Improving Covariance Conditioning of the SVD Meta-layer by Orthogonality –Supplementary Material–

Yue Song[0000−0003−1573−5643], Nicu Sebe, and Wei Wang

DISI, University of Trento, Trento 38123, Italy
yue.song@unitn.it
https://github.com/KingJamesSong/OrthoImproveCond

A Mathematical Derivation and Proof

A.1 Derivation of Nearest Orthogonal Gradient

The problem of finding the nearest orthogonal gradient can be defined as:

$$\min_{\mathbf{R}} ||\frac{\partial l}{\partial \mathbf{W}} - \mathbf{R}||_{\mathrm{F}} \ \textit{subject to}\ \mathbf{R}\mathbf{R}^T = \mathbf{I} \tag{1}$$

To solve this constrained optimization problem, We can construct the following error function:

$$e(\mathbf{R}) = Tr\Big((\frac{\partial l}{\partial \mathbf{W}} - \mathbf{R})^T(\frac{\partial l}{\partial \mathbf{W}} - \mathbf{R})\Big) + Tr\Big(\mathbf{\Sigma}\mathbf{R}^T\mathbf{R} - \mathbf{I}\Big) \tag{2}$$

where $Tr(\cdot)$ is the trace measure, and $\mathbf{\Sigma}$ denotes the symmetric matrix Lagrange multiplier. Setting the derivative to zero leads to:

$$\begin{aligned} \frac{de(\mathbf{R})}{d\mathbf{R}} &= -2(\frac{\partial l}{\partial \mathbf{W}} - \mathbf{R}) + 2\mathbf{R}\mathbf{\Sigma} = 0 \\ \frac{\partial l}{\partial \mathbf{W}} &= \mathbf{R}(\mathbf{I} + \mathbf{\Sigma}),\ \mathbf{R} = \frac{\partial l}{\partial \mathbf{W}}(\mathbf{I} + \mathbf{\Sigma})^{-1} \end{aligned} \tag{3}$$

The term $(\mathbf{I}+\mathbf{\Sigma})$ can be represented using $\frac{\partial l}{\partial \mathbf{W}}$. Consider the covariance of $\frac{\partial l}{\partial \mathbf{W}}$:

$$\begin{aligned} (\frac{\partial l}{\partial \mathbf{W}})^T\frac{\partial l}{\partial \mathbf{W}} &= (\mathbf{I} + \mathbf{\Sigma})^T\mathbf{R}^T\mathbf{R}(\mathbf{I} + \mathbf{\Sigma}) = (\mathbf{I} + \mathbf{\Sigma})^T(\mathbf{I} + \mathbf{\Sigma}) \\ (\mathbf{I} + \mathbf{\Sigma}) &= \Big((\frac{\partial l}{\partial \mathbf{W}})^T\frac{\partial l}{\partial \mathbf{W}}\Big)^{\frac{1}{2}} \end{aligned} \tag{4}$$

Substituting the term $(\mathbf{I} + \mathbf{\Sigma})$ in eq. (3) with the above equation leads to the closed-form solution of the nearest orthogonal gradient:

$$\mathbf{R} = \frac{\partial l}{\partial \mathbf{W}}\Big((\frac{\partial l}{\partial \mathbf{W}})^T\frac{\partial l}{\partial \mathbf{W}}\Big)^{-\frac{1}{2}} \tag{5}$$

A.2 Derivation of Optimal Learning Rate

To jointly optimize the updated weight $\mathbf{W}-\eta\frac{\partial l}{\partial \mathbf{W}}$, we need to achieve the following objective:

$$\min_{\eta} ||(\mathbf{W}-\eta\frac{\partial l}{\partial \mathbf{W}})(\mathbf{W}-\eta\frac{\partial l}{\partial \mathbf{W}})^T - \mathbf{I}||_{\mathrm{F}} \tag{6}$$

This optimization problem can be more easily solved in the form of vector. Let $\mathbf{w}$, $\mathbf{i}$, and $\mathbf{l}$ denote the vectorized $\mathbf{W}$, $\mathbf{I}$, and $\frac{\partial l}{\partial \mathbf{W}}$, respectively. Then we construct the error function as:

$$e(\eta) = \Big((\mathbf{w}-\eta\mathbf{l})^T(\mathbf{w}-\eta\mathbf{l}) - \mathbf{i}\Big)^T \Big((\mathbf{w}-\eta\mathbf{l})^T(\mathbf{w}-\eta\mathbf{l}) - \mathbf{i}\Big) \tag{7}$$

