13. Bilinear Complexity, Tensor Rank, Matrix Multiplication Exponent

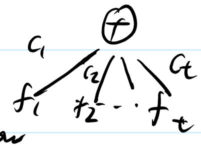
Saturday, September 30, 2023 8:50 PM

We say a polynomial is a <u>quadratic form</u> if it is homogeneous of degree 2,
and a <u>linear form</u> if it is homogeneous of degree 1.

<u>Def:</u> The multiplicative complexity of a set $\{f_1, \dots, f_m\}$ of quadratic forms in $X_1, \dots, X_n$ is

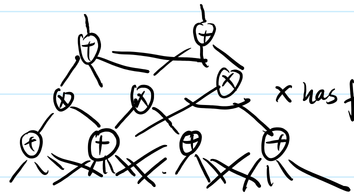
$$L(f_1, \dots, f_m) := \min_{\substack{C \text{ computing} \\ f_1, \dots, f_m \\ \text{simultaneously}}} (\#\text{ of multiplication gates of } C)$$

Assume the addition gates can compute $\sum c_i f_i$ now



<u>Def:</u> A <u>quadratic circuit</u> is a multi-output circuit of un<u>bounded fan-in</u> that has the form $\sum(\sum \times \sum)$.

So it looks like:



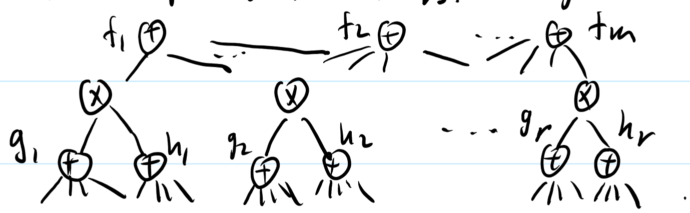
$\times$ has fan-in two.

We would like to show that to compute (one or several) quadratic forms with minimum $\times$ gates, it suffices to use quadratic circuits.

<u>Thm 1</u> (Strassen): For quadratic forms $f_1, \dots, f_m$,

$$L(f_1, \dots, f_m) = \min\left\{ r : \begin{array}{l} \exists \text{ linear forms } g_1, \dots, g_r, h_1, \dots, h_r \\ \text{such that } f_1, \dots, f_m \in \operatorname{span}\{g_1 h_1, \dots, g_r h_r\} \end{array} \right\}.$$

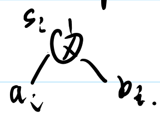
<u>Pf:</u> $\leq$: Suppose $f_1, \dots, f_m \in \operatorname{span}\{g_1 h_1, \dots, g_r h_r\}$. Then just build the quadratic circuit



$\#$ multiplication gates $= r$.

$\geq$: Let C be a (not necessarily quadratic) circuit computing $f_1, \dots, f_m$ with $r = L(f_1, \dots, f_m)$ multiplication gates.

Let $s_1, \dots, s_r$ be the outputs of the r multiplication gates s.t. s_i depends only on $s_1, \dots, s_{i-1}$. (via a topological sort) on $s_1, \dots, s_{i-1}$. (via a topological sort)

Suppose $s_i = a_i \times b_i$. 

Recall $\mathrm{Hom}_k(f)$ denotes the homogeneous degree-k part of f.

<u>Claim</u>: $\mathrm{Hom}_2(s_i) \in \mathrm{span}\{\mathrm{Hom}_1(a_1)\cdot\mathrm{Hom}_1(b_1), \dots, \mathrm{Hom}_1(a_i)\cdot\mathrm{Hom}_1(b_i)\}$ for $i=1,\dots,r$.

This claim is proved by induction on i.

Note
$$\mathrm{Hom}_2(s_i) = \mathrm{Hom}_2(a_i \cdot b_i) = \mathrm{Hom}_0(a_i)\cdot\mathrm{Hom}_2(b_i) + \mathrm{Hom}_1(a_i)\,\mathrm{Hom}_1(b_i) + \mathrm{Hom}_2(a_i)\cdot\mathrm{Hom}_0(b_i). \quad (*)$$

Base case: $i=1$. As the computation of a_1 and b_1 does not use multiplication, $\deg(a_1), \deg(b_1) \le 1$. So $\mathrm{Hom}_2(a_1) = \mathrm{Hom}_2(b_1) = 0$.

$$\Rightarrow \mathrm{Hom}_2(s_1) = \mathrm{Hom}_1(a_1)\cdot\mathrm{Hom}_1(b_1).$$

Now consider $i>1$. By $(*)$ and the fact that $\mathrm{Hom}_0(a_i), \mathrm{Hom}_0(b_i) \in \mathbb{F}$, we just need to show $\mathrm{Hom}_2(a_i), \mathrm{Hom}_2(b_i) \in \mathrm{span}\{\mathrm{Hom}_1(a_1)\cdot\mathrm{Hom}_1(b_1), \dots, \mathrm{Hom}_1(a_i)\mathrm{Hom}_1(b_i)\}$.

