i in **A** and column j in **B**, where $i > j$).
- The upper-triangle of **C** contains dot products between *earlier* rows in **A** and *later* columns in **B** (row i in **A** and column j in **B**, where $i < j$).

The first point above is relevant for understanding data covariance matrices. The second and third points are important for understanding several matrix decompositions, most importantly QR decomposition and generalized eigendecomposition.

(2) The "layer perspective" In contrast to the element perspective, in which each element is computed independently of each other element, the layer perspective involves conceptualizing the product matrix as a series of layers, or "sheets," that are summed together. This is implemented by creating outer products from the columns of $\mathbf{A}$ and the rows of $\mathbf{B}$, and then summing those outer products together.

Remember that the outer product is a matrix. Each outer product is the same size as $\mathbf{C}$, and can be thought of as a layer. By analogy, imagine constructing an image by laying transparent sheets of paper on top of each other, with each sheet containing a different part of the image (Figure 6.4).

Below is an example using the same matrix as in the previous section. Make sure you understand how the two outer product matrices are formed from column $\mathbf{a}_i$ and row $\mathbf{b}_j$. You can also use nearly the same hand gesture as with the element perspective (Figure 6.3), but swap the motions of the left and right hands.

$$\begin{bmatrix} 3 & 4 \\ -1 & 2 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 5 & 1 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 15 & 3 \\ -5 & -1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 12 & 4 \\ 6 & 2 \\ 12 & 4 \end{bmatrix} = \begin{bmatrix} 27 & 7 \\ 1 & 1 \\ 12 & 4 \end{bmatrix}$$

Each of these layers is a rank-1 matrix. Rank will be discussed in more detail in a separate chapter, but for now, you can think of a rank-1 matrix as containing only a single column's worth of information; all the other columns are scaled versions.

Notice that in each of the "layer matrices," the columns form a dependent set (the same can be said of the rows). However, the sum of these singular matrices—the product matrix—has columns that form a linearly *in*dependent set.

The layer perspective of matrix multiplication is closely related to the spectral theorem of matrices, which says that any matrix can be represented as a sum of rank-1 matrices. It's like each rank-1 matrix is a single color and the matrix is the rainbow. This important and elegant idea is the basis for the singular value decomposition, which you will learn about in Chapter 16.

(3) The "column perspective" From the column perspective, all matrices (the multiplying matrices and the product matrix) are thought of as sets of column vectors. Then the product matrix is created one column at a time.

The first column in the product matrix is a linear weighted combination of all columns in the left matrix, where the weights are defined by the elements in the first column of the right matrix. The second column in the product matrix is again a weighted combination of all columns in the left matrix, except that the weights now come from the second column in the right matrix. And so on for all N columns in the right matrix. Let's start with a simple example:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a \begin{bmatrix} 1 \\ 3 \end{bmatrix} + c \begin{bmatrix} 2 \\ 4 \end{bmatrix} & & b \begin{bmatrix} 1 \\ 3 \end{bmatrix} + d \begin{bmatrix} 2 \\ 4 \end{bmatrix} \end{bmatrix} \tag{6.2}$$

Let's go through Equation 6.2 slowly. The first column of the product matrix is the sum of all columns in matrix $\mathbf{A}$. But it's not just the columns added together—each column in $\mathbf{A}$ is weighted according to the corresponding element from the first column of matrix $\mathbf{B}$. Then, the second column of matrix $\mathbf{C}$ is created by again summing all of the columns in matrix $\mathbf{A}$, except now each column is weighted by a different element of column 2 from matrix $\mathbf{B}$. Equation 6.2 shows only two columns, but this procedure would be repeated for however many columns are in matrix $\mathbf{B}$.

Now for the same numerical example you've seen in the previous two perspectives:

$$\begin{bmatrix} 3 & 4 \\ -1 & 2 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 5 & 1 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 5 \begin{bmatrix} 3 \\ -1 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 4 \\ 2 \\ 4 \end{bmatrix} & & 1 \begin{bmatrix} 3 \\ -1 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 4 \\ 2 \\ 4 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 27 & 7 \\ 1 & 1 \\ 12 & 4 \end{bmatrix}$$

The column perspective of matrix multiplication is useful in statistics, when the columns of the left matrix contain a set of *regressors* (a simplified model of the data), and the right matrix contains *coefficients*. The coefficients encode the importance of each regressor, and the goal of statistical model-fitting is to find the best coefficients such that the weighted combination of regressors matches the data. More on this in Chapter 14!

(4) The "row perspective" You guessed it—it's the same concept as the column perspective but you build up the product matrix one row at a time, and everything is done by taking weighted combinations of rows. Thus: each *row* in the product matrix is the weighted sum of all *rows* in the right matrix, where the weights are given by the elements in each *row* of the left matrix. Let's begin with the simple example:

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}a & b\\c & d\end{bmatrix} = \begin{bmatrix}1\begin{bmatrix}a & b\end{bmatrix} + 2\begin{bmatrix}c & d\end{bmatrix}\\ \\ 3\begin{bmatrix}a & b\end{bmatrix} + 4\begin{bmatrix}c & d\end{bmatrix}\end{bmatrix} \tag{6.3}$$

The top row of the product matrix is created by summing together the two rows of the right matrix, but each row is weighted according to the corresponding element of the top row of the left matrix. Same story for the second row. And of course, this would continue for however many rows are in the left matrix.

I won't repeat the other example multiplication I've been showing in previous pages; that's for you to do on your own with pencil and paper. (Hint: The result will be identical.)

The row perspective is useful, for example in principal components analysis, where the rows of the right matrix contain data (observations in rows and features in columns) and the rows of the left matrix contain weights for combining the features. Then the weighted sum of data creates the principal component scores. Figure 6.5 visually summarizes the different perspectives.

Practice problems Multiply the following pairs of matrices four times, using each of four perspectives. Make sure you get the same result each time.

a) $\begin{bmatrix}3 & 0 & 3\\1 & 1 & 0\end{bmatrix}\begin{bmatrix}4 & 4\\0 & 1\\4 & 1\end{bmatrix}$

b) $\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}\begin{bmatrix}a & b & c\\d & e & f\\g & h & i\end{bmatrix}$

Answers

a) $\begin{bmatrix}24 & 15\\4 & 5\end{bmatrix}$

b) $\begin{bmatrix}1a+2d+3g & 1b+2e+3h & 1c+2f+3i\\4a+5d+6g & 4b+5e+6h & 4c+5f+6i\\7a+8d+9g & 7b+8e+9h & 7c+8f+9i\end{bmatrix}$

