Rank-revealing QR factorizations: applications, algorithms, and theory

Anil Damle

Abstract

Rank-revealing factorizations, e.g., [2, 7], have a long history in numerical linear algebra. We continue this story in multiple directions by discussing recent highlights of their development and use. This starts with a discussion about how pivoted QR factorizations play a central role in techniques for compressing modern, large-scale deep learning models [3, 5]. Motivated by that work we briefly highlight recent advances in computational methods for computing interpolative decompositions that leverage tools from randomized numerical linear algebra [1] and discuss associated theoretical developments that more clearly capture the behavior of low-rank matrix approximations derived from pivoted factorizations

Modern deep learning models are often vastly overparametrized for their desired task; it is difficult to determine a optimal model size based on a description of the problem and/or training data. However, this has consequence as it leads to large models that are expensive to store and run inference on. We show that given a small amount of (potentially unlabeled) data we can compress a given model into one of smaller size that retains the same structure as the original model—it is just smaller. To illustrate this process we can consider a one-hidden layer neural network

$$f(x) = \sigma(x^T W)\alpha,$$

where $x \in \mathbb{R}^d$ represents a data point, $W \in \mathbb{R}^{d\times n}$ is the weight matrix, $\alpha \in \mathbb{R}^n$ is a linear last layer, and σ is a non-linear function applied entrywise. Our task is to compute $\widehat{W}^{d\times m}$ and $\widehat{\alpha}^m$ with $m < n$ such that $f(x) \approx \sigma(x^T\widehat{W})\widehat{\alpha}$ to the desired accuracy and for all sensible x.

Given some small amount of data points, which we encode as the columns of X_C, we accomplish this goal by computing an interpolative decomposition [4] of $Z = \sigma(X_C^T W)\alpha$ as

$$Z \approx Z(:\mathcal{C})T,$$

where $\mathcal{C}$ represents a subset of the columns of Z. Because the non-linear function is applied entrywise it commutes with subset selection and we have that

$$f(x) \approx \sigma(x^T W(:,\mathcal{C}))(T\alpha).$$

Letting $\widehat{W} = W(:,\mathcal{C})$ and $\widehat{\alpha} = T\alpha$ accomplishes our goal. This idea can be extended to multiple layers and more complicated layer types.

In the preceding use case, the matrices that we have to compute interpolative decompositions of can be quite large. However, the final quality of the process is not typically dependent on the exact subset of columns chosen—we just need a sufficiently good subset. This motivates the use of randomized algorithms to rapidly compute a suitable $\mathcal{C}$. Numerous algorithms exist for this task, and we provide a novel randomized version of the Golub-Klema-Stewart subset selection algorithm [6] that performs admirably in practice. In particular, we observe that its performance (and that of alternatives) depends on properties of singular vectors and we derive theoretical bounds that highlight this fact [1].

References

[1] R. Armstrong, A. Buzali, and A. Damle, *Structure-aware analyses and algorithms for interpolative decompositions*, arXiv preprint arXiv:2310.09452, (2023).

[2] S. Chandrasekaran and I. C. Ipsen, *On rank-revealing factorisations*, SIAM Journal on Matrix Analysis and Applications, 15 (1994), pp. 592–622.

[3] J. Chee, M. Renz, A. Damle, and C. D. Sa, *Model preserving compression for neural networks*, in Advances in Neural Information Processing Systems, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, eds., 2022.

[4] H. Cheng, Z. Gimbutas, P.-G. Martinsson, and V. Rokhlin, *On the compression of low rank matrices*, SIAM Journal on Scientific Computing, 26 (2005), pp. 1389–1404.

[5] M. Flynn, A. Wang, D. E. Alvarez, C. De Sa, and A. Damle, *STAT: Shrinking transformers after training*, arXiv preprint arXiv:2406.00061, (2024).

[6] G. Golub, V. Klema, and G.W. Stewart, *Rank degeneracy and least squares problems*, Stanford University department of Computer Science, (1976).

[7] M. Gu and S. C. Eisenstat, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM Journal on Scientific Computing, 17 (1996), pp. 848–869.