Note a_i and b_i are linear combinations over $\mathbb{F}$ of elements in
$$\mathbb{F} \cup \{x_1, \dots, x_n\} \cup \{s_1, \dots, s_{i-1}\}.$$

So $\mathrm{Hom}_2(a_i), \mathrm{Hom}_2(b_i) \in \mathrm{span}\{s_1, \dots, s_{i-1}\}$
$$\overset{\text{induction hypothesis}}{\subseteq} \mathrm{span}\{\mathrm{Hom}_1(a_1)\cdot\mathrm{Hom}_1(b_1), \dots, \mathrm{Hom}_1(a_i)\cdot\mathrm{Hom}_1(b_i)\}$$

This proves the claim.

For $i=1,\dots,m$, the output f_i of C is a linear combination over $\mathbb{F}$ of elements in
$$\mathbb{F} \cup \{x_1, \dots, x_n\} \cup \{s_1, \dots, s_r\}.$$

As f_i is homogeneous of degree 2, $f_i = \mathrm{Hom}_2(f_i) \in \mathrm{span}\{\mathrm{Hom}_2(s_1), \dots, \mathrm{Hom}_2(s_r)\}$
$$\overset{\text{by the claim}}{\subseteq} \mathrm{span}\{\mathrm{Hom}_1(a_1)\cdot\mathrm{Hom}_1(b_1), \dots, \mathrm{Hom}_1(a_r)\cdot\mathrm{Hom}_1(b_r)\}$$

So $r = L(f_1, \dots, f_m)$ is $\le$ RHS of Thm 1. $\square$

So $r = L(f_1, \dots, f_m)$ is $\leq$ RHS of Thm 1. $\square$

A reformulation of Thm 1 is that there always exists a quadratic circuit computing $f_1, \dots, f_m$ with $L(f_1, \dots, f_m)$ multiplications.

We say $f \in \mathbb{F}[X_1, \dots, X_n, Y_1, \dots, Y_m]$ is a <u>bilinear form</u> in (X_i) and (Y_i) if every monomial has the form $X_i Y_j$.

We say a quadratic circuit is a <u>bilinear circuit</u> in (X_i) and (Y_i) if every multiplication gate has the form $g \otimes h$ | g is a linear form in X's, h in Y's.

bilinear forms in (X_i) and (Y_i)

<u>Thm 2</u>: Suppose $f_1, \dots, f_p$ are computed by a quadratic circuit with r multiplication gates. Then they are computed by a bilinear circuit with $2m$ multiplication gates.

<u>Pf</u>: We know $f_1, \dots, f_p \in \text{span}\{g_1 h_1, \dots, g_r h_r\}$ for linear forms $g_1, \dots, g_r, h_1, \dots, h_r$ in $\{X_i\} \cup \{Y_i\}$

Each
$$(g_i h_i)(X, Y) = g_i(X,0)\, h_i(X,0) + g_i(X,0)\, h_i(0,Y) + g_i(0,Y)\, h_i(X,0) + g_i(0,Y)\, h_i(0,Y)$$
by linearity.

As $f_1, \dots, f_p$ are bilinear in (X_i) and (Y_i),
$f_1, \dots, f_p \in \text{span}\{g_i(X,0)\, h_i(0,Y),\ g_i(0,Y)\, h_i(X,0) : 1 \leq i \leq r\}$. $\square$.

Denote by $L^*(f_1, \dots, f_p)$ the # multiplication gates needed in a bilinear circuit computing $f_1, \dots, f_p$.

By Thm 2, $L(f_1, \dots, f_p) \leq L^*(f_1, \dots, f_p) \leq 2L(f_1, \dots, f_p)$.

Tensor rank.