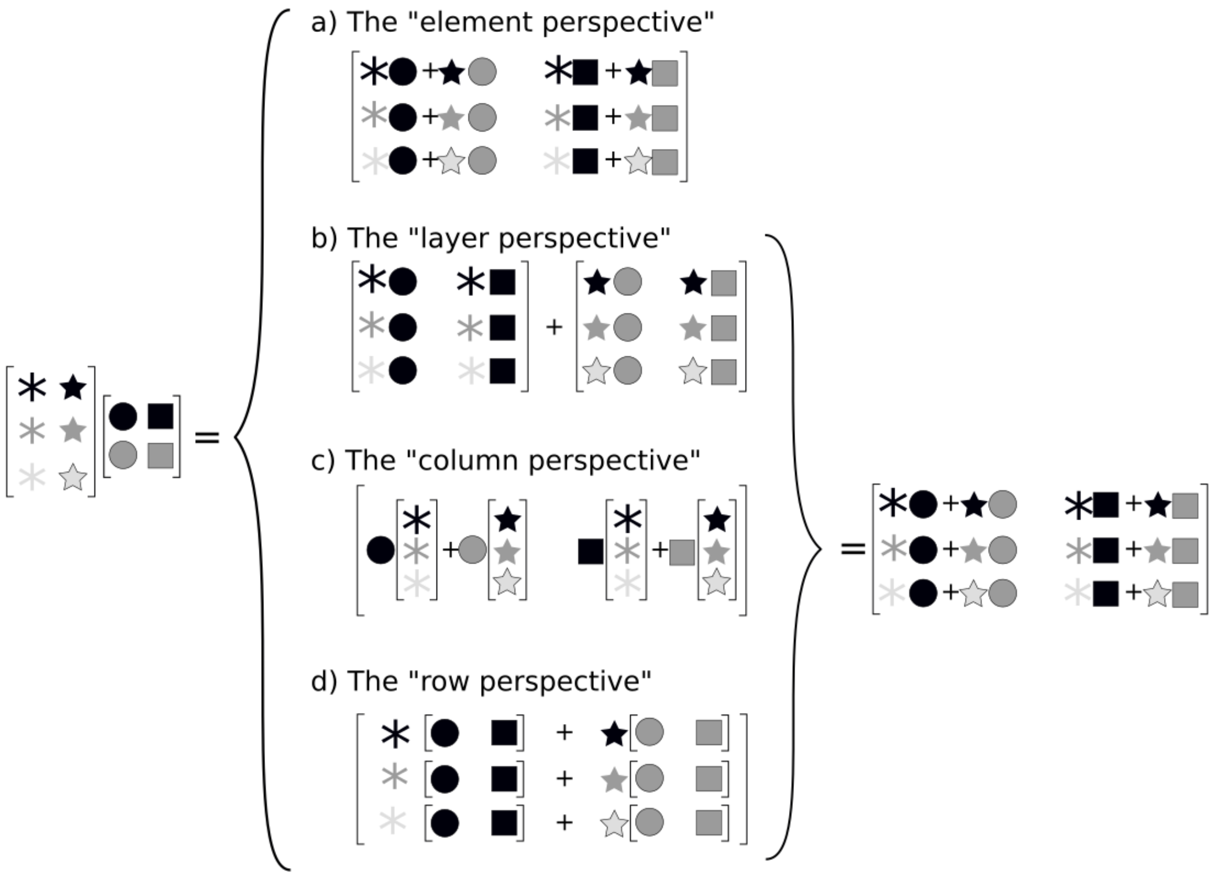

Figure 6.5: Visual representation of the four perspectives on matrix multiplication.

Practice problems Perform the following matrix multiplications. Use whichever perspective you find most confusing (that's the perspective you need to practice!).

a) $\begin{bmatrix} 3 & 4 & 0 \\ 0 & 4 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 2 \\ 4 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$

b) $\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix}$

Answers

a) $\begin{bmatrix} 22 & -3 & 6 \\ 16 & 1 & 1 \end{bmatrix}$

b) $\begin{bmatrix} 3 & 3 \\ 3 & 4 \end{bmatrix}$

6.2 Multiplication and equations

You know from your high-school algebra course that you are allowed to add and multiply terms to an equation, as long as you apply that operation to both sides. For example, I can multiply both sides of this equation by 7:

$$4 + x = 5(y + 3)$$

$$7(4 + x) = 7(5(y + 3))$$

$$7(4 + x) = (5(y + 3))7$$

Notice that the bottom two equations are the same: Because scalars obey the commutative law, the 7 can go on the left or the right of the parenthetic term.

However, because matrix multiplication is not commutative, you need to be mindful to put the matrices on the same side of the equation. For example, the following is OK:

$$\mathbf{B} = \lambda(\mathbf{C} + \mathbf{D})$$

$$\mathbf{AB} = \lambda\mathbf{A}(\mathbf{C} + \mathbf{D})$$

$$\mathbf{AB} = \mathbf{A}(\mathbf{C} + \mathbf{D})\lambda$$

Matrix sizes are not stated, so assume that the sizes make the operations valid.

The λ can be moved around because it is a scalar, but $\mathbf{A}$ must pre-multiply both sides (or post-multiply both sides, but it must be consistent on the left- and right-hand sides of the equation). In contrast to the above, the following progression of equations is **WRONG**.

$$\mathbf{B} = \lambda(\mathbf{C} + \mathbf{D})$$

$$\mathbf{AB} = \lambda(\mathbf{C} + \mathbf{D})\mathbf{A}$$

In other words, if you pre-multiply on one side of an equation, you must pre-multiply on the other side of the equation. Same goes for post-multiplication. If there are rectangular matrices in the equation, it is possible that pre- or post-multiplying isn't even valid.

Code The non-commutivity of matrix multiplication is easy to confirm in code. Compare matrices C1 and C2.

Code block 6.3: Python

```
A = np.random.randn(2,2)
B = np.random.randn(2,2)
C1 = A@B
C2 = B@A
```

Code block 6.4: MATLAB

```
A = randn(2,2);
B = randn(2,2);
C1 = A*B;
C2 = B*A;
```

6.3 Matrix multiplication with a diagonal matrix

There is a special property of multiplication when one matrix is a diagonal matrix and the other is a dense matrix:

- Pre-multiplication by a diagonal matrix scales the *rows* of the right matrix by the diagonal elements.
- Post-multiplication by a diagonal matrix scales the *columns* of the left matrix by the diagonal elements.

Let's see two examples of 3×3 matrices; notice how the diagonal elements appear in the rows (pre-multiply) or the columns (post-multiply). Also notice how the product matrix is the same as the dense matrix, but either the columns or the rows are scaled by each corresponding diagonal element of the diagonal matrix.

$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 1a & 2a & 3a \\ 4b & 5b & 6b \\ 7c & 8c & 9c \end{bmatrix} \tag{6.4}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} = \begin{bmatrix} 1a & 2b & 3c \\ 4a & 5b & 6c \\ 7a & 8b & 9c \end{bmatrix} \tag{6.5}$$

Of course, the mechanism of matrix multiplication is exactly the same as what you learned in the previous section. But all the zeros in the diagonal matrices allow us to simplify things a bit.

It is worth remembering this property of diagonal matrix multiplication, because you will see applications of this in several subsequent chapters, including systems of equations (Chapter 10), diagonalization (Chapter 15), and singular-value decomposition (Chapter 16). I tried to come up with an easy mnemonic for remembering this rule. It is, admittedly, not such a great mnemonic, but you should take this as a challenge to come up with a better one.

Order of matrices for modulating rows vs. columns

P**R**e-multiply to affect **R**ows

P**O**st-multiply to affect c**O**lumns.

Practice problems Perform the following matrix multiplications. Note the differences between **a** and **c**, and between **b** and **d**.

a) $\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 5 & 0 \\ 0 & 6 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}$ **c)** $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$ **d)** $\begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 0 \\ 0 & 6 \end{bmatrix}$

Answers

a) $\begin{bmatrix} 2 & 2 \\ 3 & 3 \end{bmatrix}$ **b)** $\begin{bmatrix} 20 & 15 \\ 18 & 24 \end{bmatrix}$ **c)** $\begin{bmatrix} 2 & 3 \\ 2 & 3 \end{bmatrix}$ **d)** $\begin{bmatrix} 20 & 18 \\ 15 & 24 \end{bmatrix}$

Multiplying two diagonal matrices The product of two diagonal matrices is another diagonal matrix whose diagonal elements are the products of the corresponding diagonal elements. That's a long sentence, but it's a simple concept. Here's an example:

$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} d & 0 & 0 \\ 0 & e & 0 \\ 0 & 0 & f \end{bmatrix} = \begin{bmatrix} ad & 0 & 0 \\ 0 & be & 0 \\ 0 & 0 & cf \end{bmatrix} \tag{6.6}$$

Take a minute to work through the mechanics of matrix multiplication to convince yourself that this is the correct answer. And then you can just remember that multiplying two diagonal matrices is easy—in fact, for two diagonal matrices, standard matrix multiplication is the same as element-wise multiplication. That becomes relevant when learning about eigendecomposition!

