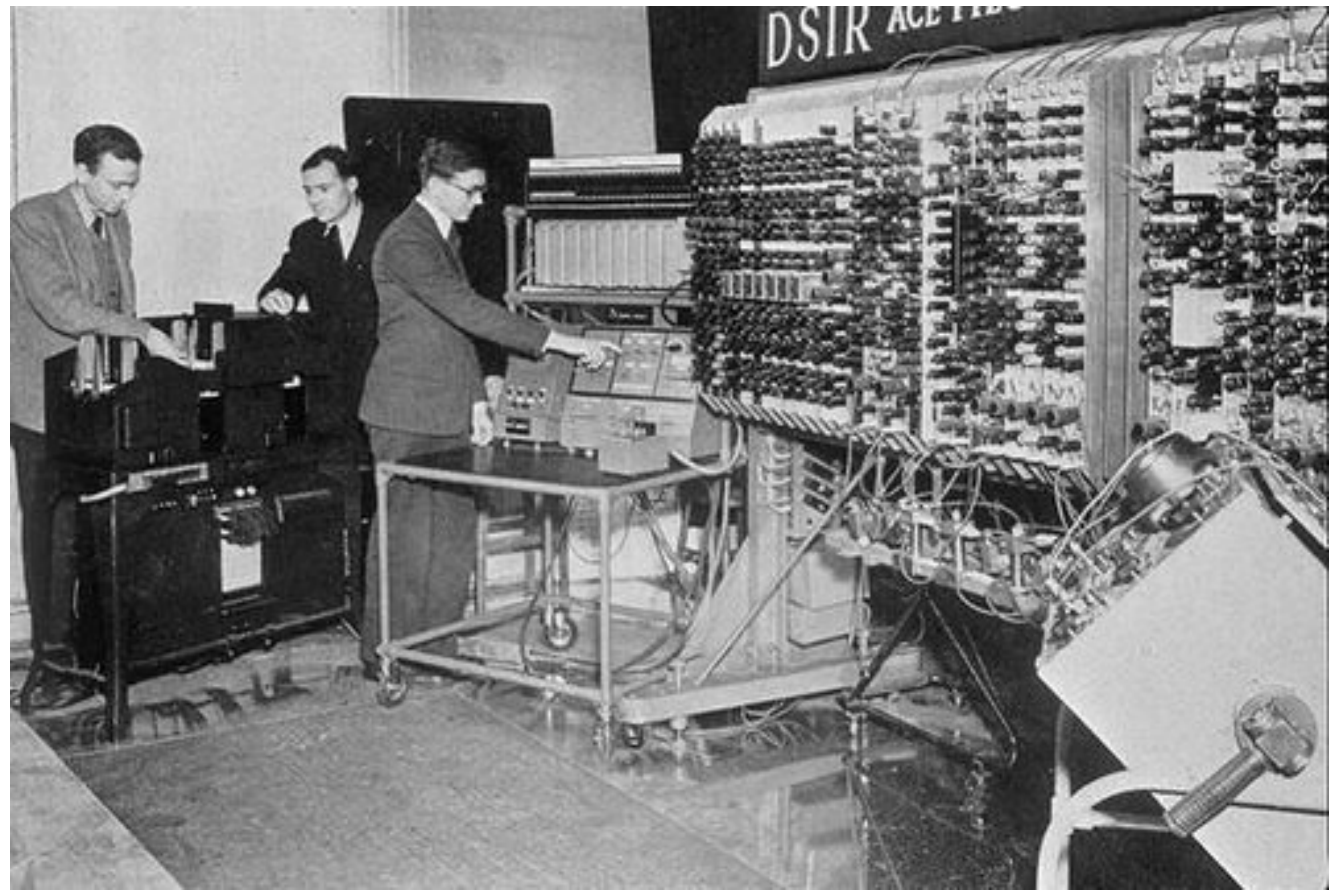
Efficient Attention

Lessons for theory-driven algorithm design

CS in two questions

1. Can we compute it?
2. How fast?

ChatGPT

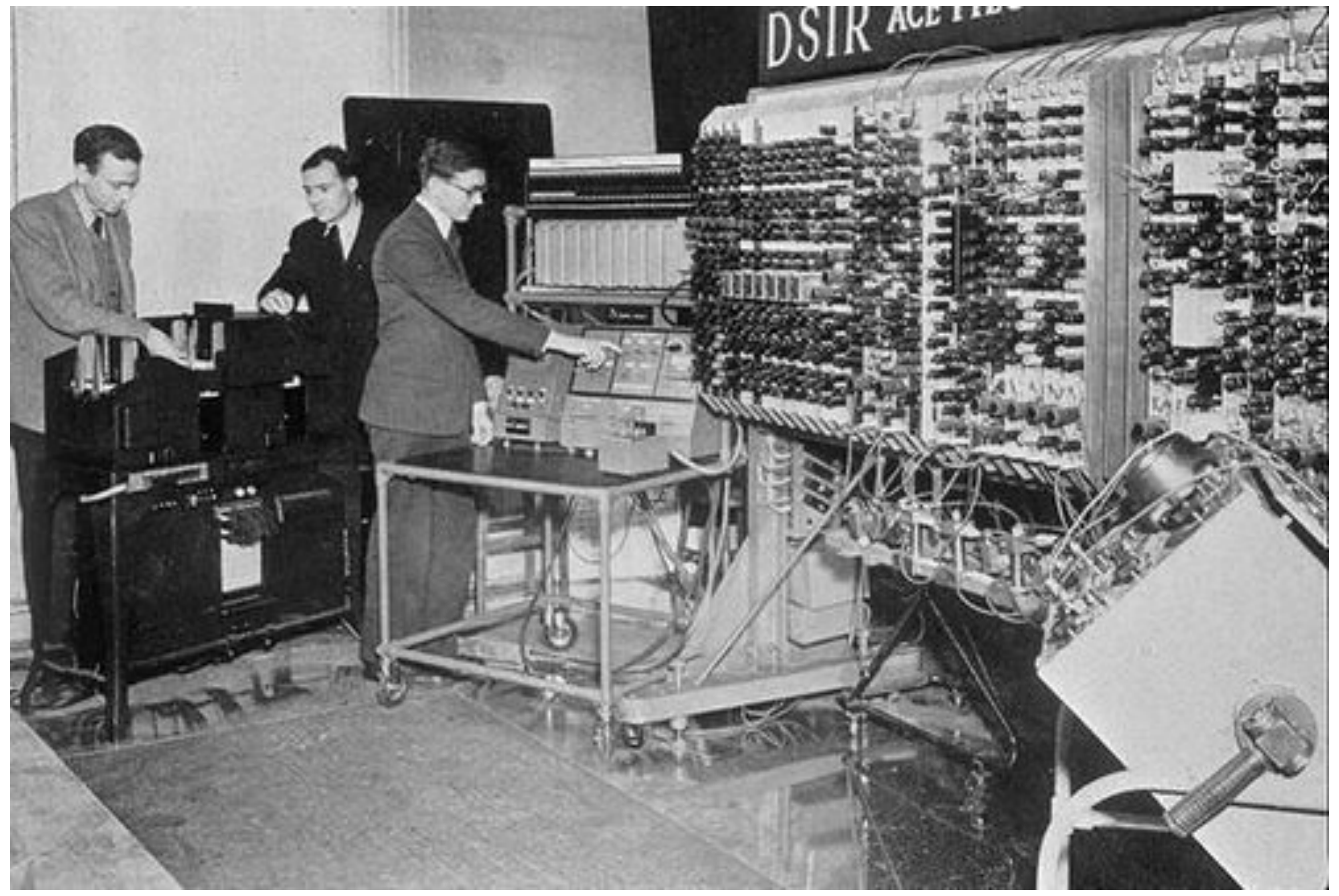


CS in two questions

1. Can we compute it?
2. How fast?

This talk

ChatGPT



Low-rank Thinning

With Lester Mackey, Albert Gong, Abhishek Shetty & Raaz Dwivedi

Attention approximation

