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

Lecture 21: 03/29/2022

Lecturer: Xiaorui Sun *Scribe: Arvind Gupta*

1 Last time and today

- Tensor Rank and Matrix multiplication
- Quadratic problem
- Bilinear problem
- Matrix multiplication as a Bilinear problem

2 Tensor rank - bilinear problem

Proof: Consider a Bilinear problem:

$$X = (x_1, x_2, ...x_{n-1}, x_n) \; and \; Y = (y_1, y_2, ...y_{m-1}, y_m)$$

$$F = (f_1, f_2, ..., f_k),$$

where f_k is a bilinear function.

$$f_k = \sum_{i=1}^{N} \sum_{j=1}^{M} t_{ijk} * x_i * y_j \tag{1}$$

Goal: Compute $f_1, f_2, ..., f_k$ **for given:**

$$X = (x_1, x_2, ...x_{n-1}, x_n) \; and \; Y = (y_1, y_2, ...y_{m-1}, y_m)$$

$x_{i,j} \ * \ y_{j,k}$ where i, j, k = (1 to n) and,

$$z_{i,k} = \sum_{j=1}^{n} x_{ij} * y_{jk}, \tag{2}$$

where i = row, y = column in X * Y

$$z_{i,k} = \sum_{i,j',j,k',k,i'} t_{(i,j')(j,k')(i',k)} * x_{ij'} * y_{jk'} = \begin{cases} 1, & \text{if } i = i', j = j', k = k' \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

2.1 Trilinear Problem

$$\sum_{i,j,k} t_{ijk} * x_i * y_j * z_k, \tag{4}$$

where z_k represents function f_k, and t_{ijk} represents order 3 tensor,

$$\begin{aligned} \vec{z} &= (1, 0, 0, 0, 0........, 0) \\ &= (0, 1, 0, 0, 0........, 0) \\ &= (0, 0, 1, 0, 0........, 0) \end{aligned}$$

2.2 Rank of Tensor

Rank of :

$$\sum_{i,j,k}^{N,M,K} t_{ijk} * x_i * y_j * z_k,$$

is the minimum of l, such that :

$$\begin{aligned} \exists\ \vec{\mu} &= (\mu_{\lambda_1}, \mu_{\lambda_2}, ..., \mu_{\lambda_n}), \\ \vec{\nu} &= (\nu_{\lambda_1}, \nu_{\lambda_2}, ..., \nu_{\lambda_n}), \\ \vec{\omega} &= (\omega_{\lambda_1}, \omega_{\lambda_2}, ..., \omega_{\lambda_n}), \end{aligned}$$

for all $1 \leq \lambda \leq$ l

$$\sum_{i,j,k}^{N,M,K} t_{ijk} * x_i * y_j * z_k = \sum_{\lambda=1}^{l} ((\sum_{i=1}^{N} \mu_{\lambda_i} * x_i) * (\sum_{j=1}^{M} \nu_{\lambda_j} * y_j) * (\sum_{k=1}^{K} \omega_{\lambda_k} * z_k)) \tag{5}$$

for given $\vec{\mu_\lambda}$, $\vec{\nu_\lambda}$, $\vec{\omega_\lambda}$, where

$$\mu_\lambda \bigotimes \nu_\lambda \bigotimes \omega_\lambda = (\sum_{i=1}^{N} \mu_{\lambda_i} * x_i) * (\sum_{j=1}^{M} \nu_{\lambda_j} * y_j) * (\sum_{k=1}^{K} \omega_{\lambda_k} * z_k), \quad (6)$$

the atomic tensors are defined in the above mentioned way, and atomic tensors are basically the tensors with rank 1.

2.2.1 Complexity to evaluate tensors

In order to evaluate single atomic tensor, the running time is O(N + M + K)
If the rank of tensor is (l), then running time = O((N * M * K) * l)

3 Connecting the tensor rank with running time of matrix multiplication

3.1 Defining computing model :

Straight-line program model (SLP)

$$Input = X_1........X_n$$
$$Goal = F(X_1......X_N),$$

a sequence of operations $(g_1.....g_s)$ = F$(X_1.....X_n)$, where operations allowed are :

$$g_i = X_j \bigodot X_j \quad (7)$$

$$g_i = X_j \bigodot C,\ whereC\ is\ a\ constant \quad (8)$$

$$g_i = X_j \bigodot g_k,\ wherek < i \quad (9)$$

$$g_i = g_j \bigodot C,\ whereC\ is\ a\ constant \quad (10)$$

$$g_i = g_j \bigodot g_k,\ where(j, k < i) \quad (11)$$

where $\bigodot$ = (+, -, *, /)

Complexity C(F) = minimum number of operations used in an SLP to compute F
Complexity of using only multiply (*) and divide (/) operations :
$C^{*/}$ = minimum number of multiply (*) and divide (/) operations used in SLP for (F)

4 Strassen Algorithm (1973)

Theorem: For a bilinear function (F) : $(X_1, ..., X_n)$ **and** $(Y_1, ..., Y_m)$**, if complexity** $C^{*/}$ **= l, then (F) is a linear combination of :**

$$P_\lambda = (\sum_{i=1}^{n} \mu_{\lambda_i} * x_i) * (\sum_{j=1}^{m} \nu_{\lambda_j} * y_j), \quad (12)$$

for all $1 \leq \lambda \leq l$ and for a tensor (t), if $C^{*/}$(t) = l, then

$$C^{*/}(t) \leq R(t) \leq Z * C^{*/}(t) \quad (13)$$

where R(t) is the rank of tensor (t),

Above equation implies that :

$$\omega \leq log_n R(< n, n, n >) \quad (14)$$

where ω is used to denote the exponent of number of operations used.

4.1 Kronecker Product

Let (t) be a tensor : $N \times M \times K$
and $t^/$ be a tensor : $N^/ \times M^/ \times K^/$, then

$$(t \bigotimes t^/)_{ii^/,jj^/,kk^/} is\ a\ tensor\ = NN^/ \times MM^/ \times KK^/ \quad (15)$$

$$(t \bigotimes t^/)_{ii^/,jj^/,kk^/} = t_{ijk} t_{i^/ j^/ k^/} \quad (16)$$