Adaptive Sampling for Efficient Softmax Approximation







Tavor Z. Baharav*

Broad Institute

Ryan Kang*

Stanford

Colin Sullivan*

SEI

Mo Tiwari

Stanford

Eric Luxenberg

Gridmatic

David Tse

Stanford

Mert Pilanci

Stanford

Background

- ChatGPT queries use **1GWh**[1] and cost OpenAI **$700k**[2] daily
- The **fully-connected layer w/ softmax** - σ(Ax) - is the one of most common and expensive operations in deep learning[3]
- Existing methods **make distributional assumptions** and provide **no approximation guarantees**[4]
  - *Hierarchical softmax*
  - *Differentiated softmax*
  - *Target sampling*

[1] "Q&A: UW researcher discusses just how much energy ChatGPT uses." *UW News*
[2] "The Inference Cost Of Search Disruption – Large Language Model Cost Analysis." *SemiAnalysis*
[3] "Neural information retrieval: at the end of the early years." *Information Retrieval Journal*
[4] *"Strategies for Training Large Vocabulary Neural Language Models." Proceedings of the 54th Annual Meeting of the Associationfor Computational Linguistics*

Contributions & Results

1. Propose **AdaSoftmax**, the first PAC estimation of σ(Ax)
2. Provide several **variance-reducing tricks**
3. Demonstrate the efficacy of AdaSoftmax in practice

Dataset (Model)	$\delta = 10\%$	$\delta = 5\%$	$\delta = 1\%$
EuroSAT (VGG-19)	5.18x (80.62%)	5.16x (83.00%)	4.54x (98.37%)
MNIST (Shallow CNN)	8.95x (92.25%)	8.81x (93.75%)	8.13x (99.38%)

Computer Vision

Dataset (Model)	$\delta = 10\%$	$\delta = 5\%$	$\delta = 1\%$
Wikitext (GPT-2)	8.25x (88.94%)	7.80x (93.54%)	6.67x (98.26%)
Wikitext (Llama3-7B)	14.68x (91.44%)	11.43x (94.04%)	6.88x (99.38%)
Wikitext (Mistral7B)	32.65x (89.08%)	26.37x (91.20%)	17.71x (97.77%)
Penn Treebank (GPT-2)	8.10x (81.68%)	7.50x (90.73%)	6.66x (96.79%)
Penn Treebank (Llama3-7B)	19.18x (87.82%)	16.57x (91.60%)	10.72x (97.81%)

