ELEN4020 Laboratory 1: Multidimensional array analysis

Emma Clark (1088496)
Jason Smit (709363)
Isabel Tollman (728359)

School of Electrical & Information Engineering, University of the Witwatersrand
Private Bag 3, 2050, Johannesburg, South Africa

Abstract: Rank 2 and rank 3 tensor addition and multiplication procedures are discussed. By initially presenting algorithms to perform rank 2 procedures, it becomes clear that those of rank 3 are extensions of the traditional methods used to achieve 2-dimensional addition and multiplication.

Key words: Multidimesional tensor, addition, multiplication, pseudo-code, C++

1 INTRODUCTION

To implement 3-dimensional (3-D) array multiplication, an investigation of tensor operations is performed. Tensor operation discussion begins with tensors of rank 2. The pseudo-code developed to achieve rank 2 tensor procedures is presented.

2-Dimensional (2-D) tensor addition is elaborated on in Section 2, while multiplication of 2-D arrays is discussed in Section 3. Using the basis of rank 2 tensors, rank 3 tensor operations are demonstrated, and the pseudo-code developed to implement the procedures is presented. 3-D array addition and multiplication are illustrated in Sections 4 and 5, respectively. The pseudo-code of each procedure is used to produce C++ code from which to perform all multidimensional array operations.

2 2-DIMENSIONAL TENSOR ADDITION

The addition of two 2-D tensors is congruent to addition of two 2-D matrices. Therefore, element-by-element addition is required to achieve the output 2-D matrix, as shown in Equation 1 [1]. Algorithm 1 displays the pseudo-code used to calculate the addition of two N×N matrices.

$$C_{ij} = A_{ij} + B_{ij} \tag{1}$$

3 2-DIMENSIONAL TENSOR MULTIPLICATION

The multiplication of rank 2 tensors is equivalent to 2-D matrix multiplication, performed by taking the dot product of each row and column, respectively. This is displayed in Equation 2 [2]. Algorithm 2 shows the pseudo-code used to multiply two N×N matrices.

$$C_{ij} = \sum_{k} A_{ij} \times B_{ij} \tag{2}$$

4 3-DIMENSIONAL TENSOR ADDITION

Rank 2 tensor addition, discussed in Section 2, is used as a basis for 3-D array addition. Algorithm 1 is extended to produce Algorithm 3, applicable for rank 3 tensor addition. When a third rank is incorporated, array addition follows the same procedure as presented in Algorithm 1, however, the additional rank corresponds to an additional vertex of summation.

Element-by-element addition is implemented in a similar fashion to the 2-D array procedure. The difference comes from the inclusion of the additional dimension. Figure 3 graphically depicts 3-D tensor addition, while Algorithm 3 shows the pseudo-code used to sum two N×N×N matrices.

5 3-DIMENSIONAL TENSOR MULTIPLICATION

The multiplication of two 3-D arrays (rank 3 tensor contraction) uses 2-D matrix multiplication as a basis. The i^{th} row-plane of array A and the j^{th} column-plane of array B are multiplied using traditional 2-D matrix multiplication shown in Algorithm 2. The result is the k^{th} layer-plane of array C. Algorithm 4 provides the pseudo-code used to multiply two N×N×N arrays. Figure 6 illustrates the implemented algorithm.

6 IMPLEMENTATION

The algorithms presented in Sections 2, 3, 4, and 5 are implemented using C++. The generated code performs as detailed by the pseudo-code. One .cpp file exists. It is accessible via the public GitHub repository found through link `https://github.com/IsabelTollman/ELEN4020A_Group7/tree/Code` under the '*Code*' branch and within the *Code* folder.

7 CONCLUSIONS

The formation of rank 3 tensor operations is based on the widely known traditional rules of rank 2 tensors. Rank 3 tensor addition requires a further iteration than 2-D tensor addition. 3-D by 3-D array

Algorithm 1: rank2TensorAdd finds the addition of two $N \times N$ matrices

Input: Three $N \times N$ constant integer arrays: $a[N][N], b[N][N]$ and $c[N][N]$
Output: void

$N \leftarrow \text{size}(a)$
for $i \leftarrow 0$ ***to*** $N-1$ **do**
  for $j \leftarrow 0$ ***to*** $N-1$ **do**
    $c[i][j] = a[i][j] + b[i][j]$

Figure 1 : Listing showing the pseudo code for Rank 2 addition

Algorithm 2: rank2TensorMulti finds the multiplication of two $N \times N$ matrices