6.4 LIVE EVIL! (a.k.a. order of operations)

Let's start with an example to highlight the problem that we need a solution for. Implement the following matrix multiplication and then transpose the result:

$$\left(\begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \right)^{\mathrm{T}} = \textbf{?}$$

I assume you got the matrix $\begin{bmatrix} 4 & 2 \\ 6 & 3 \end{bmatrix}$. Now let's try it again, but this time transpose each matrix individually before multiplying them:

$$\begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^{\mathrm{T}} = \textbf{?}$$

Did you get the same matrix as above? Well, if you did the math correctly, then you will have gotten $\begin{bmatrix} 1 & 3 \\ 2 & 6 \end{bmatrix}$, which is the not the same as the previous result. It's not even the same result but transposed. In fact, it is an entirely different matrix.

OK, now let's try it one more time. But now, *swap the order of the matrices* before applying the transpose operation to each individual matrix. Thus:

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix}^{\mathrm{T}} \quad = \quad \mathbf{?}$$

And now you get the same result as the first multiplication: $\begin{bmatrix} 4 & 2 \\ 6 & 3 \end{bmatrix}$.

And this brings us to the main topic of this section: An operation applied to multiplied matrices gets applied to each matrix individually *but in reverse order*.

It's a weird rule, but that's just how it works. **"LIVE EVIL"** is a mnemonic that will help you remember this important rule. Notice that LIVE spelled backwards is EVIL. It's a palindrome.

n.b.: LIVE EVIL is *not* a recommendation for how to interact with society. Please be nice, considerate, and generous.

The LIVE EVIL rule: Reverse matrix order

$$(\mathbf{A}\ldots\mathbf{B})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\ldots\mathbf{A}^{\mathrm{T}} \tag{6.7}$$

Basically, an operation on multiplied matrices gets applied to each matrix in reverse order. Here's how it would look for four matrices:

$$(\mathbf{ABCD})^{\mathrm{T}} = \mathbf{D}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$$

The LIVE EVIL rule applies to other operations as well, such as the matrix inverse. For example:

$$(\mathbf{ABC})^{-1} = \mathbf{C}^{-1}\mathbf{B}^{-1}\mathbf{A}^{-1}$$

LIVE EVIL is, admittedly, a strange and counter-intuitive rule. But it is important, so let's look at another example, which will also highlight why this rule needs to be the case. In the equations below, the top row implements matrix multiplication first, then transposes the result; while the second row first transposes each matrix in reverse order and then performs the matrix multiplication.

$$\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix}\right)^{\mathrm{T}} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} ae+bg & ce+dg \\ af+bh & cf+dh \end{bmatrix}$$

$$= \begin{bmatrix} e & g \\ f & h \end{bmatrix}\begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} ae+bg & ce+dg \\ af+bh & cf+dh \end{bmatrix}$$

For square matrices, ignoring the LIVE EVIL rule still gives a result (though incorrect). However, for rectangular matrices, multiplication would be impossible when ignoring the LIVE EVIL rule. This is illustrated in Figure 6.6.

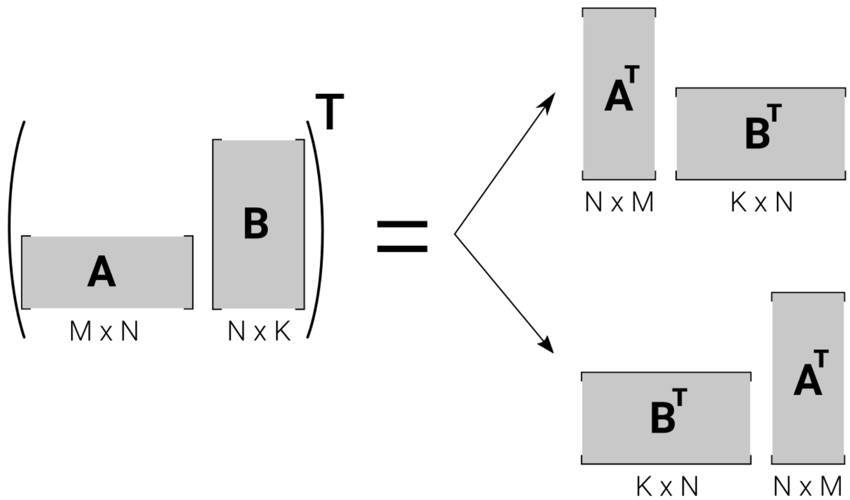

Figure 6.6: Example of the LIVE EVIL law for transposing matrix multiplication. Pay attention to the matrix sizes: Ignoring the LIVE EVIL rule and transposing without reversing leads to an invalid expression (top), whereas the multiplication remains valid when swapping matrix order (bottom).

Be careful with Equation 6.7—an operation may be valid on the product of matrices but invalid on the individual matrices. This comes up frequently in statistics and the singular value decomposition. For example, the expression $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$ may be valid whereas $\mathbf{X}^{-1}\mathbf{X}^{\text{-T}}$ may be undefined. You'll see examples of this situation in later chapters. Fortunately, all matrices can be transposed, so for now you can always apply the LIVE EVIL rule without concern.

Practice problems Perform the following matrix multiplications. Compare with the problems and answers on page 148.

a) $\begin{bmatrix} 2 & -1 & 2 \\ 4 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 3 & 4 & 0 \\ 0 & 4 & 1 \end{bmatrix}^{\mathrm{T}}$ **b)** $\begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^{\mathrm{T}}$

Answers

a) $\begin{bmatrix} 22 & 16 \\ -3 & 1 \\ 6 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 3 & 3 \\ 3 & 4 \end{bmatrix}$

6.5 Matrix-vector multiplication

Matrix-vector multiplication is the same thing as normal matrix-matrix multiplication, where you think of the vector as an $M \times 1$ or as a $1 \times N$ matrix. The important feature of matrix-vector multiplication—which is obvious when you think about it but is worth mentioning explicitly—is that the result is always a vector. This is important because it provides the connection between linear transformations and matrices: To apply a transform to a vector, you convert that transform into a matrix, and then you multiply the vector by that matrix. Here are two examples of matrix-vector multiplication.

$$\mathbf{Ab} = \begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \end{bmatrix} = \begin{bmatrix} 4{\cdot}5 + 2{\cdot}2 \\ 1{\cdot}5 + 3{\cdot}2 \end{bmatrix} = \begin{bmatrix} 24 \\ 11 \end{bmatrix}$$

$$\mathbf{b}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} 5 & 2 \end{bmatrix} \begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 4{\cdot}5 + 1{\cdot}2 & 5{\cdot}2 + 2{\cdot}3 \end{bmatrix} = \begin{bmatrix} 24 & 16 \end{bmatrix}$$

Three observations here:

1. $\mathbf{bA}$ is not defined (assuming $\mathbf{b}$ is a column vector)
2. If $\mathbf{A}$ is rectangular, then either $\mathbf{b}^{\mathrm{T}}\mathbf{A}$ or $\mathbf{Ab}$ is undefined (depending on the sizes, but they can't both be valid)
3. $\mathbf{Ab} \neq \mathbf{b}^{\mathrm{T}}\mathbf{A}$ even when both are valid operations

There is an interesting exception to this third observation, which is that if the matrix is symmetric, then pre-multiplying the vector is the same as post-multiplying the transpose of the vector. (Technically, the results are not *literally the same*, because one result is a column vector while the other is a row vector, but the elements of those vectors are identical.)

Symmetric matrix times a vector

$$\textit{if } \mathbf{A} = \mathbf{A}^{\mathrm{T}} \textit{ then } \mathbf{Ab} = (\mathbf{b}^{\mathrm{T}}\mathbf{A})^{\mathrm{T}} \tag{6.8}$$

Let's work through a proof of this claim. The proof works by transposing $\mathbf{Ab}$ and doing a bit of algebra (including applying the LIVE

EVIL rule!) to simplify and re-arrange.

$$(\mathbf{Ab})^{\mathrm{T}} = \mathbf{b}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} = \mathbf{b}^{\mathrm{T}}\mathbf{A}$$

Notice that our proof here involved transposing, expanding, and simplifying. This strategy underlies many linear algebra proofs.

