Why Approximate Matrix Square Root Outperforms Accurate SVD in Global Covariance Pooling?

Yue Song [1] **Nicu Sebe** [1] **Wei Wang** [1]

[1]DISI, University of Trento, Italy

An Intriguing Phenomenon About the Performance Gap

Global Covariance Pooling (GCP) methods compute the sample covariance of the final convolutional feature and use it as the global representation. During the past years, the GCP methods have achieved state-of-the-art performances on both generic and fine-grained classification.

Existing methods are mainly base on MPN-COV [2] that uses the **Singular Value Decomposition (SVD)** to compute the accurate solution and the iSQRT-COV [1] which uses the **Newton-Schulz iteration (NS iteration)** to derive the approximate solution. One interesting phenomenon is that the approximate method can even outperform the accurate one by a large margin (see Table 1). Our paper intends to investigate the underlying reason behind the performance gap.

Table: Validation error of ResNet-50 on ImageNet.

Method	Spectral Layer	Top-1 Err.	Top-5 Err.
MPN-COV [2]	SVD	22.73	6.54
iSQRT-COV [1]	NS iteration	**22.14**	**6.22**

Gradient Smoothness Analysis

The back-propagation algorithm of SVD involves the matrix K whose off-diagonal entry $\mathrm{K_{ij}} = \frac{1}{\lambda_i - \lambda_j}$. When the two eigenvalues are small and close, it is very likely to trigger the gradient explosion issue, *i.e.,* $\mathrm{K_{ij}} \to \infty$. We conjecture this issue might be related with the performance disparity.

Fig. 1 displays the effective β-smoothness of these two methods. The lower values indicate smoother gradients. As can be seen, the MPN-COV has far less smooth gradients than the iSQRT-COV. This phenomenon naturally arises the research question: *Can we smooth the backward gradient of the SVD to boost the performance of MPN-COV?*

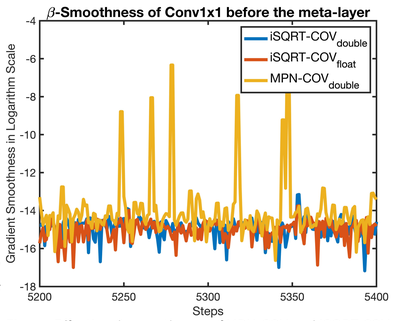

Figure: Effective β-smoothness of MPN-COV and iSQRT-COV.

Differentiable SVD Variants With Smooth Gradients

To validate the impact of gradient smoothness, we try the following SVD varaints by manipulating either the forward or the backward algorithms:

- **SVD-TopN** that keeps only the large top-N eigenvalues.
- **SVD-Trunc** where an upper limit is set to clip the gradients.
- **SVD-PI** which uses the Power Iteration (PI) to approximate the backward gradients.
- **SVD-Newton** that uses the gradient of the NS iteration instead
- **SVD-Taylor** which relies on the Taylor polynomial to decompose the gradient function as $\mathrm{K_{ij}} \approx \frac{1}{\lambda_i}(1 + \frac{\lambda_j}{\lambda_i} + (\frac{\lambda_j}{\lambda_i})^2 + \cdots + (\frac{\lambda_j}{\lambda_i})^K) \leq \frac{K+1}{\lambda_i}$.

All the SVD variants have smoothed the gradient to different extents and improved the performance of MPN-COV. However, they are still not comparable against the iSQRT-COV. **This observation implies the gradient smoothness does not fully account for the performance gap.**

Another interesting phenomenon is that as the learning rate decays, the covariance matrices are becoming better-conditioned (see Table 2). The SVD is not good at dealing with the ill-conditioned matrices in the first two stages because these matrices are not stable and close to singular.

Table: Condition number of different epochs. Higher value indicates the matrix is less stable and closer to singular.

Methods	Epoch 1	Epoch 11	Epoch 21
MPN-COV [2]	3.41 e14	5.36 e13	1.70 e13

Hybrid Training Protocol

Key Idea: Let the SVD process the well-conditioned matrices in the last stage.

We develop a hybrid training protocol: the model is trained using NS iteration until the last learning rate decays, then switch to SVD to tune the model throughout the last stage. This strategy can fully exploits the potential of SVD for stable eigendecomposition.

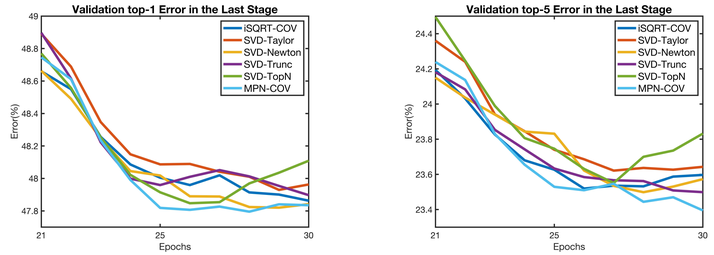

Figure: The validation top-1 (left) and top-5 error (right) of AlexNet in the last stage using the hybrid training strategy.

Fig. 2 depicts the performance in the last stage when using the hybrid training strategy. Unlike the case of standalone training, the SVD variants with smooth gradients do not show obvious improvement over ordinary SVD. **This observation challenges the necessity and effectiveness to smooth the gradients.**

Problem with Taylor Polynomial

The SVD-Taylor that uses the Taylor polynomial to approximate the gradients is a promising direction but suffers from the convergence radii. Its gradient is calculated as:

$$\mathrm{K_{ij}} = \frac{1}{\lambda_i - \lambda_j} = \frac{1}{\lambda_i} \cdot \frac{1}{1 - (\lambda_j/\lambda_i)} \approx \frac{1}{\lambda_i}(1 + \frac{\lambda_j}{\lambda_i} + (\frac{\lambda_j}{\lambda_i})^2 + \cdots + (\frac{\lambda_j}{\lambda_i})^K)$$

The Taylor series defined here is a meromorphic function with a pole at 1. When the eigenvalue ratio $\frac{\lambda_j}{\lambda_i}$ is close to the convergence boundary, the gradient can be poorly estimated.