Expanding the equation leads to:

$$e(\eta) = (\mathbf{w}^T\mathbf{w} - 2\eta\mathbf{l}^T\mathbf{w} + \eta^2\mathbf{l}^T\mathbf{l} - \mathbf{i})^T(\mathbf{w}^T\mathbf{w} - 2\eta\mathbf{l}^T\mathbf{w} + \eta^2\mathbf{l}^T\mathbf{l} - \mathbf{i}) \tag{8}$$

Differentiating $e(\eta)$ w.r.t. η yields:

$$\begin{aligned}\frac{de(\eta)}{d\eta} = -4\mathbf{w}\mathbf{w}^T\mathbf{l}^T\mathbf{w} + 4\eta\mathbf{w}\mathbf{w}^T\mathbf{l}^T\mathbf{l} + 8\eta\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w} - 12\eta^2\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{l} + 4\mathbf{l}\mathbf{w}^T\mathbf{i} \\ +4\eta^3\mathbf{l}\mathbf{l}^T - 4\eta\mathbf{i}\mathbf{l}\mathbf{l}^T\end{aligned} \tag{9}$$

Since η is typically very small, the higher-order terms (*e.g.,* η^2 and η^3) are sufficiently small such that they can be neglected. After omitting these terms, the derivative becomes:

$$\frac{de(\eta)}{d\eta} \approx -4\mathbf{w}\mathbf{w}^T\mathbf{l}^T\mathbf{w} + 4\eta\mathbf{w}\mathbf{w}^T\mathbf{l}^T\mathbf{l} + 8\eta\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w} + 4\mathbf{l}\mathbf{w}^T\mathbf{i} - 4\eta\mathbf{i}\mathbf{l}\mathbf{l}^T \tag{10}$$

Setting the derivative to zero leads to the optimal learning rate:

$$\eta^\star \approx \frac{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{w} - \mathbf{l}^T\mathbf{w}\mathbf{i}}{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{l} + 2\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w} - \mathbf{l}^T\mathbf{l}\mathbf{i}} \tag{11}$$

Notice that $\mathbf{i}$ is the vectorization of the identify matrix $\mathbf{I}$, which means that $\mathbf{i}$ is very sparse (*i.e.,* lots of zeros) and the impact can be neglected. The optimal learning rate can be further simplified as:

$$\eta^\star \approx \frac{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{w}}{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{l} + 2\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w}} \tag{12}$$

A.3 Proof of the learning rate bounds

Proposition 1 *When both* $\mathbf{W}$ *and* $\frac{\partial l}{\partial \mathbf{W}}$ *are orthogonal,* $\eta^\star$ *is both upper and lower bounded. The upper bound is* $\frac{N^2}{N^2+2}$ *and the lower bound is* $\frac{1}{N^2+2}$ *where* N *denotes the row dimension of* $\mathbf{W}$.

Proof. Since the vector product is equivalent to the matrix Frobenius inner product, we have the relation:

$$\mathbf{l}^T\mathbf{w} = \langle \frac{\partial l}{\partial \mathbf{W}}, \mathbf{W} \rangle_{\mathrm{F}} \tag{13}$$

For a given matrix pair $\mathbf{A}$ and $\mathbf{B}$, the Frobenius product $\langle \cdot \rangle_{\mathrm{F}}$ is defined as:

$$\langle \mathbf{A}, \mathbf{B} \rangle_{\mathrm{F}} = \sum A_{i,j} B_{i,j} \leq \sigma_1(\mathbf{A})\sigma_1(\mathbf{B}) + \cdots + \sigma_N(\mathbf{A})\sigma_N(\mathbf{B}) \tag{14}$$

where $\sigma(\cdot)_i$ represents the i-th largest eigenvalue, N denotes the matrix size, and the inequality is given by Von Neumann's trace inequality [6, 2]. The equality takes only when $\mathbf{A}$ and $\mathbf{B}$ have the same eigenvector. When both $\mathbf{W}$ and $\frac{\partial l}{\partial \mathbf{W}}$ are orthogonal, *i.e.,* their eigenvalues are all 1, we have the following relation:

$$\langle \frac{\partial l}{\partial \mathbf{W}}, \frac{\partial l}{\partial \mathbf{W}} \rangle_{\mathrm{F}} = N, \ \langle \frac{\partial l}{\partial \mathbf{W}}, \mathbf{W} \rangle_{\mathrm{F}} \leq N \tag{15}$$