The proof works because $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$. If the matrix weren't symmetric, then $\mathbf{A} \neq \mathbf{A}^{\mathrm{T}}$, in other words, $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$ would be different matrices. And of course, $\mathbf{b}$ and $\mathbf{b}^{\mathrm{T}}$ are the same except for orientation.

Let's look at an example.

$$\mathbf{Ab} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} d \\ e \end{bmatrix} = \begin{bmatrix} ad + be \\ bd + ce \end{bmatrix}$$

$$\mathbf{b}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} d & e \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} = \begin{bmatrix} ad + be & bd + ce \end{bmatrix}$$

Notice, as mentioned above, that the two results are not *identical vectors* because one is a column and the other is a row. However, they have *identical elements* in a different orientation.

Now watch what happens when matrix $\mathbf{A}$ is non-symmetric. In the matrices below, assume $b \neq f$.

$$\mathbf{Ab} = \begin{bmatrix} a & b \\ f & c \end{bmatrix} \begin{bmatrix} d \\ e \end{bmatrix} = \begin{bmatrix} ad + be \\ fd + ce \end{bmatrix}$$

$$\mathbf{b}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} d & e \end{bmatrix} \begin{bmatrix} a & b \\ f & c \end{bmatrix} = \begin{bmatrix} ad + fe & bd + ce \end{bmatrix}$$

Practice problems Perform the following matrix multiplications.

a) $\begin{bmatrix} 4 & 3 & 0 \\ 1 & 1 & 9 \\ 0 & 3 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 7 \\ 1 \end{bmatrix}$

b) $\begin{bmatrix} 4 & 0 & 2 & 0 & 8 & 0 \\ 0 & 3 & 0 & 5 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 6 \\ 1 \\ 2 \\ 1 \\ 6 \end{bmatrix}$

Answers

a) $\begin{bmatrix} 21 \\ 16 \\ 21 \\ 8 \end{bmatrix}$

b) $\begin{bmatrix} 26 \\ 34 \end{bmatrix}$

6.6 Creating symmetric matrices

In the previous chapter, I claimed that all hope is not lost for non-symmetric matrices who aspire to the glorious status of their symmetric conspecifics, upon whom so many luxuries of linear algebra are bestowed. How can a non-symmetric matrix become a symmetric matrix?

There are two methods: Additive and multiplicative. The additive method is not widely used in practical applications (to my knowledge), but it is worth learning. The additive method to create a symmetric matrix from a non-symmetric matrix is to add the matrix to its transpose. This method is valid only for square matrices.

$$\mathbf{C} = \frac{1}{2}(\mathbf{A}^{\mathrm{T}} + \mathbf{A}) \tag{6.9}$$

An example will illustrate why Equation 6.9 works. Notice that the diagonal elements are doubled, which is why dividing by 2 is an appropriate normalization factor. (In the matrices below, assume that $b \neq d$, $c \neq h$, and $f \neq i$.)

$$\begin{bmatrix} a & b & c \\ d & e & f \\ h & i & j \end{bmatrix} + \begin{bmatrix} a & d & h \\ b & e & i \\ c & f & j \end{bmatrix} = \begin{bmatrix} a+a & b+d & c+h \\ b+d & e+e & f+i \\ c+h & f+i & j+j \end{bmatrix} \tag{6.10}$$

This is just one example of a 3×3 matrix. What if this is some quirk of this particular matrix? How do we know that this method will always work? This is an important question, because there are several special properties of 2×2 or 3×3 matrices that do not generalize to larger matrices.

The proof that Equation 6.9 will always produce a symmetric matrix from a non-symmetric square matrix comes from the definition of symmetric matrices ($\mathbf{C} = \mathbf{C}^{\mathrm{T}}$). Therefore, the proof works by transposing both sides of Equation 6.9, doing a bit of algebra to simplify, and seeing what happens (I'm omitting the scalar division by 2 because that doesn't affect symmetry).

The matrices are summed, not multiplied, so the LIVE EVIL rule does not apply.

$$\mathbf{C} = \mathbf{A}^{\mathrm{T}} + \mathbf{A} \tag{6.11}$$

$$\mathbf{C}^{\mathrm{T}} = (\mathbf{A}^{\mathrm{T}} + \mathbf{A})^{\mathrm{T}} \tag{6.12}$$

$$\mathbf{C}^{\mathrm{T}} = \mathbf{A}^{\mathrm{TT}} + \mathbf{A}^{\mathrm{T}} \tag{6.13}$$

$$\mathbf{C}^{\mathrm{T}} = \mathbf{A} + \mathbf{A}^{\mathrm{T}} \tag{6.14}$$

Because matrix addition is commutative, $\mathbf{A} + \mathbf{A}^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}} + \mathbf{A}$. The right-hand side of Equation 6.14 is the same as the right-hand side of Equation 6.11. And if the right-hand sides of the equations are equal, then the left-hand sides must also be equal. This proves that $\mathbf{C} = \mathbf{C}^{\mathrm{T}}$, which is the definition of a symmetric matrix. This proof does not depend on the size of the matrix, which shows that our example above was not a fluke.

Practice problems Create symmetric matrices from the following matrices using the additive method.

a) $\begin{bmatrix} 1 & 11 & 1 \\ -5 & -2 & 1 \\ -1 & -5 & 2 \end{bmatrix}$

b) $\begin{bmatrix} -3 & 0 & -2 & -2 \\ -1 & -3 & -2 & -6 \\ -4 & -8 & -7 & 4 \\ 6 & -2 & 5 & 4 \end{bmatrix}$

Answers

a) $\begin{bmatrix} 1 & 3 & 0 \\ 3 & -2 & -2 \\ 0 & -2 & 2 \end{bmatrix}$

b) $\begin{bmatrix} -3 & -\frac{1}{2} & -3 & 2 \\ -\frac{1}{2} & -3 & -5 & -4 \\ -3 & -5 & -7 & \frac{9}{2} \\ 2 & -4 & \frac{9}{2} & 4 \end{bmatrix}$

The multiplicative method This involves multiplying a matrix by its transpose. In fact, this is the $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ matrix that you learned about in the previous chapter.

I claimed in the previous chapter that $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is guaranteed to be symmetric (and therefore also square), even if $\mathbf{A}$ is non-symmetric—and even if $\mathbf{A}$ is non-square. Now that you know about matrix multiplication and about the LIVE EVIL rule, you are able to prove these two important claims.

First we prove that $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is a square matrix. Let's say that $\mathbf{A}$ is size $M \times N$ (assume $M \neq N$). First note that $\mathbf{AA}$ is not a valid multiplication. Now consider that $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ means $(N \times M) \times (M \times N)$. The "inner" dimensions (M) match, and the result will be $N \times N$. So there you go: we just proved that $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is a square matrix.

Now let's prove that $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is symmetric. The proof follows the same strategy that we applied for the additive method: transpose $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, do a bit of algebra, and see what happens.

$$(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}}\mathbf{A}^{\mathrm{TT}} = \mathbf{A}^{\mathrm{T}}\mathbf{A} \tag{6.15}$$

We transposed the matrix, applied the LIVE EVIL rule (and the property that a double-transpose leaves the matrix unchanged), and got back to the original matrix. A matrix that equals its transpose is the definition of a symmetric matrix.

Are these two features unique to $\mathbf{A}^{\mathrm{T}}\mathbf{A}$? What about $\mathbf{AA}^{\mathrm{T}}$—is that also square and symmetric? The answer is Yes, but I want you to get a piece of paper and prove it to yourself.

Reflection

In practical applications, particularly in data analysis, $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ appears as often as $\mathbf{AA}^{\mathrm{T}}$. The correct form depends on how the data are stored (e.g., observations $\times$ features vs. features $\times$ observations), which usually depends on coding preferences or software format. In the written medium, I prefer $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ because of the aesthetic symmetry of having the $^{\mathrm{T}}$ in the middle.