Input: Three $N \times N$ constant integer arrays: $a[N][N], b[N][N]$ and $c[N][N]$
Output: void

$N \leftarrow \text{size}(a)$
for $k \leftarrow 0$ ***to*** $N-1$ **do**
  for $i \leftarrow 0$ ***to*** $N-1$ **do**
    for $j \leftarrow 0$ ***to*** $N-1$ **do**
      $c[i][j] = c[i][j] + a[i][j] \times b[i][j]$

Figure 2 : Listing showing the pseudo code for Rank 2 multiplication

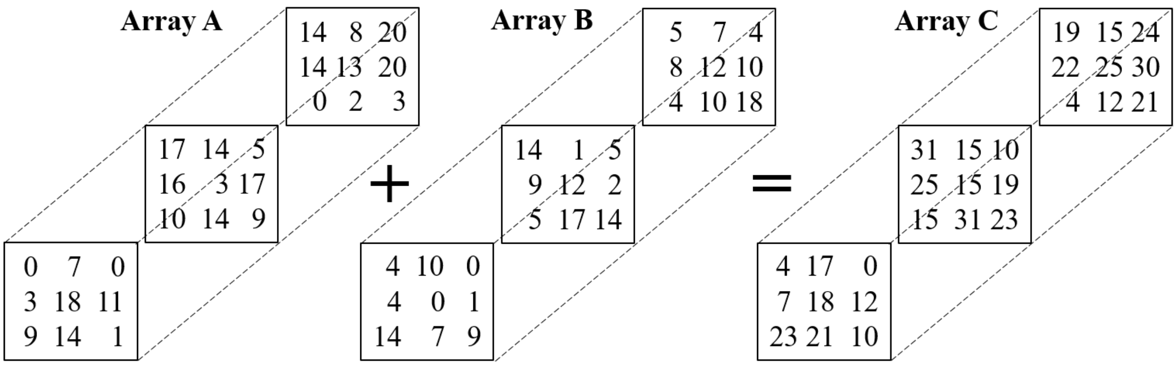

Figure 3 : Visualisation of rank 3 tensor addition

Algorithm 3: rank3TensorAdd finds the addition of two $N \times N \times N$ matrices

Input: Three $N \times N \times N$ constant integer arrays: $a[N][N][N], b[N][N][N]$ and $c[N][N][N]$
Output: void

$N \leftarrow \text{size}(a)$
for $k \leftarrow 0$ ***to*** $N-1$ **do**
  for $i \leftarrow 0$ ***to*** $N-1$ **do**
    for $j \leftarrow 0$ ***to*** $N-1$ **do**
      $c[i][j][k] = a[i][j][k] + b[i][j][k]$

Figure 4 : Listing showing the pseudo code for Rank 3 addition

Algorithm 4: rank3TensorMulti finds the multiplication of two $N \times N \times N$ matrices

Input: Three $N \times N \times N$ constant integer arrays: $a[N][N][N], b[N][N][N]$ and $c[N][N][N]$
Output: void

$N \leftarrow \text{size}(a)$
$temp\_c = zeros(N, N)$
for $k \leftarrow 0$ ***to*** $N - 1$ **do**
    $temp\_a = a[k][:][:]$
    $temp\_b = b[:][k][:]$
    $rank2TensorMulti(temp\_a[N][N], temp\_b[N][N], temp\_c[N][N])$
    $c[:][:][k] = temp\_c$

Figure 5 : Listing showing the pseudo code for Rank 3 multiplication

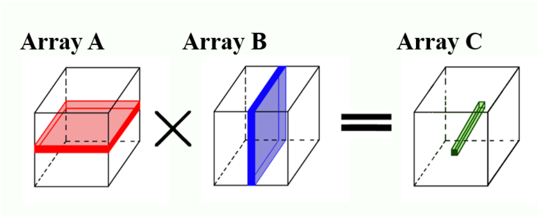

Figure 6 : Visualisation of rank 3 tensor multiplication

multiplication uses the procedure of rank 2 tensor multiplication while incorporating the additional dimension through an extra iteration.

REFERENCES

[1] E. Stapel. "Adding and Subtracting Matrices." Online, 2012. URL `https://www.purplemath.com/modules/mtrxadd.htm`.

[2] MathsIsFun.com. "How to Multiply Matrices." Online, 2017. URL `https://www.mathsisfun.com/algebra/matrix-multiplying.html`.