<u>Def</u>: The <u>tensor rank</u> of a tensor $T \in \mathbb{F}^n \times \mathbb{F}^m \times \mathbb{F}^p$ is $\min_r \left\{ T = \sum_{i=1}^r a_i \otimes b_i \otimes c_i,\ \begin{matrix} a_1, \dots, a_r \in \mathbb{F}^n \\ b_1, \dots, b_r \in \mathbb{F}^m \\ c_1, \dots, c_r \in \mathbb{F}^p \end{matrix} \right\}$

pure tensor.

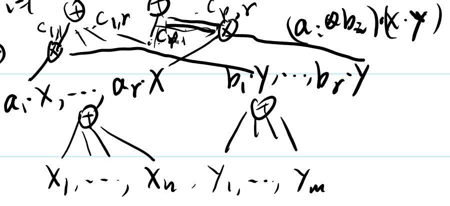
<u>Thm 3</u>: Let $f_1, \dots, f_p$ be bilinear forms in $X_1, \dots, X_n$ and $Y_1, \dots, Y_m$ with $f_k = \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} c_{ijk} X_i Y_j$

Let $T \in \mathbb{F}^n \otimes \mathbb{F}^m \otimes \mathbb{F}^p$ be a tensor defined by $T(i,j,k) = c_{ijk}$.

Then $L^*(f_1, \dots, f_p) = \text{rank}(T)$.

<u>Pf</u>: $\geq$: Let $T = \sum_{i=1}^r a_i \otimes b_i \otimes c_i$, $a_i = (a_{i,1}, \dots, a_{i,n})$, $b_i = (b_{i,1}, \dots, b_{i,m})$, $c_i = (c_{i,1}, \dots, c_{i,p})$.

k-th output is a linear combination with coefficients $c_{k,1}, \dots, c_{k,r}$. $\Rightarrow$



build a bilinear circuit with r multiplication gates.

$\leq$: reverse the proof to get a decomposition $T = \sum^r$ ...

$c_{k,1},\dots,c_{k,r}$. $x_1,\dots,x_n,\ y_1,\dots,y_m$

$\leq$: reverse the proof to get a decomposition $T=\sum_{i=1}^{r} a_i\otimes b_i\otimes c_i$ from the circuit. $\square$

Matrix Multiplication.

Consider the problem of matrix multiplication: compute $(Z_{ik})_{\substack{1\le i\le n\\ 1\le k\le p}} = (X_{ij})_{\substack{1\le i\le n\\ 1\le j\le m}}\cdot (Y_{jk})_{\substack{1\le j\le m\\ 1\le k\le p}}$

$Z_{ik}=\sum_{j=1}^{m} X_{ij}\,Y_{jk}$ is bilinear in (X_{ij}) and (Y_{jk}), $1\le i\le n$, $1\le k\le p$.

The corresponding tensor is $\langle n,m,p\rangle := \sum_{i=1}^{n}\sum_{k=1}^{p}\left(\sum_{j=1}^{m} e^X_{ij}\, e^Y_{jk}\right) e^Z_{ik} = \sum_{\substack{1\le i\le n\\ 1\le j\le m\\ 1\le k\le p}} e^X_{ij}\, e^Y_{jk}\, e^Z_{ik}$ where the (i,j)-th entry of e_{ij} is 1 and the others are 0.

$\in \mathbb{F}^{nm}\otimes\mathbb{F}^{mp}\otimes\mathbb{F}^{np}$.

Define $\omega := \inf\left(\log_n \operatorname{rank}(\langle n,n,n\rangle)\right)$, called the <u>matrix multiplication exponent</u>.

Note $n^2\le \operatorname{rank}(\langle n,n,n\rangle)\le n^3$. So $2\le\omega\le 3$.

<u>Thm</u> For any constant ε, there is an $O(n^{\omega+\varepsilon})$-time algorithm computing the product of (X_{ij}) and (Y_{jk}) ($n\times n$ matrices)
(assuming + and × take unit time)

<u>Pf</u>: By definition, $\exists\, n_0$ depending only on ε s.t. $\operatorname{rank}(\langle n_0,n_0,n_0\rangle)\le n_0^{\omega+\varepsilon}$.

View (X_{ij}) and (Y_{jk}) as $n_0\times n_0$ block matrices with block size $(n/n_0)\times(n/n_0)$

By Thm 3, $\exists$ algorithm A_0 computing $n_0\times n_0$ matrix multiplication with $\le n_0^{\omega+\varepsilon}$ multiplications and $O(1)$ additions.

Recursively multiply (X_{ij}) and (Y_{jk}): using A_0, with entries replaced by $(n/n_0)\times(n/n_0)$ blocks.