LLMs

AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

Round-Based Elimination Algorithm

Norm. Estimation

Estimate the partition function

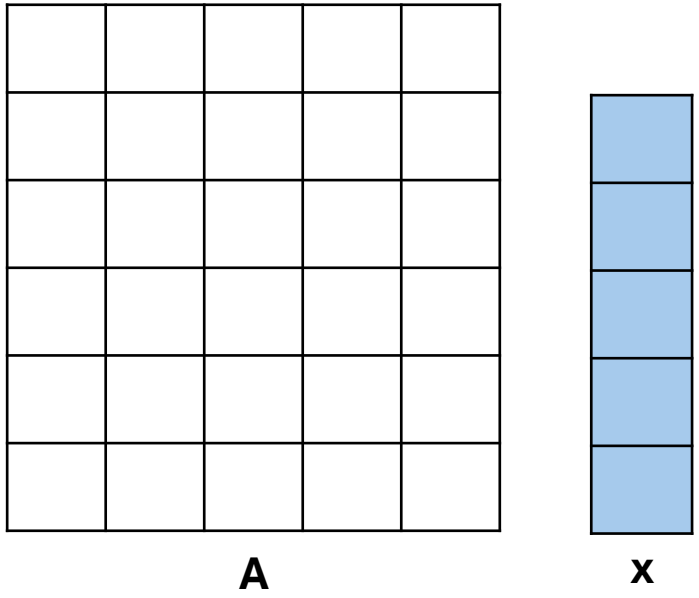
Two-Stage Estimate

AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

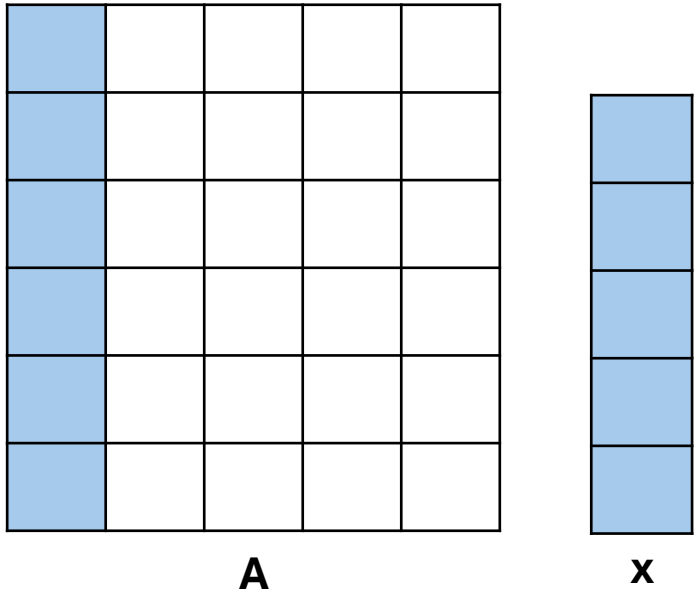
Two-Stage Estimate

AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

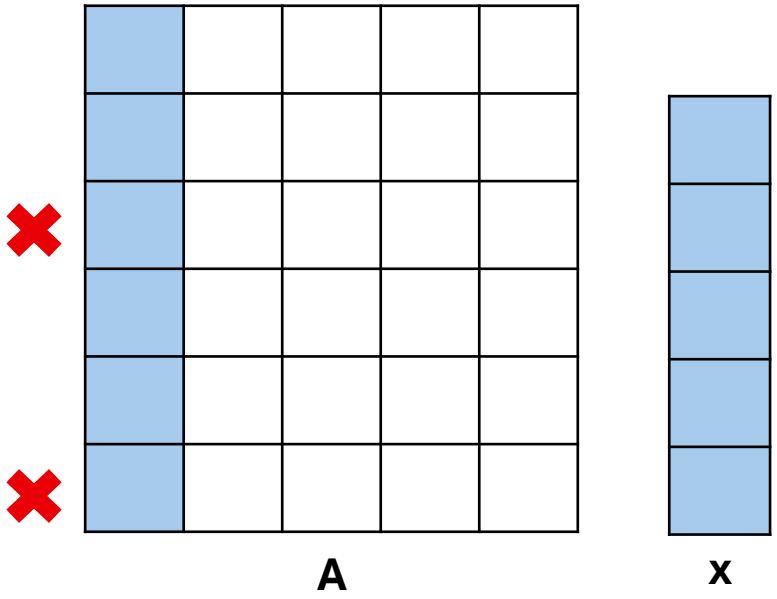
Two-Stage Estimate

AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

