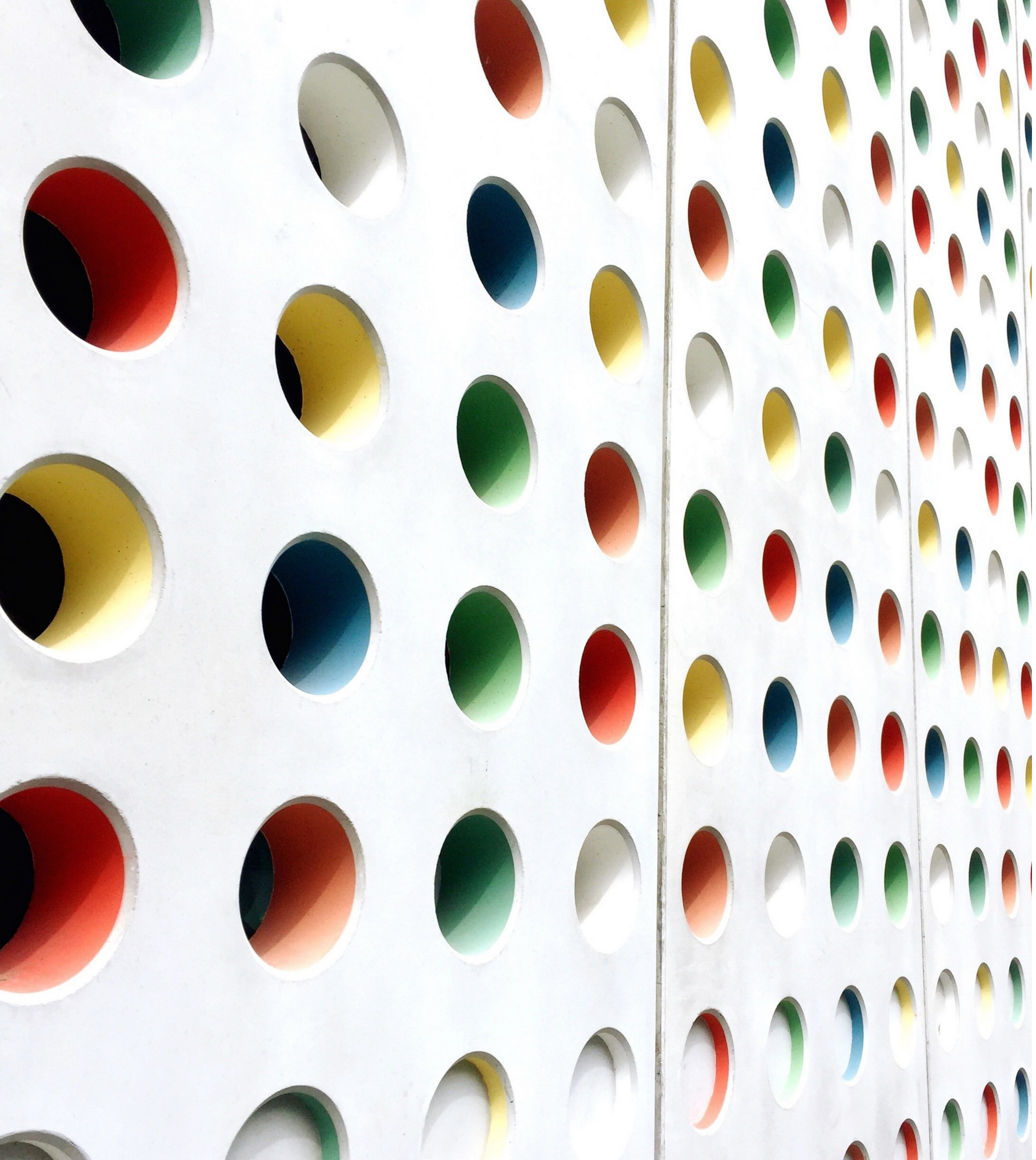

Matrix Multiplication

DEVENDER TUPPADA

FH6495

Definition

Matrix multiplication is a binary operation that produces a matrix from two matrices. For matrix multiplication, the number of columns in the first matrix must be equal to the number of rows in the second matrix. The resulting matrix, known as the matrix product, has the number of rows of the first and the number of columns of the second matrix. The product of matrices A and B is denoted as AB.

Example

If $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is an $n \times p$ matrix,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix}$$

the *matrix product* $\mathbf{C} = \mathbf{AB}$ is defined to be the $m \times p$ matrix

Example

Now the matrix C can be defined as

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{pmatrix}$$

Such that,

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj},$$

Example

for $i = 1, ..., m$ and $j = 1, ..., p$.

Therefore, $\mathbf{AB}$ can also be written as

$$\mathbf{C} = \begin{pmatrix} a_{11}b_{11} + \cdots + a_{1n}b_{n1} & a_{11}b_{12} + \cdots + a_{1n}b_{n2} & \cdots & a_{11}b_{1p} + \cdots + a_{1n}b_{np} \\ a_{21}b_{11} + \cdots + a_{2n}b_{n1} & a_{21}b_{12} + \cdots + a_{2n}b_{n2} & \cdots & a_{21}b_{1p} + \cdots + a_{2n}b_{np} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} + \cdots + a_{mn}b_{n1} & a_{m1}b_{12} + \cdots + a_{mn}b_{n2} & \cdots & a_{m1}b_{1p} + \cdots + a_{mn}b_{np} \end{pmatrix}$$

Smaller matrices

$$\begin{bmatrix} 1 & 4 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ -1 \\ 5 \end{bmatrix}$$

We have to multiply 1*3 matrix by a 3*1 matrix. The number of columns in the first is the same as the number of rows in the second, so they are compatible.