- Attention: runtime quadratic in sequence length

Attention approximation

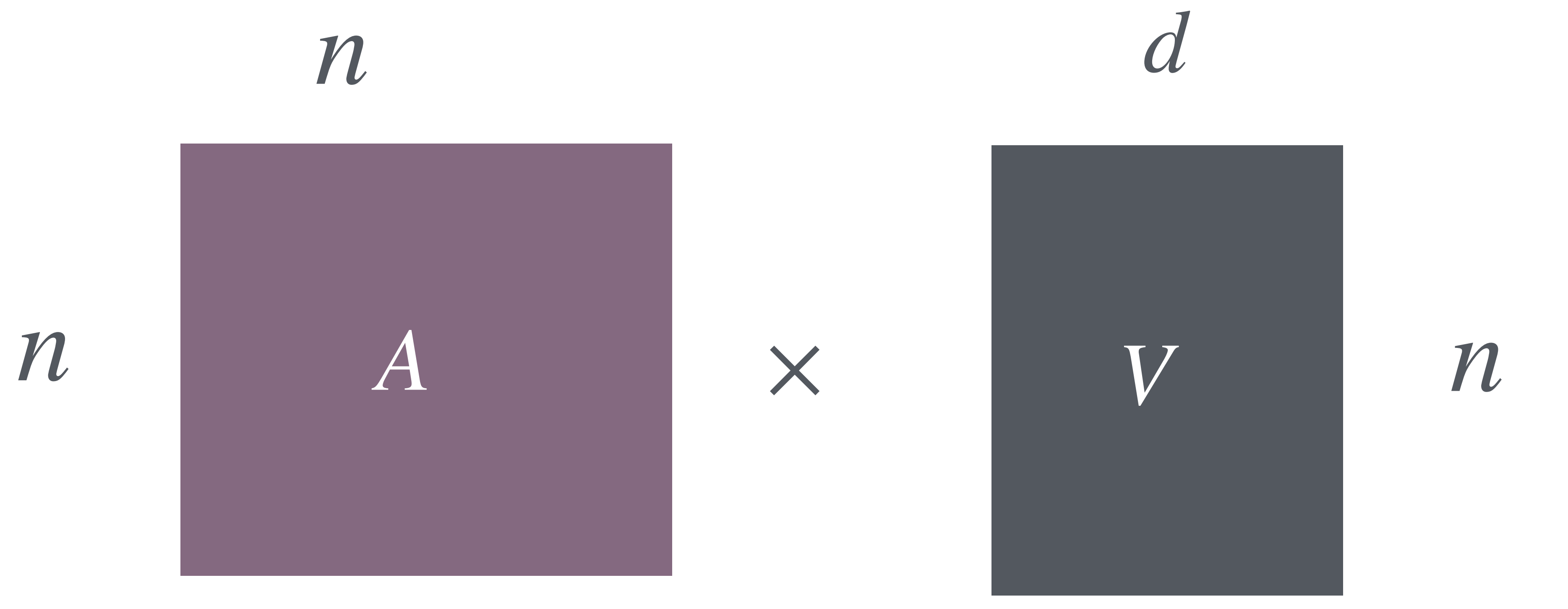
- Attention: runtime quadratic in sequence length
- Method: perform partial attention computation

Attention approximation

- Attention: runtime quadratic in sequence length
- Method: perform partial attention computation
- Goals: low error (similar quality to full attention) and fast

Thinformer

Exact attention $\Theta(n^2 d)$



Thinformer $\Theta(n n_{out} d)$

Thinformer

Attention($Q, K, V \in \mathbb{R}^{n\times d}$) = $D^{-1}AV$, where

$A = \exp(QK^T/\sqrt{d}) \quad D = \mathrm{diag}(A1_n)$

Thinformer

Attention$(Q, K, V \in \mathbb{R}^{n\times d}) = D^{-1}AV$, where

$A = \exp(QK^T/\sqrt{d}) \quad D = \mathrm{diag}(A1_n)$

Thinformer$(Q, K, V \in \mathbb{R}^{n\times d})$:

$\tilde{K}, \tilde{V}$ <- THIN(K, V) // subselect n_{out} points

Then $D^{-1}\tilde{A}\tilde{V}$, where

$\tilde{A} = \exp(Q\tilde{K}^T/\sqrt{d})$

(Dwivedi and Mackey'21, '22, Shetty-Dwivedi-Mackey '22)**

** a few changes

Part 1: error guarantees

Approximate matrix multiplication

$$\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times d}$$

$$(\mathbf{A}\mathbf{B}^\top)_{ij} = \langle \mathbf{A}_{i;.}, \mathbf{B}^\top_{.;j} \rangle = \sum_{k=1}^{n} \mathbf{A}_{ik}\mathbf{B}^\top_{kj} \approx \sum_{k=1}^{n_{\text{out}}} \mathbf{A}_{ik}\mathbf{B}^\top_{kj}.$$

How do we select n_{out} points?

Problem setup: thinning 1-d

Data points $X \triangleq [x_1, \ldots, x_n] \in \mathbb{R}^n$

Problem setup: thinning 1-d

Data points $X \triangleq [x_1, \ldots, x_n] \in \mathbb{R}^n$

Goal: Select n_{out} points "representative" of X

How do we measure? Difference in means

Problem setup: thinning 1-d

Data points $X \triangleq [x_1, \ldots, x_n] \in \mathbb{R}^n$

Goal: Select n_{out} points "representative" of X

How do we measure? Difference in means

$p = [1/n, \ldots, 1/n] \in \mathbb{R}^n$ $q = [0, 1/n_{out}, 1/n_{out}, 0, \ldots, 0] \in \mathbb{R}^n$

$\mathbb{E}_p[x] - \mathbb{E}_q[x] = X^T p - X^T q$

Guarantees: 1-d case

Assume from our thinning algorithm $X^T p - X^T q$ is sub-Gaussian: (Dwivedi and Mackey '21, '22)

$$\mathbb{E}\left[\exp(t(X^T p - X^T q))\right] \leq \exp\left(\frac{\nu^2 t^2}{2(1-\varepsilon)^2}\right) \quad (\varepsilon > 0 \;\; t > 0)$$

Guarantees: 1-d case

Assume from our thinning algorithm $X^T p - X^T q$ is sub-Gaussian: (Dwivedi and Mackey '21, '22)

$$\mathbb{E}\left[\exp(t(X^T p - X^T q))\right] \leq \exp\left(\frac{\nu^2 t^2}{2(1-\varepsilon)^2}\right) \quad (\varepsilon > 0 \;\; t > 0)$$

$$\mathbb{P}(\;X^T p - X^T q \;\; \geq t) \leq \frac{\mathbb{E}[\exp(\lambda(X^T p - X^T q))]}{\exp(\lambda t)}$$

Guarantees: 1-d case

Assume from our thinning algorithm $X^Tp - X^Tq$ is sub-Gaussian: (Dwivedi and Mackey '21, '22)

$$\mathbb{E}\left[\exp(t(X^Tp - X^Tq))\right] \le \exp\left(\frac{\nu^2 t^2}{2(1-\varepsilon)^2}\right) \quad (\varepsilon > 0 \ t > 0)$$

$$\mathbb{P}(\ X^Tp - X^Tq \ \ge t) \le \frac{\mathbb{E}[\exp(\lambda(X^Tp - X^Tq))]}{\exp(\lambda t)}$$

