CASPR: Combining Axis Preconditioners through Kronecker Approximation for Deep Learning

Sai Surya Duvvuri

Joint work with Devvrit, Rohan Anil, Cho-Jui Hsieh, Inderjit Dhillon

Optimization Algorithms in Practice

- Objective:

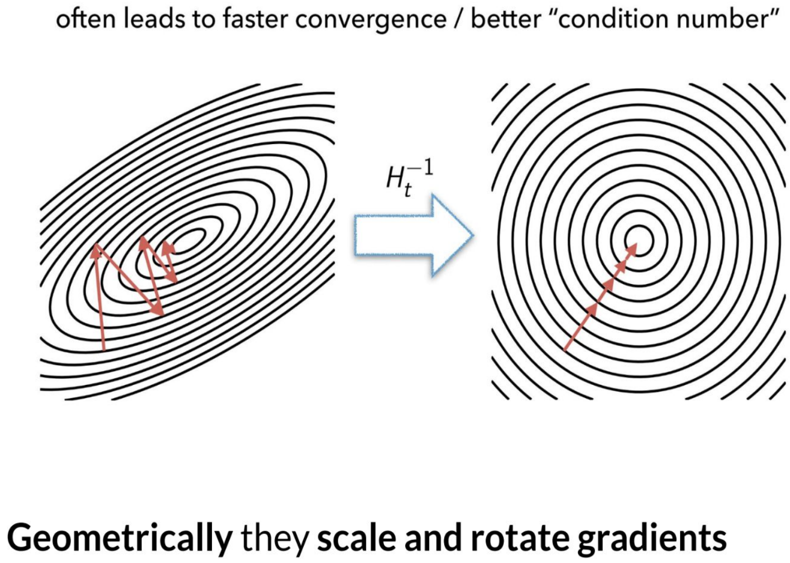
$$\min_{w} f(w), \;\; w \in \mathbb{R}^d$$

- Solution:

$$\begin{aligned} &\text{for } t = 2, \cdots, T \\ &\quad w_t = w_{t-1} - \eta_t X_t \nabla f(w_{t-1}) \end{aligned}$$

- X_t is a preconditioner matrix of size d x d.

Preconditioning



- Preconditioning typically involves inverting curvature information.

image from https://rosanneliu.com/dlctfs/dlct_210312.pdf

Optimizers for Large Deep Learning Models

- Finding preconditioner can incur high memory and compute.
- diagonal preconditioners:
  - Adam and Adagrad use coordinate wise second-moments $(g)_i^2$
  - But don't utilize cross moments $(g)_i(g)_j$
- Full-matrix Adagrad uses cross-moments → potential for faster convergence!

Full-matrix Adagrad references: “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", Duchi, Hazan, Singer'10
"Adaptive Bound Optimization for Online Convex Optimization", H. Brendan McMahan, Matthew Streeter'10

Aim

- Full matrix-Adagrad update:

$$H_t = H_{t-1} + g_t g_t^T, \qquad \rightarrow \text{memory intensive - } \mathcal{O}(d^2)$$

$$w_{t+1} = w_t - \eta H_t^{-1/2} g_t \qquad \rightarrow \text{compute intensive - } \mathcal{O}(d^3)$$

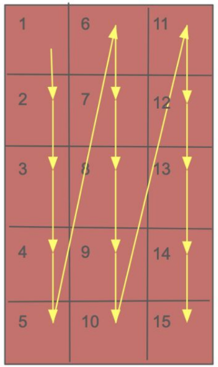
- Develop an approximation $\hat{H}_t$ to second-moment matrix H_t:
    - Accurate approximation.
    - Low memory to store.
    - Fast inversion.

Shampoo (Gupta et al., 2018)

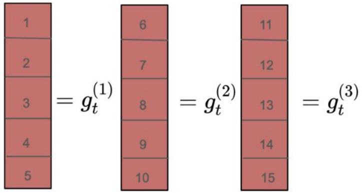
- Scalable implementation (Anil et al., 2020)
- Applications in recommendation models in Google (Anil et al., 2022)
- Practical kronecker product approximation.
- We utilize Kronecker sum to develop a better approximation.

