













In **matrix multiplication**, the shapes need to satisfy specific rules: if we have two matrices AAA and BBB, where:

* AAA has dimensions m×nm \times nm×n
* BBB has dimensions n×pn \times pn×p

then **matrix multiplication is defined** only if the number of columns in AAA (i.e., nnn) matches the number of rows in BBB (i.e., nnn). The resulting matrix CCC will have dimensions m×pm \times pm×p.

**Example of Matrix Multiplication Using np.dot or @**

Let's see an example of matrix multiplication using Numpy's np.dot() function or the @ operator, which performs matrix multiplication:

import numpy as np

# Define matrix A of shape (2, 3)

A = np.array([[1, 2, 3], [4, 5, 6]])

# Define matrix B of shape (3, 2)

B = np.array([[7, 8], [9, 10], [11, 12]])

# Matrix multiplication resulting in a shape of (2, 2)

C = np.dot(A, B)

print("Matrix multiplication result C:\n", C)

# Alternatively, using the @ operator for matrix multiplication

C\_alt = A @ B

print("Using @ operator:\n", C\_alt)

**Explanation**:

* Matrix AAA is of shape (2,3)(2, 3)(2,3), and matrix BBB is of shape (3,2)(3, 2)(3,2).
* The inner dimensions match (3), so matrix multiplication is valid.
* The result matrix CCC will have a shape of (2,2)(2, 2)(2,2).

Matrix multiplication result C:

[[ 58 64]

[139 154]]

**Calculations**:

* For C[0,0]C[0, 0]C[0,0]: 1⋅7+2⋅9+3⋅11=581 \cdot 7 + 2 \cdot 9 + 3 \cdot 11 = 581⋅7+2⋅9+3⋅11=58
* For C[0,1]C[0, 1]C[0,1]: 1⋅8+2⋅10+3⋅12=641 \cdot 8 + 2 \cdot 10 + 3 \cdot 12 = 641⋅8+2⋅10+3⋅12=64
* For C[1,0]C[1, 0]C[1,0]: 4⋅7+5⋅9+6⋅11=1394 \cdot 7 + 5 \cdot 9 + 6 \cdot 11 = 1394⋅7+5⋅9+6⋅11=139
* For C[1,1]C[1, 1]C[1,1]: 4⋅8+5⋅10+6⋅12=1544 \cdot 8 + 5 \cdot 10 + 6 \cdot 12 = 1544⋅8+5⋅10+6⋅12=154