With prob $1 - \varepsilon$,

$$X^Tp - X^Tq \ \le \ldots$$

Problem setup: thinning high-dimensional

Data points $X \triangleq [x_1, \ldots, x_n] \in \mathbb{R}^{n \times d}$

Problem setup: thinning high-dimensional

Data points $X \triangleq [x_1, \ldots, x_n] \in \mathbb{R}^{n \times d}$

Goal: Select n_{out} points "representative" of X

How do we measure? Difference in means

Problem setup: thinning high-dimensional

Data points $X \triangleq [x_1, \ldots, x_n] \in \mathbb{R}^{n \times d}$

Goal: Select n_{out} points "representative" of X

How do we measure? Difference in means

$$p = [1/n, \ldots, 1/n] \in \mathbb{R}^n \quad q = [0, 1/n_{out}, 1/n_{out}, 0, \ldots, 0] \in \mathbb{R}^n$$

$$\mathbb{E}_p[x] - \mathbb{E}_q[x] = X^T p - X^T q \in \mathbb{R}^d$$

$$\|X^T p - X^T q\|$$

Guarantees: high-d

$$\mathbb{E}\left[\exp\left(t \parallel X^T p - X^T q \parallel_2\right)\right]$$

Guarantees: high-d

$$\mathbb{E}\left[\exp\left(t \parallel X^T p - X^T q \parallel_2\right)\right] \leq \mathbb{E}\left[\exp\left(t \cdot \frac{1}{1-\varepsilon} \max_{u \in \mathscr{C}_{\varepsilon,d}} \langle \mathbf{u}, X^T p - X^T q \rangle\right)\right]$$

$$= \mathbb{E}\left[\max_{u \in \mathscr{C}_{\varepsilon,d}} \exp\left(\frac{t}{1-\varepsilon} \langle \mathbf{u}, X^T p - X^T q \rangle\right)\right]$$

Guarantees: high-d

$$\mathbb{E}\left[\exp\left(t \parallel X^T p - X^T q \parallel_2\right)\right] \leq \mathbb{E}\left[\exp\left(t \cdot \frac{1}{1-\varepsilon} \max_{u \in \mathscr{C}_{\varepsilon,d}} \langle \mathbf{u}, X^T p - X^T q \rangle\right)\right]$$

$$= \mathbb{E}\left[\max_{u \in \mathscr{C}_{\varepsilon,d}} \exp\left(\frac{t}{1-\varepsilon} \langle \mathbf{u}, X^T p - X^T q \rangle\right)\right]$$

$$\leq \sum_{u \in \mathscr{C}_{\varepsilon,d}} \mathbb{E}\left[\exp\left(\frac{t}{1-\varepsilon} \langle \mathbf{u}, X^T p - X^T q\right)\right]$$

$$\left|\mathscr{C}_{\varepsilon,d}\right| \leq \left(1 + \frac{2}{\varepsilon}\right)^{d}$$

Guarantees: high-d

$$\mathbb{E}\left[\exp\left(t \left\| X^T p - X^T q \right\|_2\right)\right] \le \mathbb{E}\left[\exp\left(t \cdot \frac{1}{1-\varepsilon} \max_{u \in \mathscr{C}_{\varepsilon,d}} \langle \mathbf{u}, X^T p - X^T q \rangle\right)\right]$$

$$= \mathbb{E}\left[\max_{u \in \mathscr{C}_{\varepsilon,d}} \exp\left(\frac{t}{1-\varepsilon} \langle \mathbf{u}, X^T p - X^T q \rangle\right)\right]$$

$$\le \sum_{u \in \mathscr{C}_{\varepsilon,d}} \mathbb{E}\left[\exp\left(\frac{t}{1-\varepsilon} \langle \mathbf{u}, X^T p - X^T q \right)\right]$$

$$\left|\mathscr{C}_{\varepsilon,d}\right| \le \left(1 + \frac{2}{\varepsilon}\right)^d$$

$$\|X^T p - X^T q\|_2 \le d + \log(\ldots)$$

Guarantees: rank-r

$\|X^T p - X^T q\|_2 = MMD_K(p, q) \quad K = XX^T$ (linear kernel)

max-mean discrepancy

Guarantees: rank-r

$\|X^T p - X^T q\|_2 = MMD_K(p, q) \quad K = XX^T$ (linear kernel)

max-mean discrepancy

Definition (informal)

K: any symmetric PSD matrix

THIN is K-sub-Gaussian if:

$$\mathbb{E}\left[\exp\left(\langle \mathbf{u}, \mathbf{K}(\mathbf{p}-\mathbf{q})\rangle\right)\right] \leq \exp\left(\frac{\nu^2}{2}\mathbf{u}^\top \mathbf{K}\mathbf{u}\right), \quad \forall \mathbf{u} \in \mathbb{R}^n.$$