Approximating Second-Moment Matrix of 2-D Parameter

$G_t =$

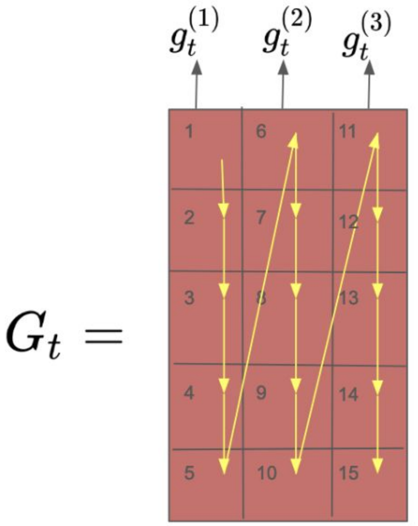


Column Major Flattening



Each column as separate gradient vector

Block-Diagonal Approximation with Identical Blocks



Column Major Flattening

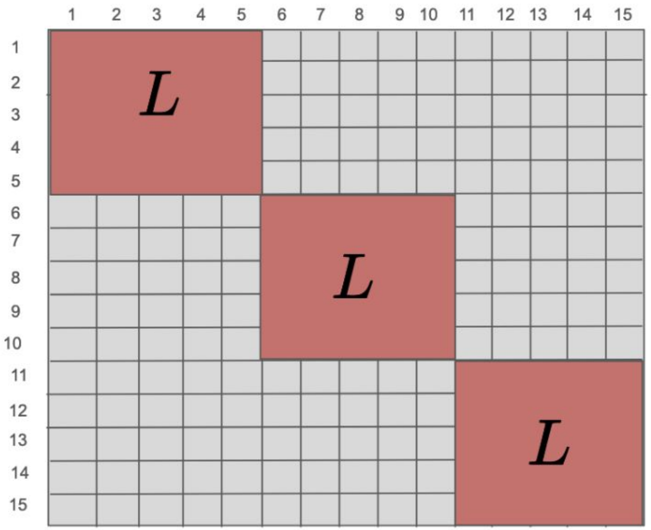


Figure: $I_n \otimes L$

Block-Diagonal Approximation with Identical Blocks

- Set all blocks to be L in the subproblem

$$L^* = \underset{L \succeq 0}{\operatorname{argmin}} \left\| I_n \otimes L - \sum_{t=1}^{T} g_t g_t^\top \right\|_F = \frac{1}{n} \sum_{t=1}^{T} G_t G_t^\top \qquad \text{(Explicit Solution!)}$$

Row Preconditioner

- We can similarly form row-preconditioner.

$$R^* = \operatorname*{argmin}_{L \succeq 0} \left\| R \otimes I_m - \sum_{t=1}^{T} g_t g_t^\top \right\|_F = \frac{1}{m} \sum_{t=1}^{T} G_t^\top G_t$$

Axes Preconditioners

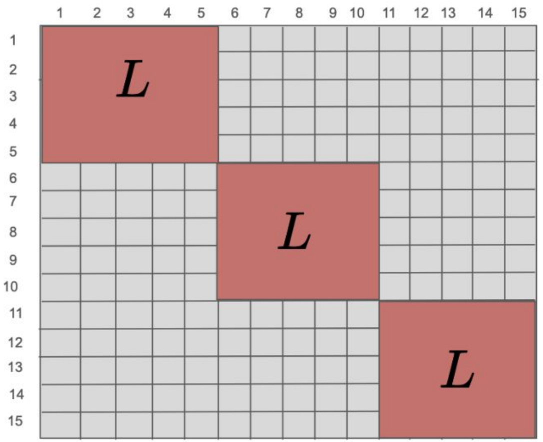


Figure: $I_n \otimes L$

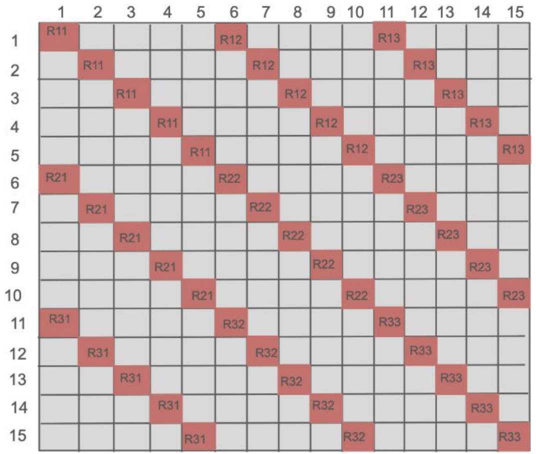


Figure: $R \otimes I_m$

- Individually both approximations miss out on a lot of cross-moments.