Solution: rational approximation to avoid this drawback.

SVD-Padé: Padé Approximants for Close Gradient Approximation

We propose to use Padé approximants, a particular kind of rational approximation, to closely estimate the backward gradients.

Formulation in SVD back-propagation. We first compute the Padé coefficients $\mathrm{P_M(x)}$ and $\mathrm{Q_N(x)}$ by matching to the power series. Afterwards, we re-express the gradient function as:

$$\mathrm{K_{ij}} \approx \frac{1}{\lambda_i} \cdot \frac{\mathrm{P_M}(\lambda_j/\lambda_i)}{\mathrm{Q_N}(\lambda_j/\lambda_i)} = \frac{1}{\lambda_i} \cdot \frac{\sum_{m=0}^{M} p_m (\frac{\lambda_j}{\lambda_i})^m}{1 + \sum_{n=1}^{N} q_n (\frac{\lambda_j}{\lambda_i})^n}$$

Enlarged Convergence Range. As seen from Table 3, our SVD-Padé has the enlarged convergence range and consistently give better approximation for any possible eigenvalue ratio.

Table: Approximation error of Taylor polynomial and diagonal Padé approximants of degree 100 in double precision.

λ_j/λ_i	0.1	0.3	0.5	0.7	0.9	0.99	0.999
SVD-Taylor	9e-19	7e-18	2e-21	8e-16	2e-4	36	904
SVD-Padé	9e-19	5e-18	1e-21	5e-17	3e-16	8e-13	3e-10

Results on ImageNet

Table: Validation errors of ResNet-50 and ResNet-101. The best three results are highlighted in red, blue, and green.

	Methods	Standalone Training		Hybrid Training	
		top-1	top-5	top-1	top-5
ResNet-50	iSQRT-COV	22.81	6.60	22.81	6.60
	SVD-Padé	22.67	6.51	22.60	6.44
	SVD-Taylor	22.91	6.67	22.77	6.53
	SVD-Newton	22.86	6.65	22.72	6.55
	SVD-Trunc	22.85	6.70	22.74	6.52
	SVD-TopN	22.91	6.68	22.76	6.51
	MPN-COV	22.93	6.75	22.79	6.50
	Vanilla ResNet-50	23.85	7.13	23.85	7.13
ResNet-101	iSQRT-COV	21.60	5.88	21.60	5.88
	SVD-Padé	21.48	5.80	21.40	5.69
	MPN-COV	21.79	5.99	21.58	5.80
	Vanilla ResNet-101	22.63	6.44	22.63	6.44

Table 4 displays the performance on ImageNet. Our SVD-Padé achieves state-of-the-art performances no matter which training strategy is used.

Results on Fine-grained Benchmarks

Table: Comparison of accuracy (%) on fine-grained datasets. The best three results are highlighted in red, blue, and green respectively. / means the method cannot converge.

Methods	Birds		Dogs		Cars	
	Final	Best	Final	Best	Final	Best
iSQRT-COV	85.95	86.45	82.34	83.45	90.93	91.56
SVD-Padé	87.05	87.29	83.40	84.34	92.55	92.99
SVD-Taylor	86.95	87.20	83.23	84.22	92.46	92.71
SVD-Newton	86.97	87.22	83.08	83.94	92.35	92.51
SVD-Trunc	87.16	87.25	82.95	84.03	92.43	93.04
SVD-TopN	/	/	/	/	/	/
MPN-COV	86.89	87.19	83.34	84.24	92.45	92.84

On fine-grained benchmarks, our SVD-Padé also has the best evaluation results on most metrics. Furthermore, the SVD-based methods significantly outperform iSQRT-COV by at least $1\,\%$.

Upper Bound of Gradient

Table: Upper bound of the gradient $\mathrm{K_{ij}}$ for each SVD method.

Methods	SVD-Padé	SVD-Taylor	SVD-Trunc	SVD-TopN	SVD-Newton	SVD
Analytical Form	$\frac{1}{\lambda_i} \cdot \frac{\sum_{m=0}^{M} p_m}{1 + \sum_{n=1}^{N} q_n}$	$\frac{K+1}{\lambda_i}$	T	$\frac{1}{\lambda_N}$	/	$\frac{1}{\lambda_i - \lambda_j}$
Maximal Value	6.00e36	4.55e17	1e10	4.50e15	/	∞
Trigger Condition	$\lambda_i = \lambda_j \leq$ EPS	$\lambda_i = \lambda_j \leq$ EPS	$\frac{1}{\lambda_i - \lambda_j} \geq$ T	$\lambda_N \leq$ EPS	/	$\lambda_i = \lambda_j$

Table. 6 summarizes the upper bound of gradient. Our proposed SVD-Padé allows for the largest upper bound, but the maximal value is still acceptable in both double precision and float precision.

References

[1] Peihua Li, Jiangtao Xie, Qilong Wang, and Zilin Gao.
Towards faster training of global covariance pooling networks by iterative matrix square root normalization.
In *CVPR*, 2018.

[2] Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo.
Is second-order information helpful for large-scale visual recognition?
In *ICCV*, 2017.