Guarantees: rank-r

Assume data is low-rank

Theorem (informal):

If THIN is K-sub-Gaussian (def), assuming K is rank r with prob $1 - \delta - \delta'$,

$$\mathrm{MMD}_K^2(\mathbf{p}_{\text{in}}, \mathbf{q}_{\text{out}}) \leq \nu^2 \left[7.4r + 2.8 \log\left(\frac{1}{\delta'}\right)\right] + \lambda_{r+1}\left(\frac{1}{n_{\text{out}}} - \frac{1}{n_{\text{in}}}\right)$$

Guarantees: rank-r

Theorem (informal):

If THIN is K-sub-Gaussian (def), assuming K is rank r with prob $1-\delta-\delta'$,

$$\mathrm{MMD}_K^2(\mathbf{p}_{\text{in}}, \mathbf{q}_{\text{out}}) \leq \nu^2 \left[7.4r + 2.8\log\left(\frac{1}{\delta'}\right)\right] + \lambda_{r+1}\left(\frac{1}{n_{\text{out}}} - \frac{1}{n_{\text{in}}}\right)$$

Guarantees: rank-r

Theorem (informal):

If THIN is K-sub-Gaussian (def), assuming K is rank r with prob $1 - \delta - \delta'$,

$$\mathrm{MMD}_K^2(\mathbf{p}_{\text{in}}, \mathbf{q}_{\text{out}}) \leq \nu^2 \left[7.4r + 2.8 \log\left(\frac{1}{\delta'}\right)\right] + \lambda_{r+1}\left(\frac{1}{n_{\text{out}}} - \frac{1}{n_{\text{in}}}\right)$$

Rank r term

Guarantees: rank-r

Theorem (informal):

If THIN is K-sub-Gaussian (def), assuming K is rank r with prob $1 - \delta - \delta'$,

$$\mathrm{MMD}_K^2(\mathbf{p}_{\mathrm{in}}, \mathbf{q}_{\mathrm{out}}) \le \nu^2 \left[7.4r + 2.8 \log\left(\frac{1}{\delta'}\right)\right] + \lambda_{r+1}\left(\frac{1}{n_{\mathrm{out}}} - \frac{1}{n_{\mathrm{in}}}\right)$$