The Product is

$$\begin{aligned} &[(1)(2) + (4)(-1) + (0)(5)] \\ &= [2 + (-4) + 0] \\ &= [-2] \end{aligned}$$

Larger matrices

Multiplying a 2*3 matrix with a 3*2 matrix, to get a 2*2 matrix. The entries of the product matrix are called eij when they're in the ith row and jth column.

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 3 & 5 \\ -1 & 0 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix}$$

To get e_{11} , multiply Row 1 of the first matrix by Column 1 of the second.

$$e_{11} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} = 1\,(3) + 0\,(-1) + 1\,(2) = 5$$

To get e_{12} , multiply Row 1 of the first matrix by Column 2 of the second.

$$e_{12} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 0 \\ -1 \end{bmatrix} = 1\,(5) + 0\,(0) + 1\,(-1) = 4$$

Larger matrices

To get e_{21} , multiply Row 2 of the first matrix by Column 1 of the second.

$$e_{21} = \begin{bmatrix} 0 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} = 0\,(3) + 1\,(-1) + 2\,(2) = 3$$

To get e_{22} , multiply Row 2 of the first matrix by Column 2 of the second.

$$e_{22} = \begin{bmatrix} 0 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 0 \\ 1 \end{bmatrix} = 0\,(5) + 1\,(0) + 2\,(-1) = -2$$

Writing the product matrix, we get:

$$\begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 3 & -2 \end{bmatrix}$$

Therefore, we have shown:

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

Algorithm

```
input A and B, both n by n matrices
initialize C to be an n by n matrix of all zeros
for i from 1 to n:
    for j from 1 to n:
        for k from 1 to n:
            C[i][j] = C[i][j] + A[i][k]*B[k][j]
output C (as A*B)
```

Time Complexity

The naive matrix multiplication algorithm contains three nested loops. For each iteration of the outer loop, the total number of the runs in the inner loops would be equivalent to the length of the matrix. Here, integer operations take O(1) time. In general, if the length of the matrix is N, the total time complexity would be

$$\mathbf{O(\ N * N * N) = O(N^3)}$$

Strassen algorithm

Matrix multiplication is based on a divide and conquer-based approach. Here we divide our matrix into a smaller square matrix, solve that smaller square matrix and merge into larger results. For larger matrices this approach will continue until we recurse all the smaller sub matrices. Suppose we have two matrices, A and B, and we want to multiply them to form a new matrix, C.

C=AB, where all A,B,C are square matrices. We will divide these larger matrices into smaller sub matrices n/2; this will go on.

Strassen algorithm

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

$r=ae+bg$

$t=ce+dg$

$s=af+bh$

$u=cf+dh$

Strassen algorithm

Each of the above four equations satisfies two multiplications of n/2Xn/2 matrices and addition of their n/2xn/2 products. Using these equations to define a divide and conquer strategy we can get the relation among them as:

T(N) = 8T(N/2) + O(N2)
From the above we see that simple matrix multiplication takes eight recursion calls.
T(n)=O(n^3)
Thus, this method is faster than the ordinary one.
It takes only seven recursive calls, multiplication of n/2xn/2 matrices and O(n^2) scalar additions and subtractions, giving the below recurrence relations.

Strassen algorithm

T(N) = 7T(N/2) + O(N2)

Steps of Strassen’s matrix multiplication:

1. Divide the matrices A and B into smaller submatrices of the size n/2xn/2.
2. Using the formula of scalar additions and subtractions compute smaller matrices of size n/2.
3. Recursively compute the seven matrix products Pi=AiBi for i=1,2,…7.
4. Now compute the r,s,t,u submatrices by just adding the scalars obtained from above points.

Submatrix Products

We have read many times how two matrices are multiplied. We do not exactly know why we take the row of one matrix A and column of the other matrix and multiply each by the below formula.

Pi=AiBi
=(α1ia+α2ib+α3ic)(β1ie+β2if+β2ig)
Where a,b ,β,α are the coefficients of the matrix that we see here, the product is obtained by just adding and subtracting the scalar.

Time Complexity : **O(N^2)**

Submatrix Products

p1 = a(f - h)
p2 = (a + b)h
p3 = (c + d)e
p4 = d(g - e)
p5 = (a + d)(e + h)
p6 = (b - d)(g + h)
p7 = (a - c)(e + f)

The A x B can be calculated using above seven multiplications.
Following are values of four sub-matrices of result C

$$\underset{A}{\begin{bmatrix} a & b \\ c & d \end{bmatrix}} \times \underset{B}{\begin{bmatrix} e & f \\ g & h \end{bmatrix}} = \underset{C}{\begin{bmatrix} p5 + p4 - p2 + p6 & p1 + p2 \\ p3 + p4 & p1 + p5 - p3 - p7 \end{bmatrix}}$$

Thank You