# Memory-Efficient Backpropagation through Large Linear Layers

Daniel Bershatsky, Julia Gusak, Aleksandr Mikhalev, Daniil Merkulov, Alexandr Katrutsa, and Ivan Oseledets

[1]AIRI, Moscow, Russia [2]INRIA Center, University of Bordeaux, France [3]Skoltech AI Center, Moscow, Russia

ICOMP INTERNATIONAL CONFERENCE ON COMPUTATIONAL OPTIMIZATION

Skoltech AI Center for Artificial Intelligence Technology

## Summary

**Q:** Large models have many linear layers that require significant amount of memory to store inputs for gradient calculation in training. Can we reduce it?

**A:** Yes, with the power of randomized matmul (RMM)!

## Method

Linear layer acts on input batch $X \in \mathbb{R}^{B \times d_{\text{in}}}$ and output gradients $Y \in \mathbb{R}^{B \times d_{\text{out}}}$ as

$$\begin{aligned} X &\to XW^\top + \mathbb{1}_B b^\top, \\ \nabla_X \mathcal{L} &\text{ and } \nabla_W \mathcal{L} \leftarrow Y. \end{aligned} \quad (1)$$

Leveraging approximate matmul techniques, RMM divides gradient estimation $\nabla_W \mathcal{L} = Y^\top X$ in two steps

$$\begin{aligned} X_{\text{proj}} &= S^\top X, && \text{(precompute)} \\ \nabla_W \mathcal{L} &= Y^\top S X_{\text{proj}}, && \text{(calculate)} \end{aligned} \quad (2)$$

where $S \in \mathbb{R}^{B_{\text{proj}} \times B}$, $B_{\text{proj}} = \kappa B < B$ and $\mathbb{E}\, SS^\top = I_{B \times B}$. It can be deterministic (e.g. DCT or DFT) or random (e.g. Gaussian or Rademacher distributions), i.e. $S$ is easy to reconstruct in backward pass.

## Error Analysis

Lemma 1, 2 and Theorem 1 give the criterion of applicability. Variance of gradient estimate $D^2_{\text{SGD}}$ and perturbation $D_{\text{RMM}}$ should be of the same scale.

**Lemma 1** *(Aposteriori variance of SGD)* Let $X \in \mathbb{R}^{B \times d_{\text{in}}}$ and $Y \in \mathbb{R}^{B \times d_{\text{out}}}$ be the input to the linear layer in the forward pass and the input to it in the backward pass ($B$ here is the batch size). Then, we can estimate the variance of the noise induced by a random selection of the samples as

$$D^2_{\text{SGD}}(X, Y) = \frac{B}{B-1} \sum_{k=1}^{B} \|x_k\|^2 \|y_k\|^2 - \frac{\left\|X^\top Y\right\|_F^2}{B-1}, \quad (3)$$

where $x_k = X^\top e_k,$, $y_k = Y^\top e_k,$, $k = 1, \dots, B$, i.e., $x_k$ and $y_k$ are the columns of $X^\top$ and $Y^\top$, respectively.

**Lemma 2** *(Apriori variance of RMM)* Let $X \in \mathbb{R}^{B \times d_{\text{in}}}$ and $Y \in \mathbb{R}^{B \times d_{\text{out}}}$, then the variance of a randomized matrix multiplication through a matrix $S \in \mathbb{R}^{B \times B_{\text{proj}}}$ with i.i.d. elements following the normal distribution $\mathcal{N}\left(0, B_{\text{proj}}^{-1/2}\right)$ defined as

$$D^2(X, Y) = \mathbb{E}_S \left\|X^\top S S^\top Y - X^\top Y\right\|_F^2 \quad (4)$$