Rank r term

Residual term

Part II: speed results

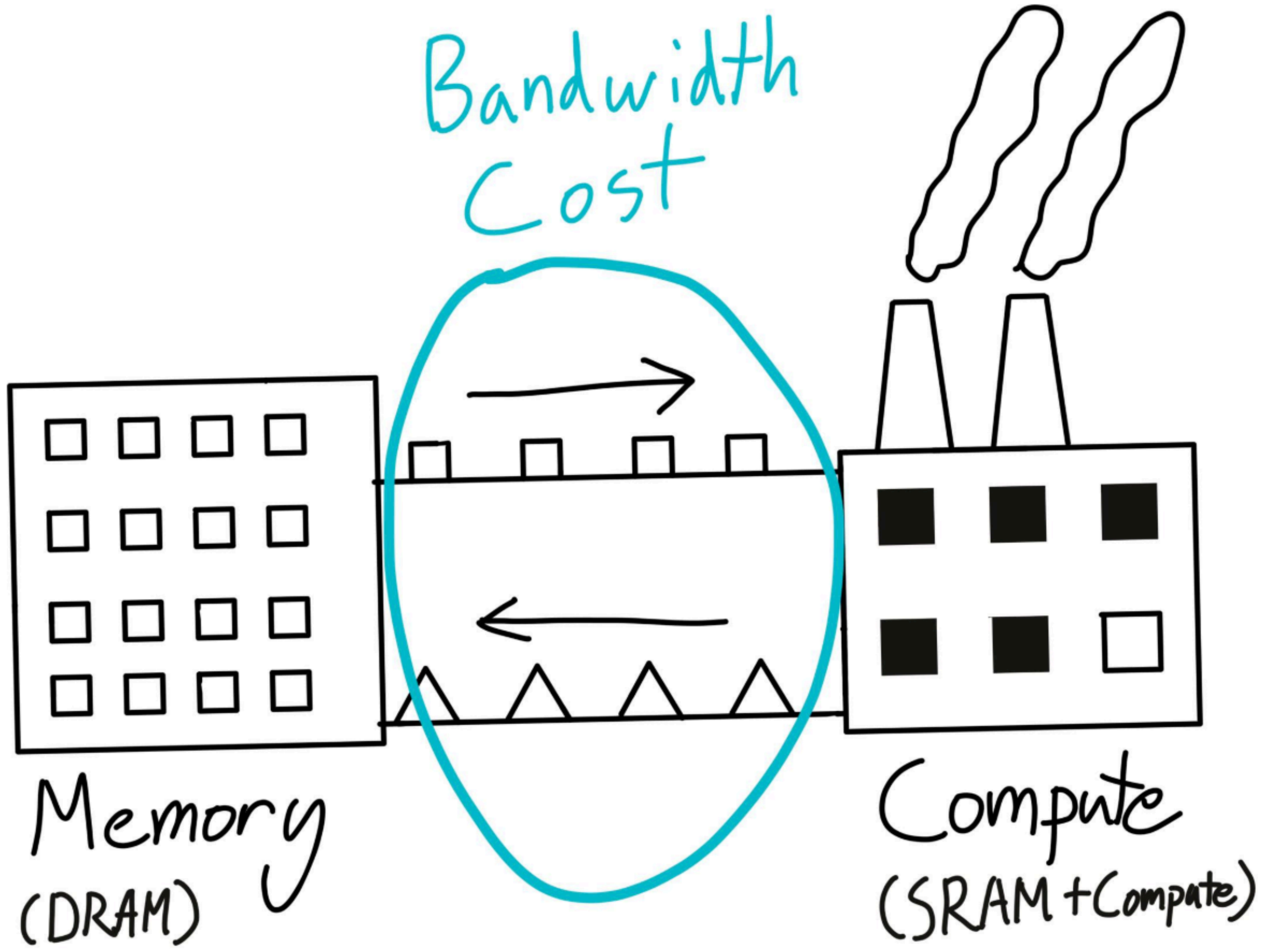
Results

ViT on Imagenet, replacing two layers at inference

Attention Algorithm	Top-1 Accuracy (%)	Layer 1 Runtime (ms)	Layer 2 Runtime (ms)
Exact	82.55 ± 0.00	18.48 ± 0.12	1.40 ± 0.01
Performer	80.56 ± 0.30	2.54 ± 0.01	0.60 ± 0.01
Reformer	81.47 ± 0.06	7.84 ± 0.03	1.53 ± 0.01
KDEformer	82.00 ± 0.07	5.39 ± 0.03	2.28 ± 0.03
Scatterbrain	82.05 ± 0.08	6.86 ± 0.02	1.55 ± 0.03
Thinformer (Ours)	82.18 ± 0.05	2.06 ± 0.01	0.54 ± 0.00

Hardware lessons

- Moving data >> performing GPU computations
- Matrix multiplication << other operations
- Non-parallel and element wise operations: slow



(Horace He, blog post)

Efficiency hacks

- Load tensors onto device when initialized (moving data)
- Fuse elementwise operations using torch.compile() (element wise operations are slow)
- Only copy data when necessary (repeat() vs view()) (initializing/moving data)
- “Vectorise” any operation (matmuls are efficient)

Parallelism: what works

THIN(X):

If $X = 4^g$ return X

1. $X_1, X_2, X_3, X_4 <- X$ // divide input into 4

2. For X_i $i \in [1,2,3,4]$:

   S_i <- THIN(X_i)

3. $S = [S_1, S_2, S_3, S_4]$ // concatenate

4. return HALVE(S)

HALVE(X):

\\ return $X/2$ points by selecting one point from each pair based on a threshold

Subroutine parallelizable!