Practice problems Create two symmetric matrices from each of the following matrices, using the multiplicative method ($\mathbf{A}^{\mathrm{T}}\mathbf{A}$ and $\mathbf{AA}^{\mathrm{T}}$).

a) $\begin{bmatrix} 3 & 0 & 3 \\ 0 & 7 & 0 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 & 6 \\ 6 & 1 \end{bmatrix}$

Answers

a) $\mathbf{A}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} 9 & 0 & 9 \\ 0 & 49 & 0 \\ 9 & 0 & 9 \end{bmatrix}$, $\mathbf{AA}^{\mathrm{T}} = \begin{bmatrix} 18 & 0 \\ 0 & 49 \end{bmatrix}$ **b)** $\mathbf{A}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} 37 & 12 \\ 12 & 37 \end{bmatrix}$, $\mathbf{AA}^{\mathrm{T}} = \begin{bmatrix} 37 & 12 \\ 12 & 37 \end{bmatrix}$

6.7 Multiplication of two symmetric matrices

If you multiply two symmetric matrices, will the product matrix also be symmetric? Let's try an example to find out:

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} d & e \\ e & f \end{bmatrix} = \begin{bmatrix} ad+be & ae+bf \\ bd+ce & be+cf \end{bmatrix} \tag{6.16}$$

Is the result symmetric? On its face, it seems like it isn't. However, this equation reveals an interesting condition on the symmetry of the result of multiplying two 2×2 symmetric matrices: If $a = c$ and $d = f$ (in other words, the matrix has a constant diagonal), then the product of two symmetric matrices is itself symmetric. Observe and be amazed!

$$\begin{bmatrix} a & b \\ b & a \end{bmatrix} \begin{bmatrix} d & e \\ e & d \end{bmatrix} = \begin{bmatrix} ad+be & ae+bd \\ bd+ae & be+ad \end{bmatrix} \tag{6.17}$$

Is this a general rule that applies to symmetric matrices of any size? Let's repeat with 3×3 matrices, using the constant-diagonal idea.

$$\begin{bmatrix} a & b & c \\ b & a & d \\ c & d & a \end{bmatrix} \begin{bmatrix} e & f & g \\ f & e & h \\ g & h & e \end{bmatrix} = \begin{bmatrix} ae+bf+cg & af+be+ch & ag+bh+ce \\ be+af+dg & bf+ae+dh & bg+ah+de \\ ce+df+ag & cf+de+ah & cg+dh+ae \end{bmatrix}$$

A quick glance reveals the lack of symmetry. For example, compare the element in position *2,1* with that in position *1,2*. I won't write out the product of two 4×4 symmetric matrices (if you want to try it, go for it), but you can take my word for it: the resulting product matrix will not be symmetric.

The lesson here is that, in general, *the product of two symmetric matrices is not a symmetric matrix.* There are exceptions to this rule, like the 2×2 case with constant diagonals, or if one of the matrices is the identity or zeros matrix.

Is this a surprising result? Refer back to the discussion at the end of the "element perspective" of matrix multiplication (page 144) concerning how the upper-triangle vs. the lower-triangle of the product matrix is formed from earlier vs. later rows of the left matrix. Different parts of the two multiplying matrices meet in the lower triangle vs. the upper triangle of the product matrix.

You can also see that the product matrix is not symmetric by trying to prove that it *is* symmetric (the proof fails, which is called proof-by-contradition). Assume that $\mathbf{A}$ and $\mathbf{B}$ below are both symmetric matrices.

$$(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} = \mathbf{BA} \neq \mathbf{AB} \tag{6.18}$$

To be sure, Equation 6.18 is a perfectly valid equation, and all of the individual multiplications are valid. However, matrix multiplication is not commutative, which is why we had to put a $\neq$ sign at the end of the equation. Thus, we cannot assume that $\mathbf{AB} = (\mathbf{AB})^{\mathrm{T}}$, therefore, the multiplication of two symmetric matrices is, in general, not a symmetric matrix.

Reflection

This may seem like a uselessly academic factoid, but it leads to one of the biggest limitations of principal components analysis, and one of the most important advantages of generalized eigendecomposition, which is the computational backbone of many machine-learning methods, most prominently linear classifiers and discriminant analyses.

6.8 Element-wise (Hadamard) multiplication

Before having had any exposure to matrix multiplication (in this chapter or elsewhere), Hadamard multiplication is probably what you would have answered if someone asked you to guess what it means to multiply two matrices.

Hadamard multiplication involves multiplying each element of one matrix by the corresponding element in the other matrix. You have already learned about Hadamard multiplication in Chapter 3;

the concept and notation is the same for matrices as for vectors. Thus:

$$\mathbf{C} = \mathbf{A} \odot \mathbf{B} \tag{6.19}$$

The formal definition of Hadamard multiplication is

Hadamard (element-wise) multiplication

$$c_{i,j} = a_{i,j} \times b_{i,j} \tag{6.20}$$

You might have already guessed that Hadamard multiplication is valid only for two matrices that are both $M \times N$, and the product matrix is also size $M \times N$.

One example will suffice for understanding.

$$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 6 & 3 \end{bmatrix} \odot \begin{bmatrix} 3 & 8 & 5 \\ 4 & 1 & -5 \end{bmatrix} = \begin{bmatrix} 0 & 8 & 10 \\ -4 & 6 & -15 \end{bmatrix} \tag{6.21}$$

Because Hadamard multiplication is implemented element-wise, it obeys the commutative law just like individual numbers (scalars). That is,

$$\mathbf{A} \odot \mathbf{B} = \mathbf{B} \odot \mathbf{A}$$

There is also element-wise division, which is the same principle but for division. This operation is valid only when the divisor matrix contains all non-zero elements.

$$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 6 & 3 \end{bmatrix} \varnothing \begin{bmatrix} 3 & 8 & 5 \\ 4 & 1 & -5 \end{bmatrix} = \begin{bmatrix} 0/3 & 1/8 & 2/5 \\ -1/4 & 6/1 & -3/5 \end{bmatrix}$$

$\varnothing$ indicates element-wise division.

One can debate whether Hadamard multiplication and division are really *matrix* operations; arguably, they are simply scalar multiplication or division, implemented *en masse* using compact notation. Indeed, element-wise matrix multiplication in computer applications facilitates convenient and efficient coding (e.g., to avoid using for-loops), as opposed to utilizing some special mathematical properties of Hadamard multiplication. That said, Hadamard multiplication does have applications in linear algebra. For example, it is key to one of the algorithms for computing the matrix inverse.

Code Here is another case where matrix multiplication is confusingly different between MATLAB and Python. In MATLAB, `A*B` indicates standard matrix multiplication and `A.*B` indicates Hadamard multiplication (note the *dot-star*). In Python, `A@B` gives standard matrix multiplication and `A*B` gives Hadamard multiplication.

Code block 6.5: Python

```
M1 = np.random.randn(4,3)
M2 = np.random.randn(4,3)
C = M1 * M2
```

Code block 6.6: MATLAB

```
M1 = randn(4,3);
M2 = randn(4,3);
C = M1 .* M2
```

Practice problems Hadamard-multiply the following pairs of matrices.

a) $\begin{bmatrix} -4 & -3 & -9 \\ -5 & -2 & 3 \\ 0 & -7 & 7 \end{bmatrix}, \begin{bmatrix} -1 & 4 & -3 \\ -3 & -4 & 2 \\ 3 & 2 & 5 \end{bmatrix}$

b) $\begin{bmatrix} -4 & -5 & -16 \\ -4 & -1 & 2 \\ 7 & 1 & 5 \end{bmatrix}, \begin{bmatrix} -3 & 6 & -10 & 0 \\ -4 & 9 & -6 & 4 \\ 1 & 2 & -6 & 0 \end{bmatrix}$

Answers

a) $\begin{bmatrix} 4 & -12 & 27 \\ 15 & 8 & 6 \\ 0 & -14 & 35 \end{bmatrix}$

b) Undefined!