Axes Preconditioners

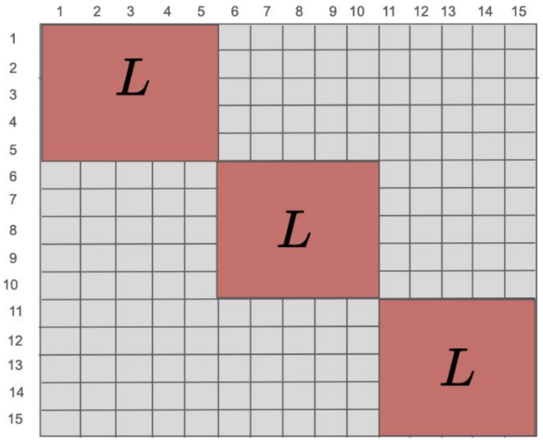


Figure: $I_n \otimes L$

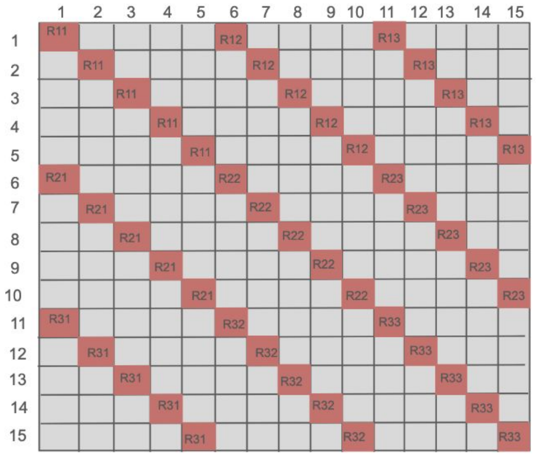


Figure: $R \otimes I_m$

- Individually both approximations miss out on a lot of cross-moments.
- Should combine both to approximate the remaining cross-moments?

CASPR Update

- CASPR update for p = 2 is:

$$X_t := \left((R_t^{-1/4} \otimes I_m + I_n \otimes L_t^{-1/4})/2 \right)^2; \quad W_t := W_{t-1} - \eta X_t g_t,$$

- Expanding the update gives:

$$W_{t+1} := W_t - \eta \left(L_t^{-1/2} G_t + 2 L_t^{-1/4} G_t R_t^{-1/4} + G_t R_t^{-1/2} \right)$$

Comparison with Shampoo Update

<table>
<tr><th>CASPR update</th><th>Shampoo update</th></tr>
<tr><td colspan="2">Update preconditioners:
$L_t := L_{t-1} + G_t G_t^\top, \quad R_t := R_{t-1} + G_t^\top G_t$</td></tr>
<tr><td colspan="2">Compute $L^{-1/4},\ R^{-1/4}$</td></tr>
<tr><td colspan="2">Precondition gradient and update parameters:</td></tr>
<tr><td>$U_t := L_t^{-1/4} G_t + G_t R_t^{-1/4}$
$U_t := L_t^{-1/4} U_t + U_t R_t^{-1/4}$
$W_t := W_{t-1} - \eta U_t$</td><td>$U_t := L_t^{-1/4} G_t R_t^{-1/4}$

$W_t := W_{t-1} - \eta U_t$</td></tr>
</table>

Regret Bound Analysis

- We conduct analysis in online convex optimization framework.

Theorem (Regret upper bound of CASPR Algorithm)

Given that the loss functions $f_t : \mathbb{R}^{m\times n} \to \mathbb{R}, \forall t \in [T]$ *are convex and* G*-Lipschitz in* ℓ_2*-norm i.e.,* $\|\nabla f_t(W)\|_2 \leq G,\ W \in \mathbb{R}^{m\times n}$ *, Algorithm 1 incurs the following regret*

$$\sum_{t=1}^{T} f_t(W_t) - f_t(W^*) \leq \sqrt{2r}D\,\mathrm{tr}\left(\left((L_T^{-1/4} \otimes I_n + I_m \otimes R_T^{-1/4})/2\right)^{-2}\right)$$

$$\leq \sqrt{2r}D\,\mathrm{tr}\left(L_T^{1/4} \otimes R_T^{1/4}\right) = \mathcal{O}(\sqrt{T})$$

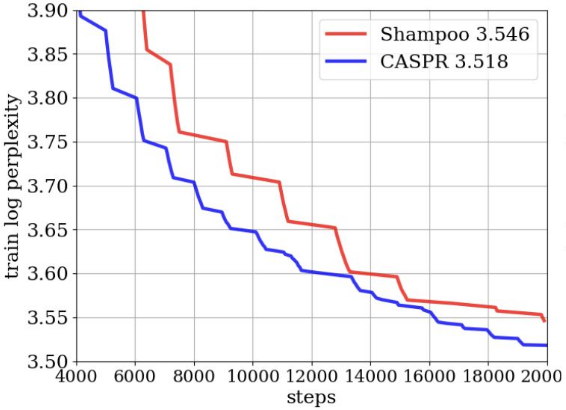
when $\eta = D/\sqrt{2r}$*, where* $r = \max_t \mathrm{rank}(G_t)$*,* $D = \|W_t - W^*\|_F$

- CASPR has tighter regret upper bound than Shampoo.

