I/O complexity

Goal: Analyze optimality of algorithms w.r.t. data movement

W - work, T - time
Q - reads + writes
I - \( W/Q \) (intensity)
\( \gamma \) - size of cache

Roofline model

\[ Q > Q_{cm} \Rightarrow I > I_{cm} \]

Ordinary MMM

\( \begin{array}{c}
A \\
\hline
B \\
\hline
C
\end{array} \)

\( \begin{array}{c}
A \\
\hline
B \\
\hline
C
\end{array} \)

Per block of C:
- read \( b \times n \) panel of A, \( n \times b \) panel of B
- \( \frac{n}{b} \) blocks of C

I/O cost:
- \( \frac{2n^3}{b} + \frac{2n^2}{b} \) read C
- \( \frac{n}{b} \) read A + \( \frac{n}{b} \) read B

I/O cost:
- \( 4n^2 \)
CDAG Model

Vertex: atom of data
Edge: dependency

MVM:
\[
\begin{bmatrix}
  c_1 \\
  c_2
\end{bmatrix} =
\begin{bmatrix}
  a_{11} & a_{12} \\
  a_{21} & a_{22}
\end{bmatrix}
\begin{bmatrix}
  b_1 \\
  b_2
\end{bmatrix}
\]

Red-blue pebble game

Start: blue pebbles on inputs

R₁ - Input: place a red pebble on a vertex w/ a blue pebble

R₂ - Output: place a blue pebble on a vertex w/ a red pebble

R₃ - Computation: place a red pebble on a vertex whose immediate predecessors have red pebbles

R₄ - Delete any pebble

End: blue pebbles on outputs

Game Moves:

\[ R₁(a_{11}) R₁(b₁) R₃(*₁₁) R₁(a_{12}) R₄(a_{11}) R₃(*₁₂) R₄(b₁) R₄(a_{12}) R₁(a_{12}) R₁(b₁) R₃(*₂₁) \ldots \]
S-span theorem (simplified)
- Partition game into segments
  - Segment 0 starts at beginning
  - If \( R_i \) or \( R_2 \) in segment \( \Rightarrow \) end current segment, begin next.

- Suppose \( h \) such segments.
- Index by \( i \in \mathbb{Z}, \ldots, h-1 \)

\[
Q_i = \sum_{i=0}^{h-1} \sigma_i \Rightarrow Q \geq \sigma (h-1)
\]

\[
W_i \leq W_{\text{max}}
\]

\[
\text{upper bound on any possible } W_i \Rightarrow h \geq \frac{W}{W_{\text{max}}}
\]

\[
Q \geq \sigma \left( \frac{W}{W_{\text{max}}} - 1 \right)
\]
Geometric problem:
Multiplication as points in 3D space
\[ C_{xy} = a_{xz} b_{zy} \]

Project set of points in \( x \) dimension:
- elem. of \( B \)
- \( y \) dimension: elem. \( a \)
- \( z \) dimension: elem. \( c \)

3D Loomis-Whitney Inequality
Let \( V \) be a finite set w/ elem in \( \mathbb{Z}^3 \)
Let \( V_x, V_y, V_z \) be orthogonal projections of \( V \) onto the coordinate planes.
\[ |V| \leq \sqrt{|V_x| \cdot |V_y| \cdot |V_z|} \]
Applying Loomis-Whitney Inequality (I Ioay et al 2004)

\[ W_{\text{max}} \leq \sqrt{N_A N_B N_C} \leq \sqrt{8\delta} \]

Substitute
\[ N_A, N_B, N_C \leq 2\delta \]

\[ Q_{\text{mm}} \geq \sigma \left( \frac{n^3}{\delta^{3/2}} - 1 \right) = \frac{n^3}{\sqrt{8\delta}} - \sigma \]

Can we improve the lower bound?

Observation 1: We can change the problem
Assume all computation is performed with FMA\(_5\)
(claim: any graph played on an MMN CDAG can be transformed into one that multiplies elements of \(a, b, c\) one adds then to an element of \(c\) immediately)

FMA: \(c_{ij} = a_{ip} \cdot b_{pj}\)
\[ \uparrow \quad \uparrow \quad \uparrow \]
all inputs

\[ N_A + N_B + N_C \leq 2\delta \]

Find constrained global maximum of \(\sqrt{N_A N_B N_C}\)

\[ \Rightarrow N_A = N_B = N_C = \frac{2\delta}{3} \Rightarrow Q_{\text{mm}} \geq \frac{3\sqrt{3}}{2\sqrt{2}} \frac{n^3}{\sqrt{8\delta}} - \sigma \]

Observation 2:
Number of \(R_1 + R_2\) during a segment is arbitrary
let \(x = \# \text{of } R_1's + R_2's \text{ in a segment}\)
- fixed variable

\[ N_A + N_B + N_C \leq \delta + x \]

\[ W_{\text{max}} \leq \left( \frac{\sqrt{8\delta + x}}{3\sqrt{3}} \right) \]

\[ Q \geq x \left( \frac{W}{W_{\text{max}}} - 1 \right) \]

Let \(x = 2\delta\)

\[ W_{\text{max}} = \sigma \sqrt{8\delta} \]

\[ Q_{\text{mm}} \geq \frac{2n^3}{3\sqrt{2}} - 2\sigma \]
Recomputation Example - Neural Networks

Model:
\[ f(x) = a \circ b \circ c(x) \]
\[ w_a, w_b, w_c : \mathbb{R}^n \rightarrow \mathbb{R}^n \]

Goal: Fit model to data. Minimize loss function \( L(f) \)

Training: Stochastic gradient descent (SGD)
- pick random \( x \)
- calculate gradient at \( x \) w.r.t. model parameters
  \[ \frac{\partial L}{\partial w_a} \]
- modify parameters based on gradient

Chain rule:
\[ \frac{\partial L}{\partial w_a} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial w_a} \]
\[ \frac{\partial L}{\partial w_b} = \left( \frac{\partial L}{\partial b} \cdot \frac{\partial b}{\partial w_b} \right) \frac{\partial b}{\partial w_b} \]
\[ \frac{\partial L}{\partial w_c} = \left( \frac{\partial L}{\partial c} \cdot \frac{\partial c}{\partial w_c} \right) \frac{\partial c}{\partial w_c} \]

Tradeoffs:

Method 1: Complete Memoization (assume \( h \) stages)
- memory \( O(h) \)
- time \( O(h) \)

Method 2: Complete recomputation
- memory \( O(1) \)
- time \( O(h^2) \)

Method 3: Partial memoization/recomputation
- Recompute every \( k \) stages
- memory \( O(h/k) \)
- time \( O(h \cdot k) \)