Lecture 16: Lowerbounds by counting dimensions

Anup Rao

May 23, 2022

Suppose we want to compute the polynomial X^d. This can be done by repeatedly squaring X with a circuit of size $\log d$. It is easy to see that this construction is tight: each additional gate can at most double the degree of the polynomials computed by the circuit, so at least $\log d$ multiplications are needed to get degree d.

What about if we want a circuit that simultaneously computes each of the polynomials $X_1^d, X_2^d, X_3^d, \ldots, X_n^d$? One way to do this is to compute each one separately, for a total size of $n \log d$. Is this the best one can do?

You might think this should be trivially true, but there is an example from Matrix multiplication that should give you pause. By counting arguments, you can show that there is an $n \times n$ Boolean matrix A such that computing Ax requires $\Omega(n^2/\log n)$ circuit size. This suggests that computing $Ax_1, Ax_2, \ldots, Ax_n$ (where $x_1, \ldots, x_n$ are independent column vectors) should require $\Omega(n^3/\log n)$ circuit size. But this is the same as computing AX for an $n \times n$ matrix X, which can be done with algorithms (and so circuits) of complexity $O(n^{2.34})$.

Nevertheless, in our setting, we can show that there is no such magic circuit:

Theorem 1 (Bauer and Strassen)**.** *Any arithmetic circuit computing $X_1^d + X_2^d + \ldots + X_n^d$ must use $\Omega(n \log d)$ wires.*

In particular, we get the following easy corollary:

Corollary 2. *Any arithmetic circuit computing each of $X_1^d, X_2^d, \ldots, X_n^d$ must use $\Omega(n \log d)$ wires.*

There are two parts to the proof of Theorem 1. First we prove Corollary 2. Then we show that any size s circuit that computes $X_1^d + \ldots + X_n^d$ can be used to obtain a size $O(s)$ circuit computing $X_1^d, X_2^d, \ldots, X_n^d$. Actually you can show that any circuit that computes a polynomial p can be used to obtain a circuit that computes all the partial derivatives of p in similar size, which gives what we want.

To see this, we proceed by induction on the number of wires in the circuit. Suppose we have a circuit computing $p(X_1, \ldots, X_n)$ using s wires. Suppose there is a gate in the circuit that computes a function of two of the input variables, say $X_1 \cdot X_2$. Replace this gate by the variable Y, to obtain a new circuit with fewer wires that computes a function $q(X_1, \ldots, X_n, Y)$, where here $p(X_1, \ldots, X_n) = q(X_1, \ldots, X_n, X_1 \cdot X_2)$. By the chain rule for partial derivatives, we

have that if $i > 2$, then

$$\frac{\partial p}{\partial X_i} = \frac{\partial q}{\partial X_i},$$

and

$$\frac{\partial p}{\partial X_1} = \frac{\partial q}{\partial X_1} + \frac{\partial q}{\partial Y} \cdot X_2,$$
$$\frac{\partial p}{\partial X_2} = \frac{\partial q}{\partial X_2} + \frac{\partial q}{\partial Y} \cdot X_1.$$

By induction, all the partial derivatives of q can be computed by a circuit of size $O(s-1)$, so all the partial derivatives of p can be computed using a circuit of size $O(s)$. Substituting $Y = X_1 \cdot X_2$ recovers the partial derivatives of p. The same idea works for $Y = X_1 + X_2$.

Next, we prove the Corollary with a proof due to Smolensky that is based on dimension counting. Suppose that there is some circuit C with s wires that computes $X_1^d, \ldots, X_n^d$. For a polynomial $r(Y;Z)$ in variables partitioned into two lists Y, Z, we say r has degree $(d-1,1)$ if the degree of any variable in Y is at most $d-1$ and any variable in Z is at most 1. Given two lists of polynomials $p = p_1, \ldots, p_k \in \mathbb{F}[X_1, \ldots, X_n]$ and $q = q_1, \ldots, q_\ell \in \mathbb{F}[X_1, \ldots, X_n]$, define the set of polynomials

$$\tau(p||q) = \{r(p;q) : r \text{ has degree } (d-1,1)\ \}.$$

We have the following claims:

Claim 3. *If $f = g \times h$, then $\tau(f, p_1, \ldots, p_k||q_1, \ldots, q_\ell) \subseteq \tau(g, h, p_1, \ldots, p_k||q_1, \ldots, q_\ell)$.*

Indeed, in any degree $(d-1,1)$ polynomial r, $f^t = g^t h^t$, so we obtain a new polynomial r' in one additional variable that computes the same thing as r.

Claim 4. *If $f = g + h$, then $\tau(f, p_1, \ldots, p_k||q_1 \ldots, q_\ell) \subseteq \tau(g, h, p_1, \ldots, p_k||q_1 \ldots q_\ell)$.*

Again, in any degree $(d-1,1)$ polynomial r, we can replace $f^t = (g+h)^t$ and again obtain a degree $(d-1,1)$ polynomial q' that computes the same thing as q.

Claim 5. $\tau(f, f, p_1, \ldots, p_k||q_1, \ldots, q_\ell) \subseteq \tau(f, p_1, \ldots, p_k||f^d, q_1, \ldots, q_\ell)$.

To prove this claim, note that since r has access to two copies of f, it can actually compute f^t for any $t \leq 2(d-1)$. To simulate this new computation, it is enough to have access to f^d with degree up to 1 and f with degree up to $d-1$.

Claim 6. *If a circuit C uses s wires to compute polynomials $p_1, \ldots, p_k$ at k distinct gates, then there exist at most s polynomials $q_1, \ldots, q_s$ such that $\tau(p_1, \ldots, p_k||) \subseteq \tau(X_1, \ldots, X_n||q_1, \ldots, q_s)$.*

Proof We prove the claim inductively. Suppose we are working with the space $\tau(p_1, \ldots, p_k || q_1, \ldots, q_r)$, where each p_i is a polynomial computed at a distinct gate of the circuit. Suppose p_1 is a polynomial of maximal depth in the circuit (namely it corresponds to the gate that is farthest away from an input variable). If p_1 is equal to X_i for some i, then we are done, since all p_i's must be at depth 0. Otherwise, by Claims 3 or 4, we get that $\tau(p_1, \ldots, p_k || q_1, \ldots, q_r) \subseteq \tau(g, h, p_2, \ldots, p_k || q_1, \ldots, q_r)$, where g, h are polynomials computed at gates of lower depth in the circuit. g, h may correspond to the same gate as one of the p_i's, in which case we apply Claim 5 (possibly twice) to take care of this duplication. If we repeatedly apply this argument, note that we can apply Claim 5 at most s times. This proves the claim. ■

All that remains is to count dimensions. Note that $\tau(X_1^d, \ldots, X_n^d, X_1, \ldots, X_n ||)$ is simply the set of all polynomials in $X_1, \ldots, X_n$ whose degree in each variable is less than d^2. The dimension of this space is thus $(d^2)^n = d^{2n}$. On the other hand, $\tau(X_1, \ldots, X_n || q_1, \ldots, q_s)$ is spanned by a set of at most $2^s \cdot d^n$ polynomials. Thus $2^s \cdot d^n \geq d^{2n} \Rightarrow s \geq n \log d$.