This directly leads to:

$$\langle \frac{\partial l}{\partial \mathbf{W}}, \mathbf{W} \rangle_{\mathrm{F}} \leq \langle \frac{\partial l}{\partial \mathbf{W}}, \frac{\partial l}{\partial \mathbf{W}} \rangle_{\mathrm{F}}, \ \mathbf{l}^T\mathbf{w} \leq \mathbf{l}^T\mathbf{l} \tag{16}$$

Exploiting this inequality, the optimal learning rate has the relation:

$$\eta^\star \approx \frac{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{w}}{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{l} + 2\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w}} \leq \frac{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{l}}{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{l} + 2\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w}} \tag{17}$$

For $\mathbf{l}^T\mathbf{w}$, we have the inequality as:

$$\mathbf{l}^T\mathbf{w} = \langle \frac{\partial l}{\partial \mathbf{W}}, \mathbf{W} \rangle_{\mathrm{F}} = \sum_{i,j} \frac{\partial l}{\partial \mathbf{W}}_{i,j} \mathbf{W}_{i,j} \geq \sigma_{min}(\frac{\partial l}{\partial \mathbf{W}})\sigma_{min}(\mathbf{W}) = 1 \tag{18}$$

Then we have the upper bounded of $\eta^\star$ as:

$$\eta^\star \leq \frac{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{l}}{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{l} + 2\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w}} = \frac{N^2}{N^2 + 2\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w}} < \frac{N^2}{N^2 + 2} \tag{19}$$

For the lower bound, since we also have $\mathbf{l}^T\mathbf{w} \leq \mathbf{w}^T\mathbf{w}$, $\eta^\star$ can be re-written as:

$$\eta^\star \approx \frac{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{w}}{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{l} + 2\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w}} \geq \frac{\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w}}{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{l} + 2\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w}} = \frac{1}{\frac{\mathbf{w}^T\mathbf{w}\mathbf{l}^T\mathbf{l}}{\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w}} + 2} = \frac{1}{\frac{N^2}{\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w}} + 2} \tag{20}$$

Injecting eq. (18) into eq. (20) leads to the further simplification:

$$\eta^\star \approx \frac{1}{\frac{N^2}{\mathbf{l}^T\mathbf{w}\mathbf{l}^T\mathbf{w}} + 2} \geq \frac{1}{N^2 + 2} \tag{21}$$

As indicated above, the optimal learning rate $\eta^\star$ has a lower bound of $\frac{1}{N^2+2}$.

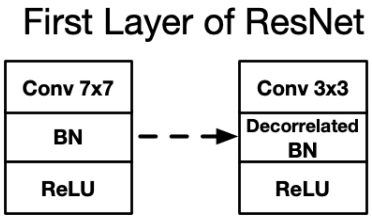

Fig. 1. The scheme of the modified ResNet for decorrelated BN. We reduce the kernel size of the first convolution layer from 7×7 to 3×3. The BN after this layer is replaced with our decorrelated BN layer.

B Detailed Experimental Settings

In this section, we introduce the implementation details and experimental settings of the two experiments.

B.1 Decorrelated Batch Normalization

We use ResNet-50 [3] as the backbone for the experiment on CIFAR10 and CIFAR100 [4]. The kernel size of the first convolution layer of ResNet is 7×7, which might not suit the low resolution of these two datasets (the images are only of size 32×32). To avoid this issue, we reduce the kernel size of the first convolution layer to 3×3. The stride is also decreased from 2 to 1. The BN layer after this layer is replace with our decorrelated BN layer (see Fig. 1). Let $\mathbf{X}\in\mathbb{R}^{C\times BHW}$ denotes the reshaped feature. The whitening transform is performed as:

$$\mathbf{X}_{whitened} = (\mathbf{X}\mathbf{X}^T)^{-\frac{1}{2}}\mathbf{X} \tag{22}$$

Compared with the vanilla BN that only standardizes the data, the decorrelated BN can further eliminate the data correlation between each dimension.