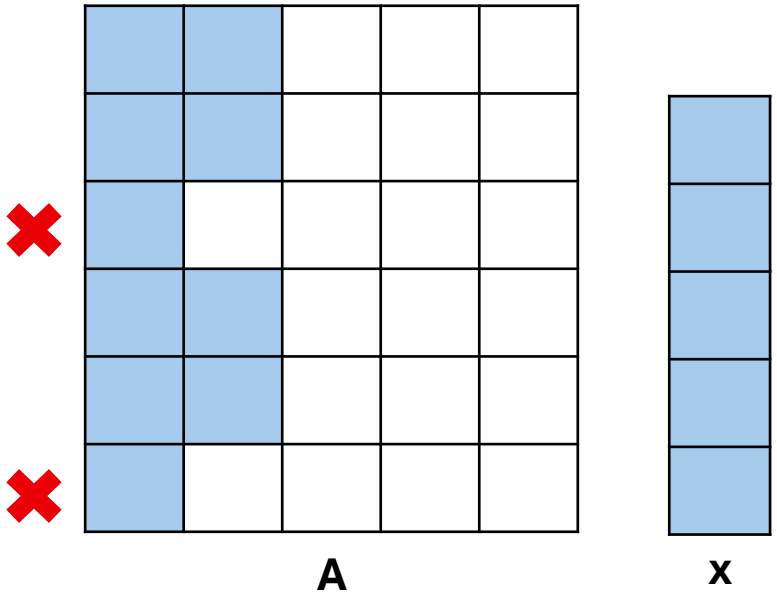
Two-Stage Estimate

AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

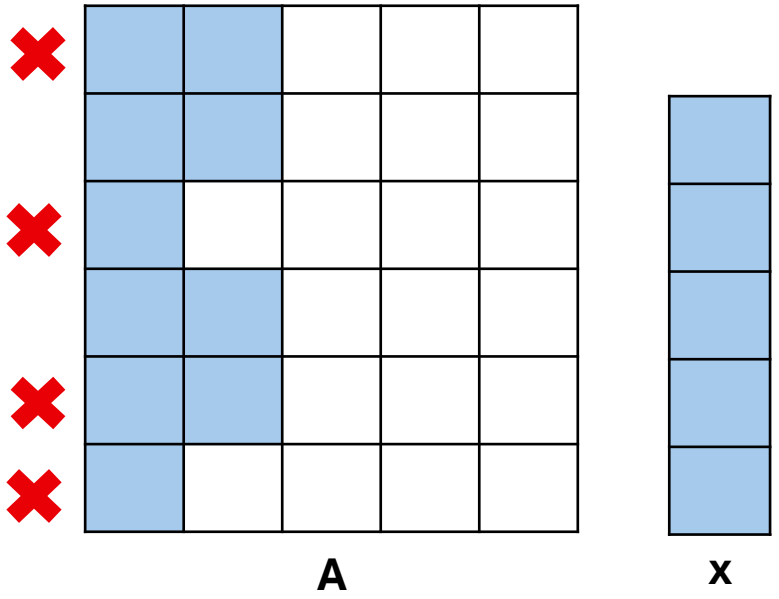
Two-Stage Estimate

AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

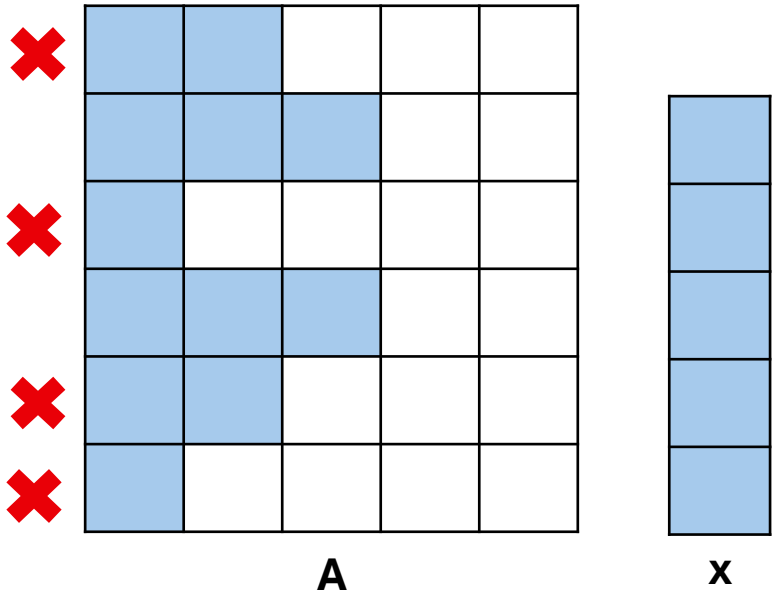
Two-Stage Estimate

AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

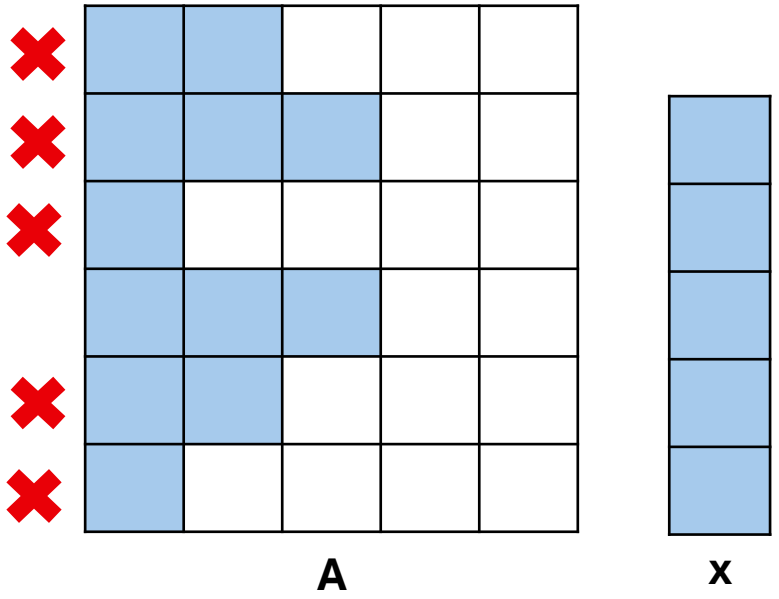
Two-Stage Estimate

AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