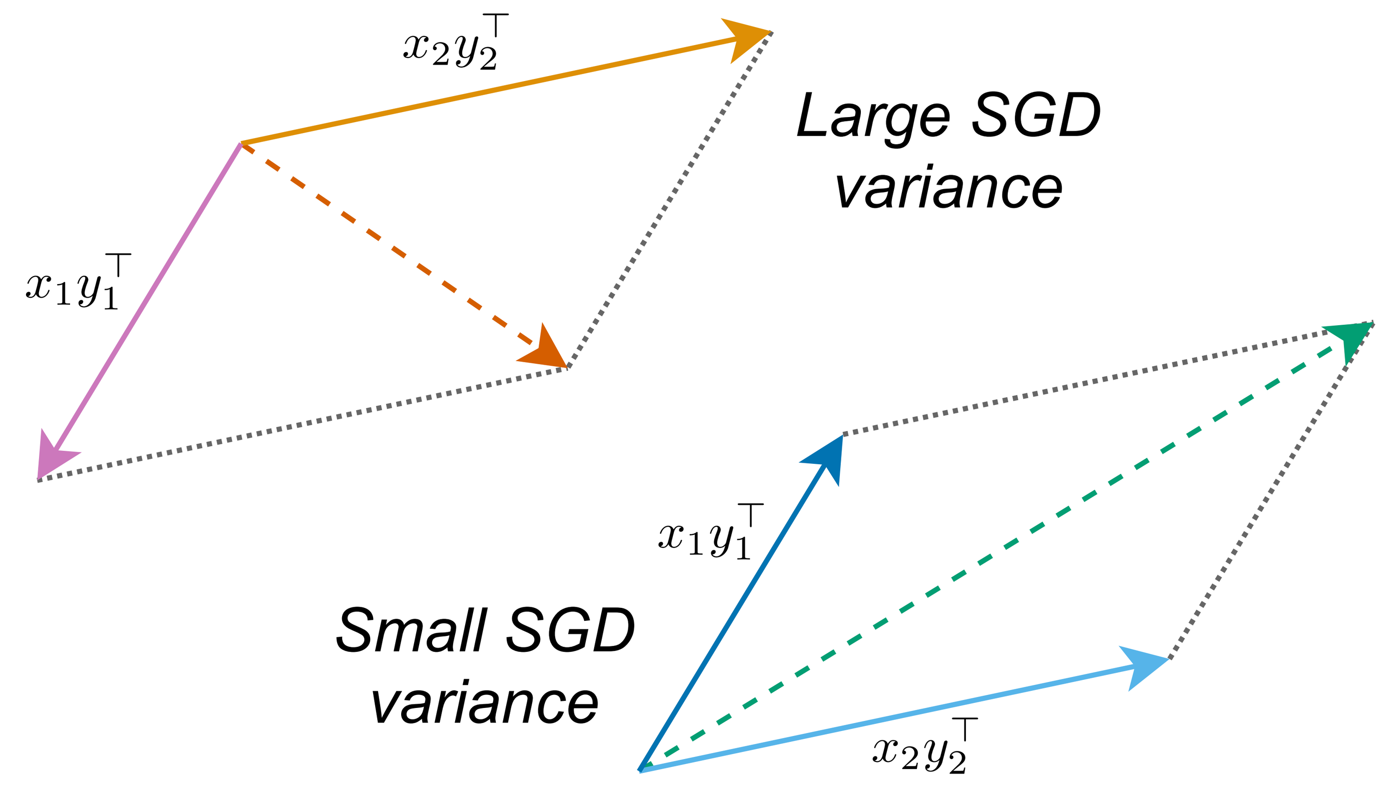


Figure 1: Visualization support for Lemma 1.

can be evaluated as follows

$$D^2_{\text{RMM}}(X, Y) = \frac{\|X\|_F^2 \|Y\|_F^2 - \left\|X^\top Y\right\|_F^2}{B_{\text{proj}}}. \quad (5)$$

**Theorem 1** *(Upper bound of variance)* In the conditions of Lemma 1 and Lemma 2 the in-sample variance $D_{\text{SGD}}$ and the variance $D_{\text{RMM}}$ induced by a randomized subsampling are tied with the following inequality

$$\frac{B_{\text{proj}}}{B-1} \frac{D^2_{\text{RMM}(X,Y)}}{D^2_{\text{SGD}(X,Y)}} \le \frac{\alpha + 1}{\alpha}, \quad (6)$$

where $\alpha = \left\|X^\top Y\right\|_F^2 / \left(\|X\|_F^2 \|Y\|_F^2\right)$, $\alpha \in [0, 1]$.

## Experiments

Most of the experiments are carried out with RoBERTa$_{\text{base}}$ on GLUE benchmark. Table 1 demonstrates how model performance changes with compression rate $\kappa = B / B_{\text{proj}}$. Figure 2 confirms empirically the statement of Theorem 1. Table 2 presents ablation study experiment on choice of matmul. Memory savings measurements are shown in Table 3.

Table 1: Fine-tuning on GLUE benchmark for different compression rates $\kappa$.

| Rate, $\kappa$ | COLA | MNLI | MRPC | QNLI | QQP | RTE | SST2 | STSB | Σ |
|---|---|---|---|---|---|---|---|---|---|
| — | 60.51 | 87.56 | 89.30 | 92.60 | 91.69 | 78.52 | 94.09 | 90.37 | 85.58 |
| 1.1 | 59.75 | 87.58 | 88.64 | 92.75 | 91.47 | 77.50 | 94.72 | 90.39 | 85.35 |
| 2 | 59.45 | 87.58 | 88.73 | 92.56 | 91.41 | 77.18 | 94.61 | 90.32 | 85.23 |
| 5 | 57.46 | 87.59 | 87.99 | 92.62 | 91.16 | 76.26 | 94.43 | 90.06 | 84.70 |
| 10 | 57.53 | 87.51 | 88.30 | 92.55 | 90.93 | 75.45 | 94.27 | 89.90 | 84.56 |