Parallelism: changes

HALVE(X):

\\ return $X/2$ points by selecting one point from each pair based on a threshold

Threshold adaptive based on previous rounds

Parallelism: changes

HALVE(X):

\\ return $X/2$ points by selecting one point from each pair based on a threshold

~~Threshold adaptive based on previous rounds~~

Simpler, faster threshold

Takeaways

- Bake empirical observations into assumptions (rank-r data)
- Provide guarantees for practical algorithms
- Make hardware-aware algorithms (parallelizable)
- Adapt algorithms if needed!

Bonus

For any indices $\mathcal{I} \subseteq [n]$, we further define the kernel max seminorm (KMS)

$$\|\mathbf{K}(\boldsymbol{p}_{\text{in}} - \boldsymbol{p}_{\text{out}})\|_{\mathcal{I}} \triangleq \max_{i\in\mathcal{I}} |\boldsymbol{e}_i^\top \mathbf{K}(\boldsymbol{p}_{\text{in}} - \boldsymbol{p}_{\text{out}})|. \quad (2)$$

Theorem 1 (Low-rank sub-Gaussian thinning). *Fix any $\delta' \in (0, 1)$, $r \leq n$, and $\mathcal{I} \subseteq [n]$. If $\textsc{Alg} \in \mathcal{G}_{\nu,\delta}(\mathbf{K})$, then the following bounds hold individually with probability at least $1 - \delta/2 - \delta'$:*

$$\begin{aligned}\mathrm{MMD}^2_{\mathbf{K}}(\boldsymbol{p}_{\text{in}}, \boldsymbol{p}_{\text{out}}) &\leq \nu^2 \left[e^2 r + e \log(\tfrac{1}{\delta'})\right] \\ &\quad + \lambda_{r+1}(\tfrac{1}{n_{\text{out}}} - \tfrac{1}{n_{\text{in}}}) \quad \textit{and} \qquad (3)\\ \|\mathbf{K}(\boldsymbol{p}_{\text{in}} - \boldsymbol{p}_{\text{out}})\|_{\mathcal{I}} &\leq \nu D_{\mathcal{I}} \sqrt{2 \log(\tfrac{2|\mathcal{I}|}{\delta'})}. \qquad (4)\end{aligned}$$

Here, λ_j denotes the j-th largest eigenvalue of $\mathbf{K}$, $\lambda_{n+1} \triangleq 0$, and $D_{\mathcal{I}} \triangleq \max_{i\in\mathcal{I}} \sqrt{\mathbf{K}_{ii}}$.

Suppose that, in addition, $\mathcal{X} \subset \mathbb{R}^d$ and $|\mathbf{K}_{il} - \mathbf{K}_{jl}| \leq L_{\mathbf{K}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$ for some $L_{\mathbf{K}} > 0$ and all $i, j \in \mathcal{I}$ and $l \in \mathrm{supp}(\boldsymbol{p}_{\text{in}})$. Then, with probability at least $1 - \delta/2 - \delta'$,

$$\begin{aligned}\|\mathbf{K}(\boldsymbol{p}_{\text{in}} - \boldsymbol{p}_{\text{out}})\|_{\mathcal{I}} &\leq \nu D_{\mathcal{I}} \sqrt{2 \log(4/\delta')}(1 + \tfrac{32}{\sqrt{3}}) \\ &\quad + \nu D_{\mathcal{I}}\, 32 \sqrt{\tfrac{2}{3} \operatorname{rank}(\mathbf{X}_{\mathcal{I}}) \log(\tfrac{3e^2 R_{\mathcal{I}} L_{\mathbf{K}}}{D_{\mathcal{I}}^2 \wedge (R_{\mathcal{I}} L_{\mathbf{K}})})} \qquad (5)\end{aligned}$$

for $R_{\mathcal{I}} \triangleq \max_{i\in\mathcal{I}} \|\boldsymbol{x}_i\|_2$ and $\mathbf{X}_{\mathcal{I}} \triangleq [\boldsymbol{x}_i]_{i\in\mathcal{I}}^\top$.