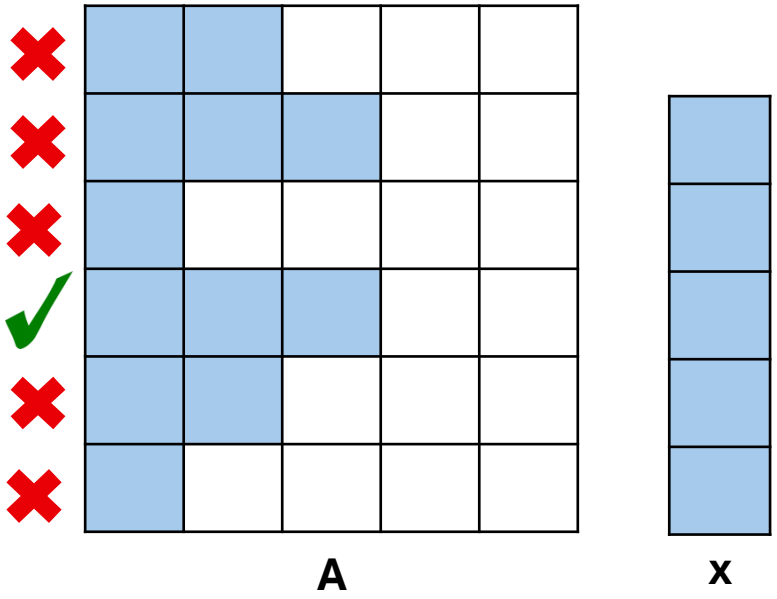
Two-Stage Estimate

AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

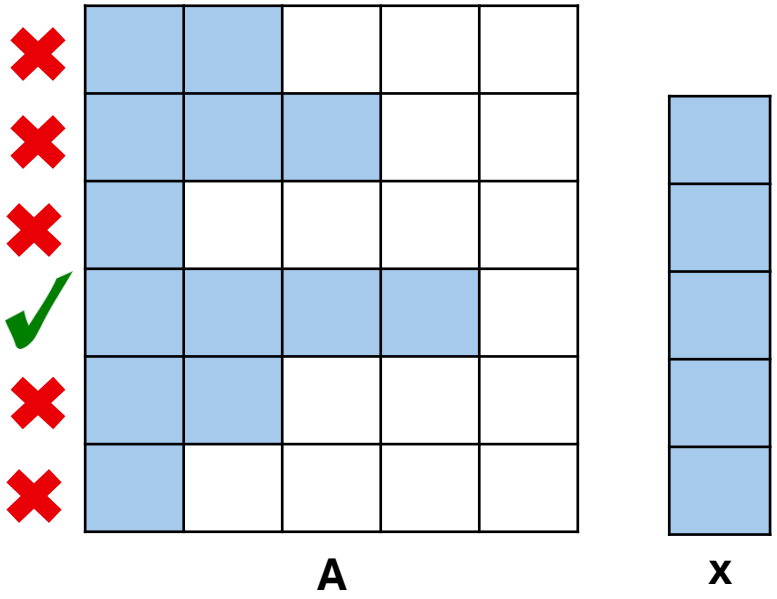
Two-Stage Estimate

AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

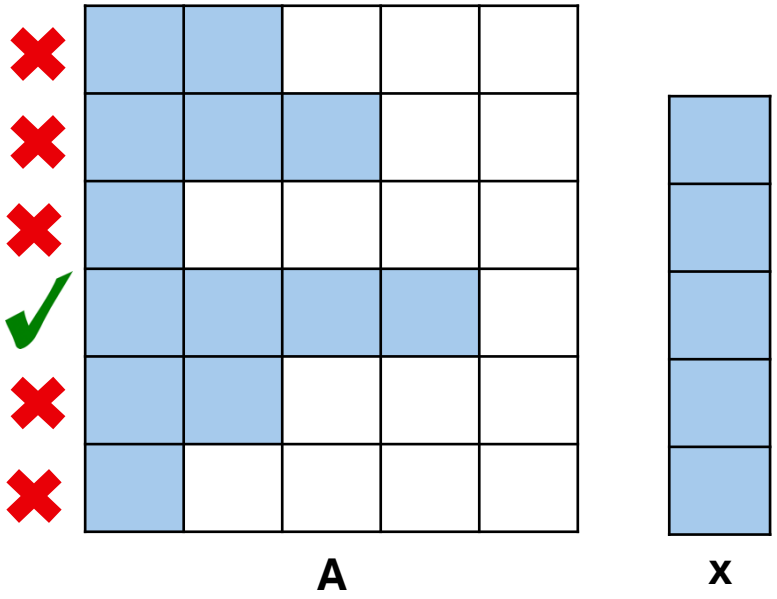
Two-Stage Estimate

AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

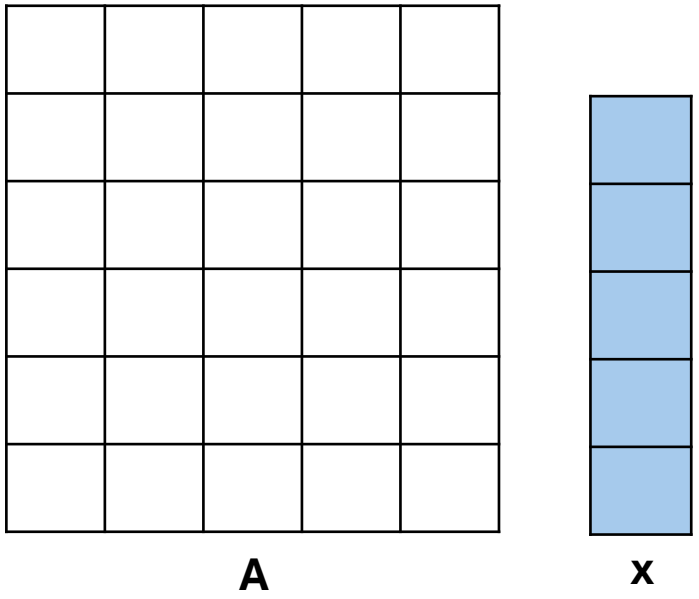
Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