6.9 Frobenius dot product

The Frobenius dot product, also called the Frobenius inner product, is an operation that produces a scalar (a single number) given two matrices of the same size ($M \times N$).

To compute the Frobenius dot product, you first *vectorize* the two matrices and then compute their dot product as you would for regular vectors.

Vectorizing a matrix means concatenating all of the columns in a matrix to produce a single column vector. It is a function that maps a matrix in $\mathbb{R}^{\mathrm{M \times N}}$ to a vector in $\mathbb{R}^{\mathrm{MN}}$.

Vectorizing a matrix

$$\mathbf{v}_n = a_{i,j}\,,\text{such that}$$

$$\mathbf{v} = [a_{1,1}, \ldots, a_{m,1}, a_{1,2}, \ldots, a_{m,2}, \ldots, a_{m,n}] \tag{6.22}$$

Here is an example of a matrix and the result of vectorizing it.

$$vec\left(\begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}\right) = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}$$

As with many other operations you've learned about so far, there is some arbitrariness in vectorization: Why not concatenate across the rows instead of down the columns? It could go either way, but following a common convention facilitates comprehension.

Code Note that Python defaults to row-based vectorization, which can be changed by specifying to use Fortran convention.

Code block 6.7: Python

```
A = np.array([ [1,2,3],[4,5,6] ])
A.flatten(order='F')
```

Code block 6.8: MATLAB

```
A = [ 1,2,3; 4,5,6 ];
A(:)
```

Anyway, with that tangent out of the way, we can now compute the Frobenius dot product. Here is an example:

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = \left\langle \begin{bmatrix} 1 & 5 & 0 \\ -4 & 0 & 2 \end{bmatrix}, \begin{bmatrix} 4 & -1 & 3 \\ 2 & 6 & 7 \end{bmatrix} \right\rangle_F = 5$$

Note the notation for the Frobenius dot product: $\langle \mathbf{A}, \mathbf{B} \rangle_F$

A curious yet useful way to compute the Frobenius dot product between matrices $\mathbf{A}$ and $\mathbf{B}$ is by taking the trace of $\mathbf{A}^{\mathrm{T}}\mathbf{B}$. Therefore, the Frobenius dot product can also be written as follows.

Frobenius dot product as the trace of $\mathbf{A}^{\mathbf{T}}\mathbf{B}$

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = tr(\mathbf{A}^{\mathrm{T}}\mathbf{B}) \tag{6.23}$$

I've omitted the matrix sizes in this equation, but you can tell from inspection that the operation is valid if both matrices are size $M \times N$, because the trace is defined only on square matrices.

The reason why Equation 6.23 is valid can be seen by working through a few examples, which you will have the opportunity to do in the exercises.

The Frobenius dot product has several uses in signal processing and machine learning, for example as a measure of "distance," or similarity, between two matrices.

The Frobenius inner product of a matrix with itself is the sum of all squared elements, and is called the *squared Frobenius norm* or *squared Euclidean norm* of the matrix. More on this in the next section.

Code The code below shows the trace-transpose trick for computing the Frobenius dot product.

Code block 6.9: Python

```
A = np.random.randn(4,3)
B = np.random.randn(4,3)
f = np.trace(A.T@B)
```

Code block 6.10: MATLAB

```
A = randn(4,3);
B = randn(4,3);
f = trace(A'*B);
```

Practice problems Compute the Frobenius dot product between the following pairs of matrices.

a) $\begin{bmatrix} 4 & 2 \\ 3 & 2 \\ -5 & -1 \end{bmatrix}, \begin{bmatrix} 7 & -2 \\ -7 & -8 \\ -1 & 8 \end{bmatrix}$

b) $\begin{bmatrix} 6 & 1 & 2 \\ 3 & 3 & -2 \end{bmatrix}, \begin{bmatrix} 4 & -11 & 1 \\ 6 & 1 & -2 \end{bmatrix}$

Answers

a) -16

b) 40

6.10 Matrix norms

In section 3.9 you learned that the square root of the dot product of a vector with itself is the magnitude or length of the vector, which is also called the norm of the vector.

Annoyingly, the norm of a matrix is more complicated, just like everything gets more complicated when you move from vectors to matrices. Part of the complication with matrix norms is that there are many of them! They all have some things in common, for example, all matrix norms are a single number that somehow corresponds to the "magnitude" of the matrix, but different norms correspond to different interpretations of "magnitude." In this section, you will learn a few of the common matrix norms, and we'll continue the discussion of matrix norms in Chapter 16 in the context of the singular value decomposition.

Let's start with the Frobenius norm, because it's fresh in your mind from the previous section. The equation below is an alternative way to express the Frobenius norm.

Frobenius matrix norm

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{i,j})^2} \tag{6.24}$$

Now you know three ways to compute the Frobenius norm: (1) directly implementing Equation 6.24, (2) vectorizing the matrix and computing the dot product with itself, (3) computing $tr(\mathbf{A}^{\mathrm{T}}\mathbf{A})$.

If we think of a matrix space as Euclidean, then the Frobenius norm of the subtraction of two matrices provides a measure of Euclidean distance between those matrices. The right-hand side of the formula below should look familiar from computing Euclidean distance between two points.

Euclidean distance between two matrices

$$\|\mathbf{A} - \mathbf{B}\|_F = \sqrt{\sum_{i,j}(a_{i,j} - b_{i,j})^2} \tag{6.25}$$

Of course, this formula is valid only for two matrices that have the same size.

The Frobenius norm is also called the $\boldsymbol{\ell}2$ norm (the $\boldsymbol{\ell}$ character is just a fancy-looking l, or a lower-case cursive L). There is also an $\boldsymbol{\ell}1$ matrix norm. To compute the $\boldsymbol{\ell}1$ norm, you sum the absolute values of all individual elements in each column, then take the largest maximum column sum.

There are many other matrix norms with varied formulas. Different applications use different norms to satisfy different criteria or to minimize different features of the matrix. Rather than overwhelm you with an exhaustive list, I will provide one general formula for the *matrix p-norm*; you can see that for $p = 2$, the following formula is equal to the Frobenius norm.

$$\|\mathbf{A}\|_p = \left(\sum_{i=1}^{M}\sum_{j=1}^{N}|a_{ij}|^p\right)^{1/p} \tag{6.26}$$

Cauchy-Schwarz inequality In Chapter 3 you learned about the Cauchy-Schwarz inequality (the magnitude of the dot product between two vectors is no larger than the product of the norms of the two vectors). There is a comparable inequality for the Frobenius norm of a matrix-vector multiplication:

$$\|\mathbf{Av}\| \leq \|\mathbf{A}\|_F \|\mathbf{v}\| \tag{6.27}$$

The proof for this inequality comes from integrating (1) the row perspective of multiplication, (2) the Cauchy-Schwarz inequality for the vector dot product introduced in Chapter 3, and (3) linearity of the squared Frobenius norm across the rows (that is, the squared Frobenius norm of a matrix equals the sum of the squared Frobenius norms of the rows; this comes from the summation in Equation 6.24).

Let's start by re-writing the squared norm of the matrix-vector multiplication as the sum of squared vector norms coming from the dot products of each row of $\mathbf{A}$ with $\mathbf{v}$ (m is the number of rows and $\mathbf{a}_i$ is the i^{th} row of $\mathbf{A}$).

$$\|\mathbf{Av}\|^2 = \|\mathbf{a}_1\mathbf{v}\|^2 + ... + \|\mathbf{a}_m\mathbf{v}\|^2 = \sum_{i=1}^{m} \|\mathbf{a}_i\mathbf{v}\|^2 \tag{6.28}$$

The dot-product Cauchy-Schwarz inequality allows us to write the following.

$$\sum_{i=1}^{m} \|\mathbf{a}_i\mathbf{v}\|^2 \leq \sum_{i=1}^{m} \|\mathbf{a}_i\|^2 \|\mathbf{v}\|^2 \tag{6.29}$$

Finally, we re-sum the norms of the rows back to the squared norm of the matrix. And that brings us back to our original conclusion in Equation 6.27: The norm of a matrix-vector product is less than or equal to the product of the Frobenius norms of the matrix and the vector. (You could also square the terms in 6.27 – or take the square root of both sides of Equation 6.29 – although that doesn't change the inequality.)