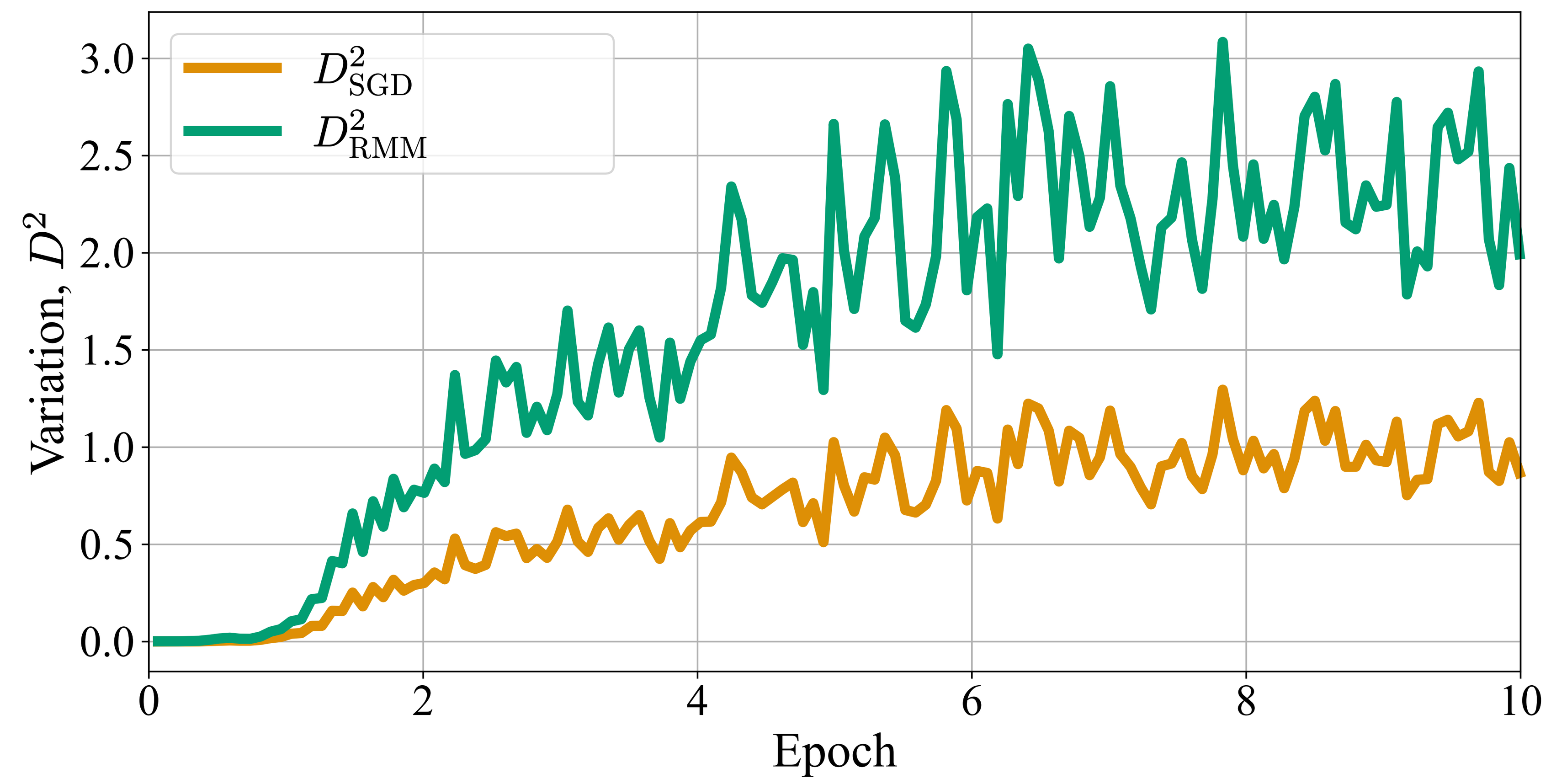


Figure 2: Evolution of estimate variances in training time.

Table 2: Comparison of different randomized matmul variants.

| MatMul | Rate, $\kappa$ | Score | Time |
|---|---|---|---|
| No RMM | — | 60.90 | 08:44 |
| | 2 | 59.17 | 16:26 |
| | 5 | 58.81 | 16:37 |
| DCT | 10 | 53.38 | 17:24 |
| | 2 | 59.05 | 12:20 |
| | 5 | 60.60 | 11:42 |
| DFT | 10 | 47.62 | 12:25 |
| | 2 | 58.60 | 10:36 |
| | 5 | 57.79 | 10:02 |
| Gauss | 10 | 56.52 | 10:03 |
| | 2 | 62.38 | 15:27 |
| | 5 | 59.11 | 15:38 |
| Radem. | 10 | 55.50 | 15:43 |

Table 3: Memory usage during training on GLUE.

| Task | Batch | Rate, $\kappa$ | Mem, GiB | Save, % |
|---|---|---|---|---|
| MRPC | 128 | 1 | 11.3 | 0.0 |
| | | 2 | 10.6 | 6.3 |
| | | 5 | 9.2 | 19.3 |
| | | 10 | 8.7 | 23.3 |
| QNLI | 16 | 1 | 11.7 | 0.0 |
| | | 2 | 11.2 | 4.2 |
| | | 5 | 10.4 | 11.6 |
| | | 10 | 10.1 | 13.8 |
| SST2 | 256 | 1 | 13.3 | 0.0 |
| | | 2 | 12.5 | 6.1 |
| | | 5 | 10.5 | 20.8 |
| | | 10 | 9.9 | 25.5 |