CS 594: Representations in Algorithm Design **Spring 2022**

Lecture on 03/29/2022

Lecturer: Xiaorui Sun *Scribe: Sai Anish Garapati*

1 Last Lecture's Review

In the last lecture Matrix multiplication was discussed and following algorithms were introduced to solve Matrix multiplication:

- Naive Algorithm
- Strassen Algorithm
- Tensor

2 Today's Lecture

- Matrix multiplication algorithm represented as Bilinear and Trilinear problem.
- Rank of a Tensor and application of Tensor Rank to solve Matrix multiplication algorithm.

3 Matrix multiplication as a linear problem

3.1 Bilinear Problem

There are two sets of variables $X_1, X_2, ..., X_N$ and $Y_1, Y_2, ..., Y_N$
The set of K Bilinear functions $F = \{f_1, f_2, ..., f_K\}$ can be defined as:

$$f_k = \sum_{i=1}^{N} \sum_{j=1}^{M} t_{ijk} \cdot X_i \cdot Y_j$$

where f_k is a bilinear function.
The goal is to compute Tensors $f_1, f_2, ..., f_k$ for given $X_1, X_2, ..., X_N$ and $Y_1, Y_2, ..., Y_N$

Then the n x n matrix multiplication of Matrices $\{x_{i,j}\}$ and $\{y_{j,k}\}$ for i, j, k = 1, 2, ..., n can be written as:

$$z_{ik} = \sum_{j=1}^{n} x_{ij} \cdot y_{jk}$$

where z_{ik} is element at i^{th} row and j^{th} column in matrix $X \cdot Y$.

In Bilinear problem:

$$z_{ik} = \sum_{i,j',j,k',k,i'} t_{(i,j')(j,k')(i',k)} \cdot x_{ij'} \cdot y_{jk'}$$

where

$$t_{(i,j')(j,k')(i',k)} = \begin{cases} 1 \text{ if } i = i', j = j', k = k' \\ 0 \text{ otherwise} \end{cases}$$

• Strassen's recursive approach can also be generelized to any Bilinear algorithm for Matrix multiplication.

3.2 Trilinear Problem

• The same set of coeffiecients used above can be used to represent the Matrix multiplication in Trilinear form:

$$\sum_{i,j,k} t_{ijk} \cdot X_i \cdot Y_j \cdot Z_k$$

where Z_k represents function f_k and t_{ijk} is called an Order 3 Tensor.

4 Rank of Tensor

• The Rank of a Tensor $\sum_{i,j,k=1}^{N,M,K} t_{ijk} \cdot X_i \cdot Y_j \cdot Z_k$ is defined as the minimum of integer l such that $\exists$

$$\vec{u}_\lambda = (u_1, u_{\lambda_2}, ..., u_{\lambda_N})$$

$$\vec{v}_\lambda = (v_{\lambda_1}, v_{\lambda_2}, ..., v_{\lambda_M})$$

$$\vec{w}_\lambda = (w_{\lambda_1}, w_{\lambda_2}, ..., w_{\lambda_K})$$

such that

$$\sum_{i,j,k=1}^{N,M,K} t_{i,j,k} \cdot X_i \cdot Y_j \cdot Z_k = \sum_{\lambda=1}^{l} (\sum_{i=1}^{N} u_{\lambda_i} \cdot X_i)(\sum_{j=1}^{M} v_{\lambda_j} \cdot Y_j)(\sum_{k=1}^{k} w_{\lambda_k} \cdot Z_k)$$

• This can be seen as representing the function as a linear combination of products of atomic tensors.

• Another way to look at the above definition is:
For given vectors $\vec{u}_\lambda, \vec{v}_\lambda, \vec{w}_\lambda$,

$$u_\lambda \otimes v_\lambda \otimes w_\lambda = (\sum_i u_{\lambda_i} X_i)(\sum_j v_{\lambda_j} Y_j)(\sum_k w_{\lambda_k} Z_k)$$

• Rank: The Rank of Tensor is defined as the number of atomic tensors required in combination to produce the original tensor.

• The Rank of a Tensor is analogous to the Time complexity of the Tensor.

• The Rank of a Matrix mirrors the number of rows required to produce all the rows inside the Matrix by some linear combination.

• For atomic tensors (rank 1 tensor or triad) the time complexity is given as $O(N + M + K)$

• For l ranked tensor the time complexity is given as $O((N + M + K) \cdot l)$

5 Rank of a Tensor and Running Time of Matrix Multiplication

5.1 Computing model: Straight Line Program (SLP) model

• For a given input $X_1, X_2, ..., X_N$, the goal is to compute $F(X_1, X_2, ..., X_N)$, a sequence of operations $g_1, g_2, ..., g_s$ are allowed which are:

1. $g_i = x_j \odot x_j$ where $\odot \in \{+, -, \times, /\}$
2. $g_i = X_j \otimes C$
3. $g_i = X_j \otimes g_k$ for $k < i$
4. $g_i = g_j \otimes C$
5. $g_i = g_j \otimes g_k$ for $j, k < i$

• $g_1, g_2, ..., g_s = F(X_1, X_2..., X_3)$ and the last operation g_s gives the function value.

• The Complexity $C(F)$ is calculated as the Minimum number of operations used in a SLP model to compute the function value F.

• $C^{\times/}(F)$ is the minimum number of $\times$ or / operations used in a SLP model to compute the function value F.

Theorem (Strassen in 1973):
For a bilinear form $F(X_1, X_2, ..., X_N)$, if complexity $C^{\times/}(F) = l$, then F is a linear combination of

$$P_\lambda = (\sum u_{\lambda_i} X_i)(\sum v_{\lambda_j} X_i)$$
$$for\ 1 <= \lambda <= l$$

- For a Tensor t:

$$if\ complexity\ C^{\times/}(t) = l \implies Rank\ of\ Tensor\ t <= l$$

$$\implies C^{\times/}(t) <= R(t) <= 2 \cdot C^{\times/}(t)$$

• Omega w is defined as the smallest P such that $C(< n, n, n >) <= O(n^p)$.

• $< n, m, k >$ is used to denote the Matrix multiplication tensor for N $\times$ M and M $\times$ K matrices.

Theorem:

$$R(< n, n, n >) = O(C(< n, n, n >))$$

• The Rank $R(< n, n, n >)$ can be related with the Rank of a smaller tensor say $R(< 2, 2, 2 >)$.

• In order to relate larger tensor with smaller tensor we use operations such as Kronecker product.

Let t be tensor N $\times$ M $\times$ K and $t^{'}$ be tensor $N^{'} \times M^{'} \times K^{'}$, then $(t \otimes t^{'})_{ii^{'},jj^{'},kk^{'}}$ is a tensor $= t_{ijk} \cdot t_{i^{'}j^{'}k^{'}}$

Observation:
Let Tensor $T = N \cdot M \cdot K$, then

$$< T, T, T > = < N, M, K > \otimes < M, K, N > \otimes < K, N, M >$$

Claim:

$$R(t \otimes t^{'}) <= R(t) \cdot R(t^{'})$$

Claim:

$$R(< N, M, K >) = R(< M, K, N >)$$

$\implies R(< T, T, T >) <= R(< N, M, K >)^3$

- If $R(< N, M, K >) <= r$ where r is an upper limit:

then $w <= \frac{3 \cdot logr}{log(KMN)}$

- For N = 2, M = 2 and K = 2, we get

$$R(< 2, 2, 2 >) \implies w <= \frac{3 \cdot log_2 7}{log_2 8} = log_2 7$$