Two-Stage Estimate

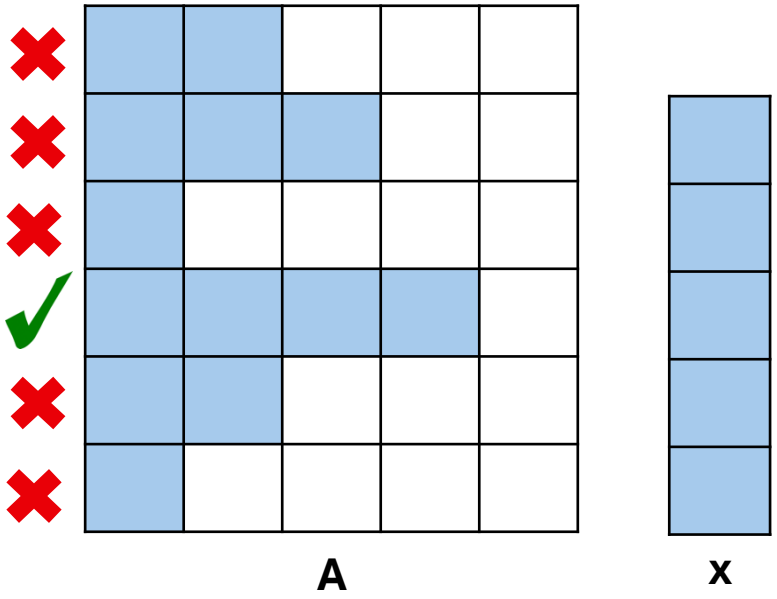


AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

Round-Based Elimination Algorithm

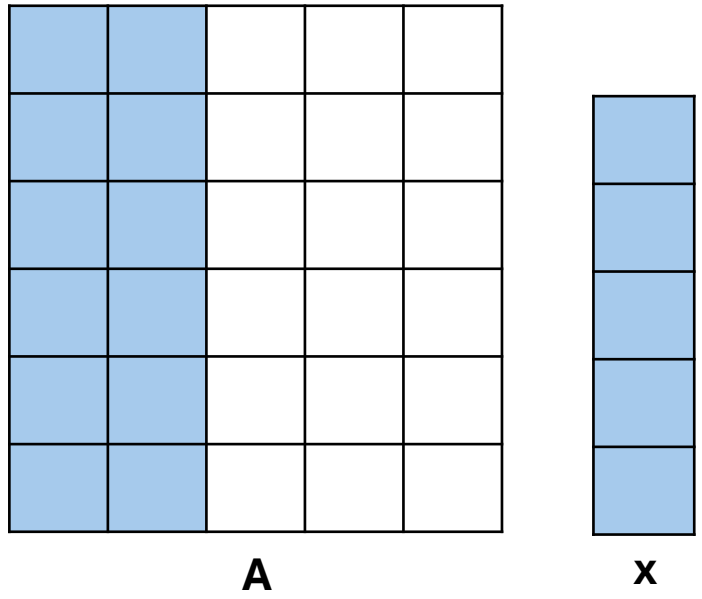


Burn-In

Norm. Estimation

Estimate the partition function

Two-Stage Estimate

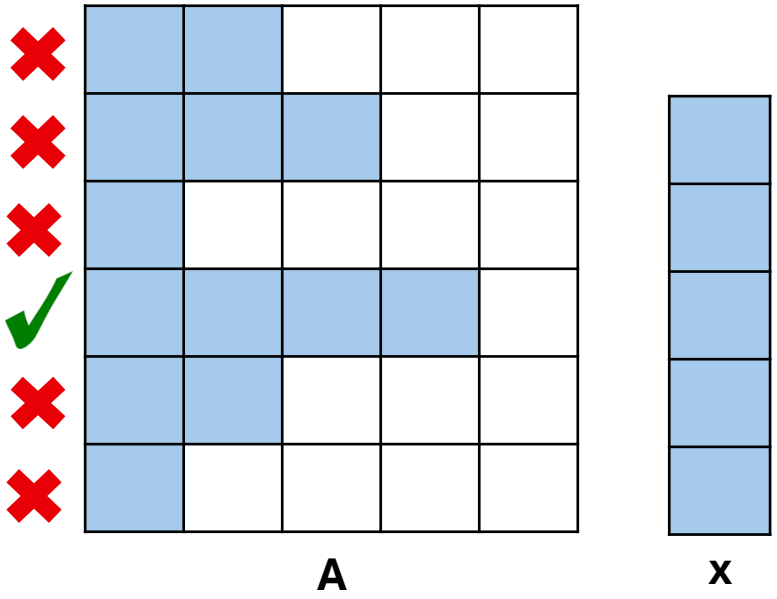


AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

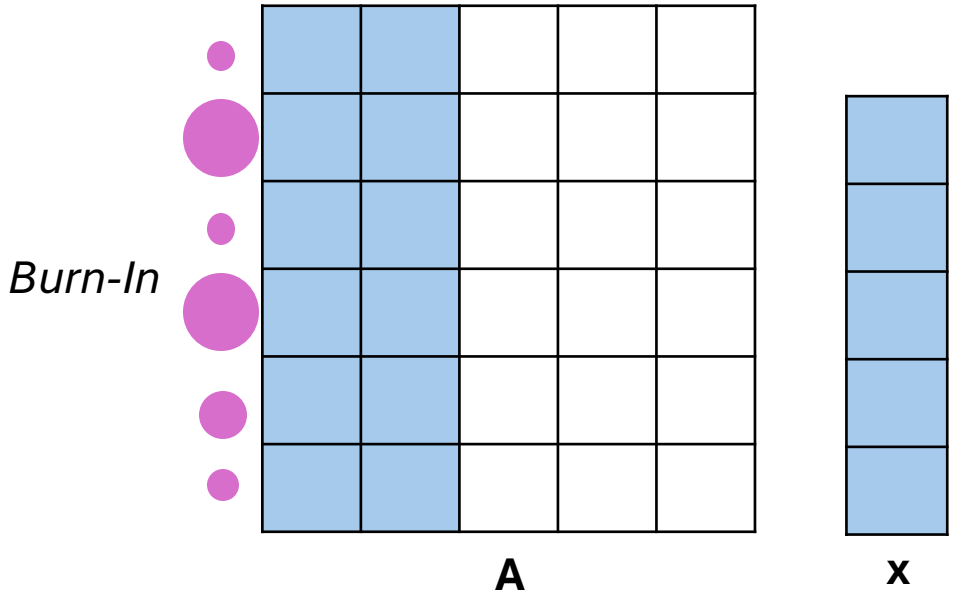
Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

Two-Stage Estimate

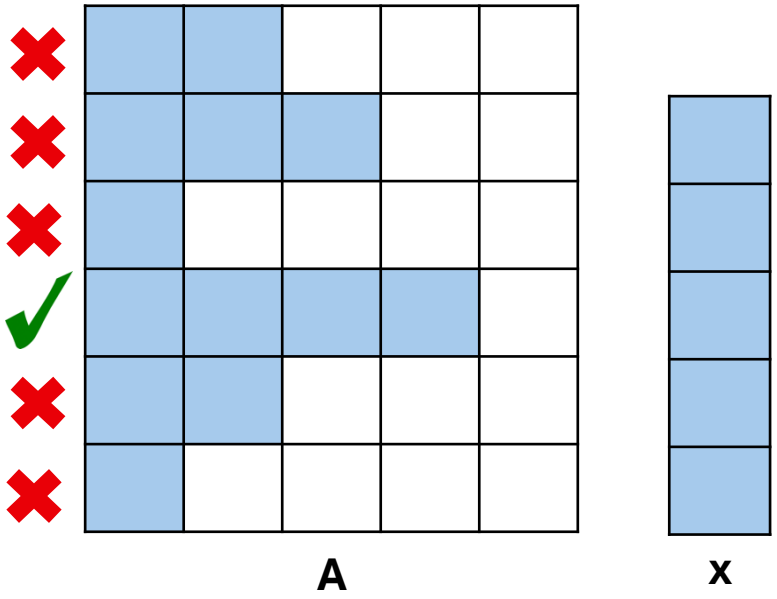


AdaSoftmax

Top-k Estimation

Identify the k row(s) with the highest logits

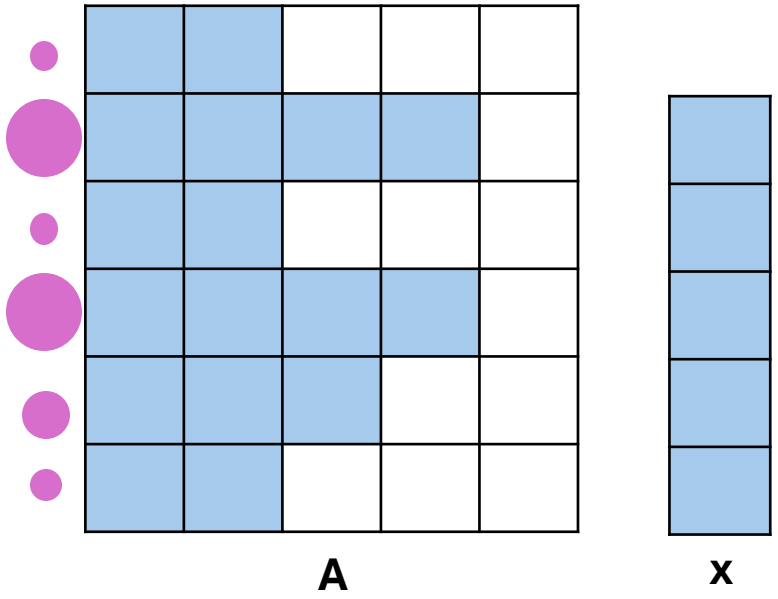
Round-Based Elimination Algorithm



Norm. Estimation

Estimate the partition function

Two-Stage Estimate



AdaSoftmax

Theorem 1. *For input $\varepsilon \in (0, 1/2)$, $\delta \in (0, 1)$, and σ satisfying Assumption 1, Algorithm 1 identifies the largest component in $\sigma_\beta(Ax)$ and estimates its value to a multiplicative accuracy of ϵ with probability at least $1 - \delta$, as in (4). On this success event, the algorithm uses T samples where*

$$T \leq C\sigma^2 \left(\beta^2 n \log\left(\frac{n}{\delta}\right) + \sum_{i=1}^{n} \frac{\log\left(\frac{n \log d}{\delta}\right)}{\Delta_i^2} + \frac{\beta^2 \log\left(\frac{n}{\delta}\right) \left(\sum_j \gamma_j\right)^2}{\epsilon \sum_j \alpha_j} + \frac{\beta^2 \log(1/\delta)}{\varepsilon^2} \right),$$