Code Different matrix norms can be obtained by specifiying different inputs into the `norm` functions. The code below shows Frobenius norm.

Code block 6.11: Python

```python
A = np.random.randn(4,3)
np.linalg.norm(A, 'fro')
```

Code block 6.12: MATLAB

```matlab
A = randn(4,3);
norm(A, 'fro')
```

Practice problems Compute Euclidean distance between the following pairs of matrices (note: compare with the exercises on page 166).

a) $\begin{bmatrix} 4 & 2 \\ 3 & 2 \\ -5 & -1 \end{bmatrix}, \begin{bmatrix} 7 & -2 \\ -7 & -8 \\ -1 & 8 \end{bmatrix}$

b) $\begin{bmatrix} 6 & 1 & 2 \\ 3 & 3 & -2 \end{bmatrix}, \begin{bmatrix} 4 & -11 & 1 \\ 6 & 1 & -2 \end{bmatrix}$

Answers

a) $\sqrt{322} \approx 17.94$

b) $\sqrt{162} \approx 12.73$

6.11 Matrix asymmetry index

Hiding inside every square non-symmetric matrix is a perfectly symmetric matrix, just waiting to be found. In fact, **every square matrix can be expressed as the sum of a symmetric matrix and a skew-symmetric matrix** (a skew-symmetric matrix can also be called an asymmetric matrix).

That's quite a claim, and it's not really obvious at first. Let me start with a simple example to illustrate the concept:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 4 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 7 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \tag{6.30}$$

A moment's inspection will reveal that the matrix on the left-hand-side is not symmetric, but can be expressed as the sum of a symmetric and a skew-symmetric matrix. You can think of the skew-symmetric matrix as a "residual" that we added to the symmetric matrix to produce the original matrix. In this example, the skew-symmetric part was "small" in the sense that it was sparse and the non-zero elements were numerically small relative to the numbers in the symmetric part. Later in this section we're going to quantify that "smallness" to compute the *matrix asymmetry index*, but first I want to show you how to decompose a non-symmetric matrix into the sum of symmetric and skew-symmetric matrices.

You will remember that the additive method for creating a symmetric matrix involved adding a square matrix to its transpose and dividing by 2. The complement of this is creating a skew-symmetric matrix, which works by *subtracting* instead of adding. In the equation below, $\mathbf{K}$ is a skew-symmetric matrix. We assume that $\mathbf{A}$ is square, but it doesn't need to be symmetric or skew-symmetric.

$$\mathbf{K} = (\mathbf{A} - \mathbf{A}^{\mathrm{T}})/2 \tag{6.31}$$

Here's an example:

$$\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} - \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \right) /2 = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

A few remarks: (1) Of course the diagonal must be zero; (2) The division by 2 is not necessary for the anti-symmetry, but keeps the numerical values in the same range as the original matrix; (3) If matrix $\mathbf{A}$ is already symmetric, then matrix $\mathbf{K}$ is the zeros matrix; (4) If matrix $\mathbf{A}$ is already skew-symmetric, then $\mathbf{K} = \mathbf{A}$.

With that in mind, we can now decompose any square matrix into the sum of a symmetric and a skew-symmetric matrix:

$$\mathbf{A}_k = (\mathbf{A} - \mathbf{A}^{\mathrm{T}})/2 \tag{6.32}$$

$$\mathbf{A}_s = \mathbf{A} - \mathbf{A}_k \tag{6.33}$$

$\mathbf{A}_k$ is the skew-symmetric "layer" of $\mathbf{A}$, and $\mathbf{A}_s$ is the symmetric "layer."

It is self-evident that $\mathbf{A}_k$ is skew-symmetric, and it is also self-evident that $\mathbf{A}_k + \mathbf{A}_s = \mathbf{A}$. But it is not obvious that $\mathbf{A}_s$ is symmetric, so let's prove that. In words: that quantity minus its transpose equals the zeros matrix. In math:

$$(\mathbf{A} - \mathbf{A}_k) - (\mathbf{A} - \mathbf{A}_k)^{\mathrm{T}} \tag{6.34}$$

$$= \mathbf{A} - \mathbf{A}_k - \mathbf{A}^{\mathrm{T}} + \mathbf{A}_k^{\mathrm{T}} \tag{6.35}$$

$$= (\mathbf{A} - \mathbf{A}^{\mathrm{T}}) - (\mathbf{A}_k - \mathbf{A}_k^{\mathrm{T}}) \tag{6.36}$$

$$= \mathbf{0} \tag{6.37}$$

Equation 6.36 contains a difference between two terms: The first term $(\mathbf{A} - \mathbf{A}^{\mathrm{T}})$ is simply $2\mathbf{A}_k$ by definition; the second term is actually also $2\mathbf{A}_k$, which cancels the first term. Thus, we subtracted $\mathbf{A}_s$ from its transpose and got the zeros matrix, which proves that $\mathbf{A}_s$ is symmetric.

Let's return to the main goal of this section: to derive a scalar index that will quantify how asymmetric a matrix is. Conceptually, this involves quantifying how "small" the residual skew-symmetric matrix is, relative to the original matrix. A purely symmetric matrix should have an index of 0, whereas a purely skew-symmetric matrix (i.e., the exact opposite of symmetry) should have an index of 1.

This index is called the *matrix asymmetric index*, and I'll use $\mathbf{A}_\sigma$ to indicate the asymmetry index of matrix $\mathbf{A}$.

The index is computed as the ratio of the norms of asymmetric "layer" to the original matrix:

$$\mathbf{A}_\sigma = \|\mathbf{A}_k\|_F^2 \;/\; \|\mathbf{A}\|_F^2 \tag{6.38}$$

How do we interpret this index? Let's start by thinking about what happens with a perfectly symmetric $\mathbf{A}$. Then $\mathbf{A}_k$ is the zeros matrix, the norm of which is zero, which means that $\mathbf{A}_\sigma = 0$. Now let's think about a perfectly skew-symmetric matrix: $\mathbf{A}_k = \mathbf{A}$, and if the matrices are the same, then their norms are the same, which means $\mathbf{A}_\sigma = 1$.

Those are the extremes; a matrix asymmetry index between 0 and 1 means that the matrix is neither purely symmetric nor purely skew-symmetric, but something in between. (By the way, if you want a measure of *symmetry* instead of a *asymmetry*, you could simply take $1 - \mathbf{A}_\sigma$).

Why use the squared Frobenius norm and not any other norm? All matrix norms have the property that they are 0 for the zeros matrix and are unique (that is, two equal matrices have equal norms). But the Frobenius norm has the following neat property that relates to the sum of matrices:

$$\|\mathbf{A}_s + \mathbf{A}_k\|_F^2 = \|\mathbf{A}_s\|_F^2 + \|\mathbf{A}_k\|_F^2 + 2\left\langle \mathbf{A}_s, \mathbf{A}_k \right\rangle_F \tag{6.39}$$

Reflection

Symmetric and skew-symmetric matrices have interesting properties on their own — and when combined. For example, symmetric and skew-symmetric matrices are orthogonal, meaning that the Frobenius dot product between them is zero ($trace(\mathbf{S}^\mathrm{T}\mathbf{K}) = 0$; try it yourself in code! And then think about why this is the case.). Also, the eigenvalues of the product $\mathbf{SK}$ come in signed pairs ($\lambda = \pm a$) (try this one too!). These complements remind me of those cartoons where the super-villain is the super-hero's twin brother. I'll let you guess which matrix is the hero and which is the villain.