The training lasts 350 epochs and the learning rate is initialized with 0.1. The SGD optimizer is used with momentum 0.9 and weight decay $5e{-}4$. We decrease the learning rate by 10 every 100 epochs. The batch size is set to 128. We use the technique proposed in [7] to compute the stable SVD gradient. The Pre-SVD layer in this experiment is the 3×3 convolution layer.

B.2 Global Covariance Pooling

We use ResNet-18 [3] for the GCP experiment and train it from scratch on ImageNet [1]. Fig. 2 displays the overview of a GCP model. For the ResNet backbone, the last Global Average Pooling (GAP) layer is replaced with our GCP layer. Consider the final batched convolutional feature $\mathbf{X}\in\mathbb{R}^{B\times C\times HW}$. We compute the matrix square root of its covariance as:

$$\mathbf{Q} = (\mathbf{X}\mathbf{X}^T)^{\frac{1}{2}} \tag{23}$$

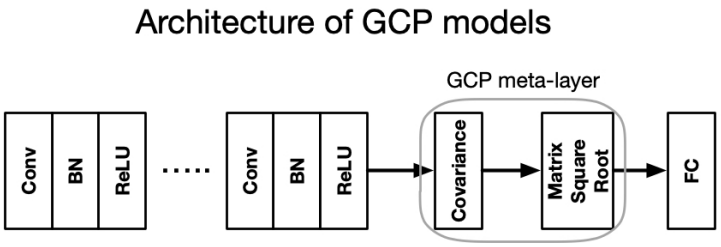

Fig. 2. The architecture of a GCP model [5, 7]. After all the convolution layers, the covariance square root of the feature is computed and used as the final representation.

where $\mathbf{Q} \in \mathbb{R}^{B \times C \times C}$ is used as the final representation and directly passed to the fully-connected (FC) layer.

The training process lasts 60 epochs and the learning rate is initialize with 0.1. We decrease the learning rate by 10 at epoch 30 and epoch 45. The SGD optimizer is used with momentum 0.9 and weight decay $1e{-}4$. The model weights are randomly initialized and the batch size is set to 256. The images are first resized to 256×256 and then randomly cropped to 224×224 before being passed to the model. The data augmentation of randomly horizontal flip is used. We use the technique proposed in [7] to compute the stable SVD gradient. The Pre-SVD layer denotes the convolution transform of the previous layer.

C Computational Cost

Methods	FP (ms)	BP (ms)
SVD	44	95
SVD + NOG	44	97 (+2)
SVD + OLR	44	96 (+1)
SVD + OW	48 (+4)	102 (+7)
SVD + OW + NOG + OLR	49 (+5)	106 (+11)
Newton-Schulz Iteration	43	93
Vanilla ResNet-50	42	90

Table 1. Time consumption of each forward pass (FP) and backward pass (BP) measured on a RTX A6000 GPU. The evaluation is based on ResNet-50 and CIFAR100.

Table 1 compares the time consumption of a single training step for the experiment of decorrelated BN. Our NOG and OLR bring negligible computational costs to the BP (2% and 1%), while the FP is not influenced. Even when all techniques are applied, the overall time costs are marginally increased by 10%. Notice that NOG and OLR have no impact on the inference speed.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
2. Grigorieff, R.D.: A note on von neumann's trace inequalitv. Mathematische Nachrichten **151**(1), 327–328 (1991)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
4. Krizhevsky, A.: Learning multiple layers of features from tiny images. Master's thesis, University of Tront (2009)
5. Li, P., Xie, J., Wang, Q., Zuo, W.: Is second-order information helpful for large-scale visual recognition? In: ICCV (2017)
6. Mirsky, L.: A trace inequality of john von neumann. Monatshefte für mathematik **79**(4), 303–306 (1975)
7. Song, Y., Sebe, N., Wang, W.: Why approximate matrix square root outperforms accurate svd in global covariance pooling? In: ICCV (2021)