Autoregressive Large Language Modeling

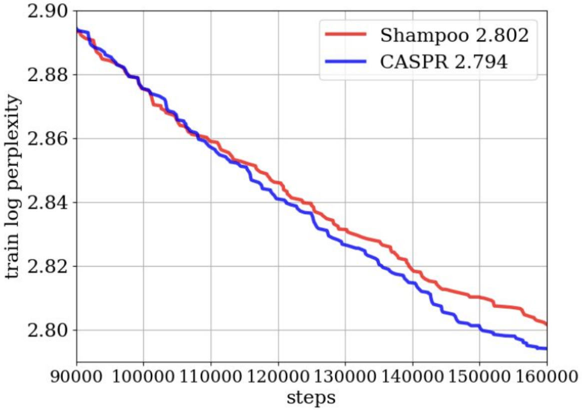
- GLU based decoder-only transformer models trained on C4 dataset.

160 billion tokens



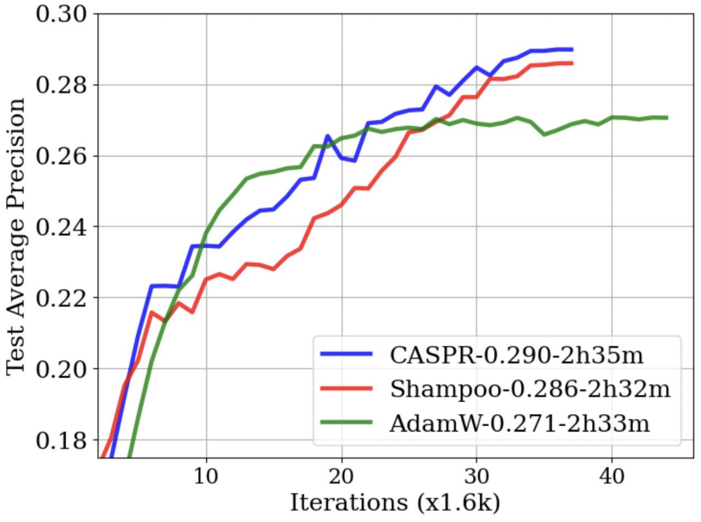
a) 8192 batch size and 14M parameters

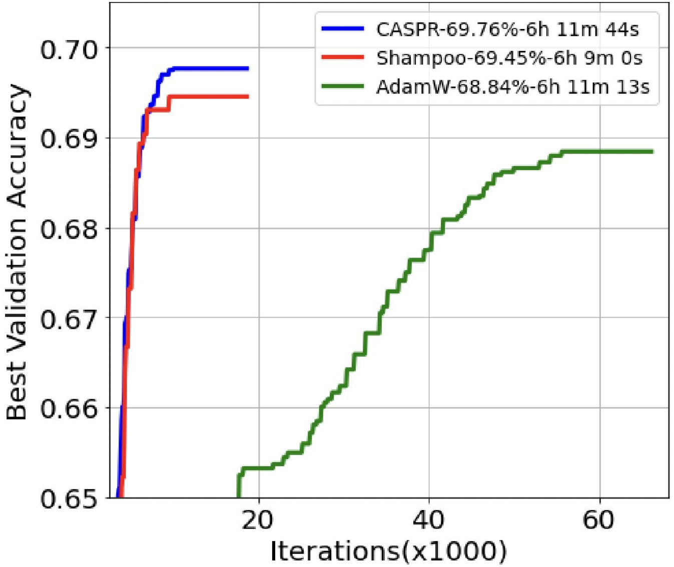
42 billion tokens



b) 256 batch size and 234M parameters.

GNN and Transformer on Parts of Speech





- Time taken for Shampoo and CASPR are about the same.
- CASPR has a better Validation Accuracy than Shampoo.
- CASPR is better than AdamW when run for fixed amount of time.

Conclusion and Future Directions

- Novel Kronecker-sum inspired combination approach to approximate the second-moment matrix using axes preconditioners.
- Stronger convergence guarantees than Shampoo, which is a special case of our framework of combining axes preconditioners.
- More accurate axes preconditioners solving the problem

$$\hat{H}_t := \arg\min_{\hat{H} \in \mathcal{S}} \left\| \hat{H} - H_t \right\|_F$$

- Adapt CASPR to approximate Hessian instead of full-matrix Adagrad.