AdaSoftmax

Theorem 1. *For input $\varepsilon \in (0, 1/2)$, $\delta \in (0, 1)$, and σ satisfying Assumption 1, Algorithm 1 identifies the largest component in $\sigma_\beta(Ax)$ and estimates its value to a multiplicative accuracy of ϵ with probability at least $1 - \delta$, as in (4). On this success event, the algorithm uses T samples where*

$$T \leq C\sigma^2 \left(\beta^2 n \log\left(\frac{n}{\delta}\right) + \sum_{i=1}^{n} \frac{\log\left(\frac{n \log d}{\delta}\right)}{\Delta_i^2} + \frac{\beta^2 \log\left(\frac{n}{\delta}\right) \left(\sum_j \gamma_j\right)^2}{\epsilon \sum_j \alpha_j} + \frac{\beta^2 \log(1/\delta)}{\varepsilon^2} \right),$$

Best Row ID.

AdaSoftmax

Theorem 1. *For input $\varepsilon \in (0, 1/2)$, $\delta \in (0, 1)$, and σ satisfying Assumption 1, Algorithm 1 identifies the largest component in $\sigma_\beta(Ax)$ and estimates its value to a multiplicative accuracy of ϵ with probability at least $1 - \delta$, as in (4). On this success event, the algorithm uses T samples where*

$$T \leq C\sigma^2 \left(\beta^2 n \log\left(\frac{n}{\delta}\right) + \sum_{i=1}^{n} \frac{\log\left(\frac{n \log d}{\delta}\right)}{\Delta_i^2} + \frac{\beta^2 \log\left(\frac{n}{\delta}\right) \left(\sum_j \gamma_j\right)^2}{\epsilon \sum_j \alpha_j} + \frac{\beta^2 \log(1/\delta)}{\varepsilon^2} \right),$$

Best Row ID. *Best Row Est.*

AdaSoftmax

Theorem 1. *For input $\varepsilon \in (0, 1/2)$, $\delta \in (0, 1)$, and σ satisfying Assumption 1, Algorithm 1 identifies the largest component in $\sigma_\beta(Ax)$ and estimates its value to a multiplicative accuracy of ϵ with probability at least $1 - \delta$, as in (4). On this success event, the algorithm uses T samples where*

$$T \leq C\sigma^2 \left(\underbrace{\beta^2 n \log\left(\frac{n}{\delta}\right)}_{\text{Burn-In}} + \underbrace{\sum_{i=1}^{n} \frac{\log\left(\frac{n \log d}{\delta}\right)}{\Delta_i^2}}_{\text{Best Row ID.}} + \frac{\beta^2 \log\left(\frac{n}{\delta}\right) \left(\sum_j \gamma_j\right)^2}{\epsilon \sum_j \alpha_j} + \underbrace{\frac{\beta^2 \log(1/\delta)}{\varepsilon^2}}_{\text{Best Row Est.}} \right),$$

AdaSoftmax

Theorem 1. *For input $\varepsilon \in (0, 1/2)$, $\delta \in (0, 1)$, and σ satisfying Assumption 1, Algorithm 1 identifies the largest component in $\sigma_\beta(Ax)$ and estimates its value to a multiplicative accuracy of ϵ with probability at least $1 - \delta$, as in (4). On this success event, the algorithm uses T samples where*

$$T \leq C\sigma^2 \left(\underbrace{\beta^2 n \log\left(\frac{n}{\delta}\right)}_{\text{Burn-In}} + \underbrace{\sum_{i=1}^{n} \frac{\log\left(\frac{n \log d}{\delta}\right)}{\Delta_i^2}}_{\text{Best Row ID.}} + \underbrace{\frac{\beta^2 \log\left(\frac{n}{\delta}\right) \left(\sum_j \gamma_j\right)^2}{\epsilon \sum_j \alpha_j}}_{\text{Norm Est.}} + \underbrace{\frac{\beta^2 \log(1/\delta)}{\varepsilon^2}}_{\text{Best Row Est.}} \right),$$

AdaSoftmax

Theorem 1. *For input $\varepsilon \in (0, 1/2)$, $\delta \in (0, 1)$, and σ satisfying Assumption 1, Algorithm 1 identifies the largest component in $\sigma_\beta(Ax)$ and estimates its value to a multiplicative accuracy of ϵ with probability at least $1 - \delta$, as in (4). On this success event, the algorithm uses T samples where*

$$T \leq C\sigma^2 \left(\underbrace{\beta^2 n \log\left(\frac{n}{\delta}\right)}_{\text{Burn-In}} + \underbrace{\sum_{i=1}^{n} \frac{\log\left(\frac{n \log d}{\delta}\right)}{\Delta_i^2}}_{\text{Best Row ID.}} + \underbrace{\frac{\beta^2 \log\left(\frac{n}{\delta}\right) \left(\sum_j \gamma_j\right)^2}{\epsilon \sum_j \alpha_j}}_{\text{Norm Est.}} + \underbrace{\frac{\beta^2 \log(1/\delta)}{\varepsilon^2}}_{\text{Best Row Est.}} \right),$$