$T(n)\le n_0^{\omega+\varepsilon}\cdot T(n/n_0)+O(n^2)$.

$\Rightarrow T(n)\le n^{\omega+\varepsilon}$

Multiplication of blocks takes time $T(n/n_0)$

Addition takes time $O((n/n_0)^2)=O(n^2)$ $\square$

Similarly, time complexity of multiplying $n\times m$ and $m\times p$ matrices can be bounded in terms of $\inf_k\left(\log \operatorname{rank}\langle nk,mk,pk\rangle\right)$.

Lemma: $\operatorname{rank}(\langle n,m,p\rangle)=\operatorname{rank}\langle\sigma(n),\sigma(m),\sigma(p)\rangle$ for any permutation σ of $\{n,m,p\}$.

Pf. Recall $\langle n,m,p\rangle=\sum_{\substack{1\le i\le n\\ 1\le j\le m\\ 1\le k\le p}} e^X_{ij}\, e^Y_{jk}\, e^Z_{ik}$. We may rename e^Z_{ik} by e^Z_{ki} $\Rightarrow \langle n,m,p\rangle=\sum e^X_{ij}\,e^Y_{jk}\,e^Z_{ki}$.

Then cyclically permuting X,Y,Z sends $\langle n,m,p\rangle$ to $\langle\sigma(n),\sigma(m),\sigma(p)\rangle$ with σ cyclic permutation of $\{n,m,p\}$.

The 3 transpositions are handled by renaming $e^*_{ij}\to e^*_{ji}$, $*=X,Y,Z$, together with cyclic permutation. $\square$

So, e.g. multiplying $n\times m$ and $m\times p$ matrices, and multiplying $p\times n$ and $n\times m$ matrices have the same [illegible], together with cyclic permutation. □

So, e.g. multiplying $n\times m$ and $m\times p$ matrices, and multiplying $p\times n$ and $n\times m$ matrices have the same complexity.

<u>Thm</u>: If $\mathrm{rank}\langle n,m,p\rangle \le r$, then $\omega \le \log_{(nmp)^{1/3}} r = 3\log_{nmp} r$.

<u>Pf</u>: Note $\mathrm{rank}(T\otimes T') \le \mathrm{rank}(T)\cdot \mathrm{rank}(T')$.

And $\langle nn', mm', pp'\rangle = \langle n,m,p\rangle \otimes \langle n',m',p'\rangle$.

So $\langle nmp, nmp, nmp\rangle = \langle n,m,p\rangle \otimes \langle m,p,n\rangle \otimes \langle p,n,m\rangle$.

Then $\mathrm{rank}(\langle nmp,nmp,nmp\rangle) = \left(\mathrm{rank}(\langle n,m,p\rangle)\right)^3 \le r^3$.

$\Rightarrow \omega \le \log_{nmp} r^3 = 3\log_{nmp} r$. □

<u>Thm</u> (Strassen '69) $\mathrm{rank}(\langle 2,2,2\rangle) \le 7 \Rightarrow \omega \le \log_2 7 = 2.807\ldots$

We now know $\mathrm{rank}(\langle 2,2,2\rangle) = 7$ (over $\mathbb{F}=\mathbb{C}$). (Winograd '71).

Computing the tensor rank is NP-hard (Håstad '90).

Current record of ω (William-Xu-Xu-Zhou '23): $\omega \le 2.371552$.

Conjecture: $\omega = 2$.

Note: If the conjecture is false, then $\mathrm{rank}(\langle n,n,n\rangle) = \Omega(n^{2+\epsilon})$ for some $\epsilon > 0$.

Then by Baur-Strassen, $\sum_{1\le i,j,k\le n} X_{ij}\cdot Y_{jk} Z_{ik}$ has general circuit lower bound $\Omega(n^{2+\epsilon}) = \Omega(N^{1+\epsilon/2})$, where $N = 3n^2$ = # variables.

So a disproof of the conjecture improves the best known general circuit lower bound!

Related work: (Andrews '22): Either $\omega = 2$, or there is a nontrivial PIT algorithm for circuits where # multiplication gates is used as the complexity measure.

(Raz '13) Explicit tensor $T: [n]^r \to \mathbb{F}$ of tensor rank $\ge n^{r(1-o(1))}$, where $\omega(1) \le r \le \log n/\log\log n$, gives a superpolynomial lower bound for the size of general algebraic circuits.