6.12 What about matrix division?

All this fuss about matrix multiplications... what about division? You did learn about element-wise matrix division in section 6.8, but that's not really *matrix* division; that's a compact notation for describing lots of scalar divisions. When you hear "matrix division" you're probably thinking about something like this:

$$\frac{\mathbf{A}}{\mathbf{B}}$$

This would be the equivalent of a scalar division like $\frac{2}{3}$. Well, that doesn't exist for matrices; it is not possible to *divide* one matrix by another. However, there is a conceptually comparable operation, and it is based on the idea of re-writing scalar division like this:

$$\frac{2}{3} = 2 \times 3^{-1}$$

The matrix version of this is $\mathbf{AB}^{-1}$. The matrix $\mathbf{B}^{-1}$ is called the *matrix inverse*. It is such an important topic that it merits its own chapter (Chapter 12). For now, I'll leave you with three important facts about the matrix inverse.

1. The matrix inverse is the matrix such that $\mathbf{AA}^{-1} = \mathbf{I}$
2. Not all matrices have an inverse. A matrix has a full inverse only if it is square and full-rank.
3. Matrices that lack a "true inverse" may still have a "one-sided inverse" or a "pseudoinverse." One-sided and pseudoinverses are sorta-kinda but not exactly like an inverse.

6.13 Exercises

1. Determine whether each of the following operations is valid, and, if so, the size of the resulting matrix.

$$\mathbf{A} \in \mathbb{R}^{2\times3}, \quad \mathbf{B} \in \mathbb{R}^{3\times3}, \quad \mathbf{C} \in \mathbb{R}^{3\times4}$$

a) $\mathbf{CB}$
b) $\mathbf{C}^\mathrm{T}\mathbf{B}$
c) $(\mathbf{CB})^\mathrm{T}$
d) $\mathbf{C}^\mathrm{T}\mathbf{BC}$
e) $\mathbf{ABCB}$
f) $\mathbf{ABC}$
g) $\mathbf{C}^\mathrm{T}\mathbf{BA}^\mathrm{T}\mathbf{AC}$
h) $\mathbf{B}^\mathrm{T}\mathbf{BCC}^\mathrm{T}\mathbf{A}$
i) $\mathbf{AA}^\mathrm{T}$
j) $\mathbf{A}^\mathrm{T}\mathbf{A}$
k) $\mathbf{BBA}^\mathrm{T}\mathbf{ABBCC}$
l) $(\mathbf{CBB}^\mathrm{T}\mathbf{CC}^\mathrm{T})^\mathrm{T}$
m) $(\mathbf{A} + \mathbf{ACC}^\mathrm{T}\mathbf{B})^\mathrm{T}\mathbf{A}$
n) $\mathbf{C} + \mathbf{CA}^\mathrm{T}\mathbf{ABC}$
o) $\mathbf{C} + \mathbf{BA}^\mathrm{T}\mathbf{ABC}$
p) $\mathbf{B} + 3\mathbf{B} + \mathbf{A}^\mathrm{T}\mathbf{A} - \mathbf{CC}^\mathrm{T}$
q) $\mathbf{A} \odot (\mathbf{ABC})$
r) $\mathbf{A} \odot \mathbf{ABC}(\mathbf{BC})^\mathbf{T}$

2. Compute the following matrix multiplications. Each problem should be completed twice using the two indicated perspectives of matrix multiplication (#1: element, #2: layer, #3: column, #4: row).

a) #1,2: $\begin{bmatrix}1 & 0\\3 & 1\end{bmatrix}\begin{bmatrix}0 & 5\\2 & 2\end{bmatrix}$

b) #2,4: $\begin{bmatrix}-3 & 2\\-2 & 3\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 2\end{bmatrix}$

c) #3,4: $\begin{bmatrix}11 & -5\\9 & -13\end{bmatrix}\begin{bmatrix}3 & 1\\-8 & .5\end{bmatrix}$

d) #1,4: $\begin{bmatrix}1 & 0\\0 & 2\end{bmatrix}\begin{bmatrix}a & b\\c & d\end{bmatrix}$

e) #2,3: $\begin{bmatrix}2 & 2\\1 & 3\end{bmatrix}\begin{bmatrix}10 & 1\\-5 & 4\end{bmatrix}$

f) #1,3: $\begin{bmatrix}a & 0\\b & 0\end{bmatrix}\begin{bmatrix}2 & 3\\4 & 1\end{bmatrix}$

g) #2,3: $\begin{bmatrix}a & b\\0 & 0\end{bmatrix}\begin{bmatrix}2 & 3\\4 & 1\end{bmatrix}$

h) #1,2: $\begin{bmatrix}1 & 0 & 4\\0 & 1 & 1\\3 & 3 & 0\end{bmatrix}\begin{bmatrix}-2 & -3 & -1\\-1 & -9 & 3\\0 & 1 & 5\end{bmatrix}$

i) #2,3: $\begin{bmatrix}a & 0 & 1\\0 & b & 0\\1 & 0 & c\end{bmatrix}\begin{bmatrix}a & b & c\\1 & 2 & 3\\0 & 0 & 1\end{bmatrix}$

3. Compute the following matrix-vector products, if the operation is valid.

a) $\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ **b)** $\begin{bmatrix} 2 \\ 3 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$ **c)** $\begin{bmatrix} 2 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

d) $\begin{bmatrix} 2 \\ 3 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 2 & 1 \\ 2 & 3 \end{bmatrix}$ **e)** $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ **f)** $\begin{bmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \\ 1 \end{bmatrix}$

g) $\begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ **h)** $\begin{bmatrix} 3 \\ 3 \\ 2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 1 & 3 & 2 \\ 6 & 1 & 5 \\ 3 & 5 & 0 \end{bmatrix}$

4. Consider square matrices $\mathbf{A}$ and $\mathbf{B}$, with nonzero values at all elements. What assumptions on symmetry would make the following equalities hold (note: the operations might also be impossible under any assumptions)? Provide a proof or example for each.

a) $\mathbf{AB} = \mathbf{A}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}$ **b)** $\mathbf{AB} = (\mathbf{AB})^{\mathrm{T}}$
c) $\mathbf{AB} = \mathbf{AB}^{\mathrm{T}}$ **d)** $\mathbf{AB} = \mathbf{A}^{\mathrm{T}}\mathbf{B}$
e) $\mathbf{AB} = \mathbf{B}^{\mathrm{T}}\mathbf{A}$ **f)** $\mathbf{AB} = (\mathbf{BA})^{\mathrm{T}}$

5. In section 6.7 you learned that the product of two symmetric matrices is generally not symmetric. That was for standard multiplication; is the *Hadamard product* of two symmetric matrices symmetric? Work through your reasoning first, then test your hypothesis on the following matrix pair.

$$\begin{bmatrix} 2 & 5 & 7 \\ 5 & 3 & 6 \\ 7 & 6 & 4 \end{bmatrix}, \begin{bmatrix} a & d & f \\ d & b & e \\ f & e & c \end{bmatrix}$$

6. For the following pairs of matrices, vectorize and compute the vector dot product, then compute the Frobenius inner product as $tr(\mathbf{A}^{\mathrm{T}}\mathbf{B})$.

a) $\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ **b)** $\begin{bmatrix} 0 & 5 \\ 7 & -2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 13 & 14 \end{bmatrix}$

c) $\begin{bmatrix} 4 & -5 & 8 \\ 1 & -1 & 2 \\ -2 & 2 & -4 \end{bmatrix}, \begin{bmatrix} 4 & -5 & 8 \\ 1 & -1 & 2 \\ -2 & 2 & -4 \end{bmatrix}$ **d)** $\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \begin{bmatrix} a & b \\ a & b \end{bmatrix}$

e) $\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ **f)** $\begin{bmatrix} 1 & 1 & 7 \\ 2 & 2 & 6 \\ 3 & 3 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$

7. Implement the indicated multiplications for the following matrices.

$$\mathbf{A} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 2 & 1 & 3 \\ 0 & 4 & 1 \\ 2 & 2 & 3 \end{bmatrix}$$

a) AB **b) AC** **c) BC** **d) CA**
e) CB **f) BCA** **g) ACB** **h) ABC**