Assuming the gap is not too small, this simplifies to…

$$T \leq C\beta^2\sigma^2 \left(\log\left(\frac{n}{\delta}\right) \left(n + \varepsilon^{-1}\right) + \varepsilon^{-2} \log(1/\delta) \right)$$

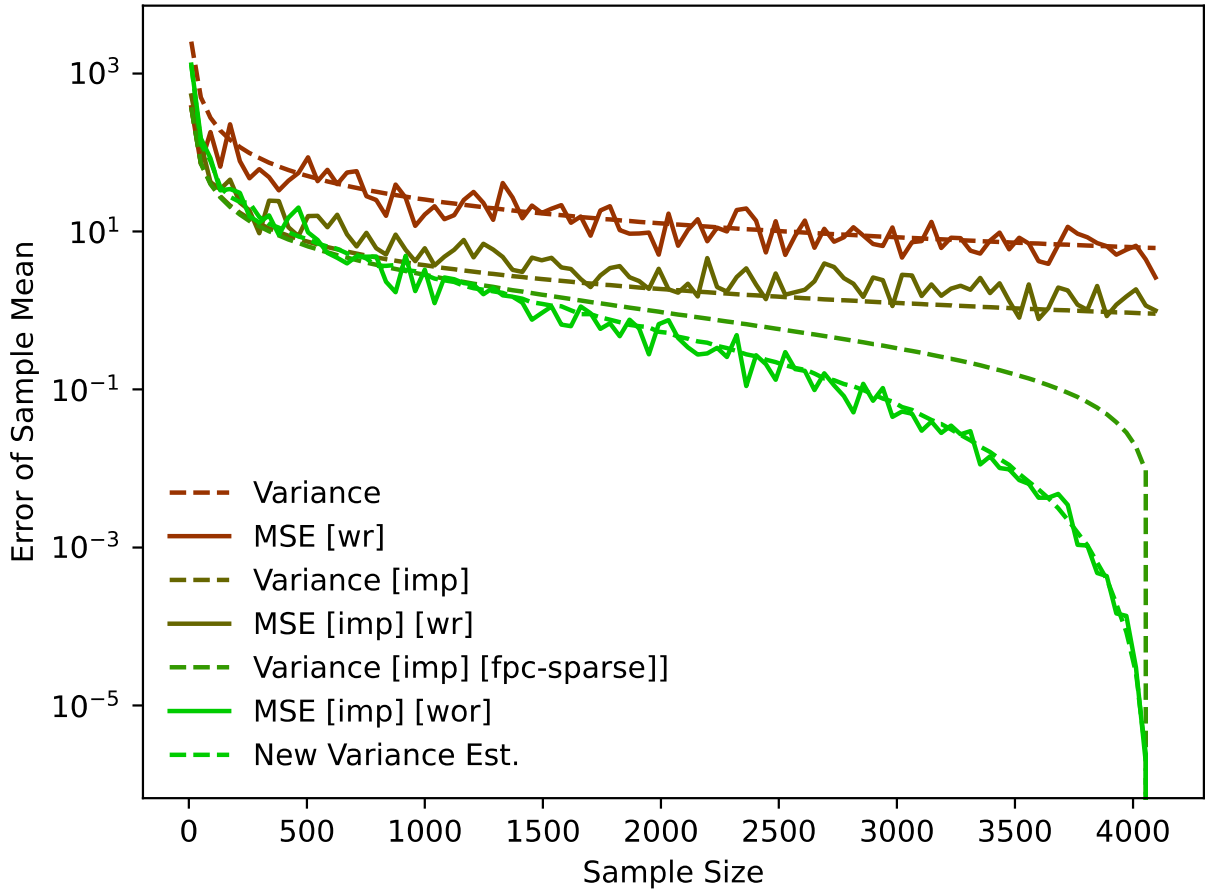
Reducing Sample Variance

Naïve Sampling
Choose columns of A at random

Importance Sampling
Weigh columns by magnitude of x

Importance Sampling + Finite-Population Correction
Adjust variance estimate w/ finite-population correction

Gumbel Sampling
Perturb logits w/ Gumbel noise and sample in descending order[1, 2]



[1] "A generalization of sampling without replacement from a finite universe." *Journal of the American Statistical Association*
[2] *"Gumbel-max trick and weighted reservoir sampling." Graduate Descent*

Future Work

"In most modern-day transformer architectures, ***memory I/O serves as the primary bottleneck****. AdaptiveSoftmax already presents an opportunity to significantly scale down the number of entries of the matrix that must be loaded at inference time, and, in the future - if memory remains the bottleneck -* ***improve model bandwidth*** *by a similar factor."*

Thank You

Come chat with us!

Poster Session 2 *on* ***Wednesday, Dec. 11th*** *from* ***4:30pm-7:30pm***

Code