Supplementary Material of Layerwise Optimization by Gradient Decomposition for Continual Learning

Shixiang Tang[1†] Dapeng Chen[3] Jinguo Zhu[2] Shijie Yu[4] Wanli Ouyang[1]
[1]The University of Sydney, SenseTime Computer Vision Group, Australia [2]Xi'an Jiaotong University
[3]Sensetime Group Limited, Hong Kong [4]Shenzhen Institutes of Advanced Technology, CAS
tangshixiang@sensetime.com dapengchenxjtu@yahoo.com wanli.ouyang@sydney.edu.au

1. Proof of Lemma 1

Proof. Following singular value decomposition (SVD) [1], $X = U\Sigma V^\top$, where $U \in \mathbb{R}^{N\times N}$ and $V \in \mathbb{R}^{n\times n}$ are unitary matrices, $\Sigma \in \mathbb{R}^{N\times n}$ is rectangular diagonal matrix with r non-zero diagonal entries. That is, Σ has $n-r$ zero column vectors. Without loss of generality, we assume that these zero column vectors are the last $n-r$ column vectors of Σ. Thus, $X = YA$, where $Y \in \mathbb{R}^{N\times r}$ is formed by the first r column vectors of U, and $A \in \mathbb{R}^{r\times n}$ is formed by the first r row vectors of $\Sigma V^\top$. The resulting Y and A satisfy that $Y^\top Y = \mathrm{I}$ and $\mathrm{rank}(A) = r$.

As $\mathrm{rank}(A) = r$, $AA^\top \in \mathbb{R}^{r\times r}$ is full rank. Therefore, for any $v \in \mathbb{R}^N$, $v^\top X = v^\top YA = \mathbf{0}$ implies $v^\top Y = \mathbf{0}$. On the other hand, $v^\top Y = \mathbf{0}$ implies $v^\top YA = v^\top X = \mathbf{0}$, which completes the proof. □

2. Solution to Equ. (10)

Since the objective $||w-g||_2^2$ is quadratic with positive definite second-order coefficient and all the constrains are affine, the optimization in Equ.(10) of the main manuscript is convex. Therefore, Karush–Kuhn–Tucker (KKT) conditions [1] applies.

The Lagrangian function of the original constraint optimization problem can be defined as follows:

$$L(w,\mu,\lambda) = \frac{1}{2}||w-g||_2^2 - \mu\bar{g}^\top w + \lambda^\top B^\top w, \qquad (1)$$

where $\mu \geq 0$. Following KKT conditions, the solution to the original optimization problem should satisfy:

$$\begin{aligned} \nabla_w L(w,\mu,\lambda) &= \mathbf{0} \\ B^\top w &= \mathbf{0} \\ -\bar{g}^\top w &\leq 0 \\ \mu &\geq 0 \\ -\mu\bar{g}^\top w &= 0 \end{aligned} \qquad (2)$$

[†]This work was done when Shixiang Tang was an intern at SenseTime.

Solving these equations, we have

$$w = \begin{cases} Pg, & \bar{g}^\top Pg \geq 0 \\ Pg - \frac{\bar{g}^\top Pg}{\bar{g}^\top P\bar{g}} P\bar{g}, & \bar{g}^\top Pg < 0 \end{cases} \qquad (3)$$

where $P = \mathbf{I} - BB^\top$.

3. Comparison with other Relaxation Methods

In this section, we first illustrate the effectiveness of PCA relaxation compared with other relaxation methods. We denote K as the rank of task-specific matrix after relaxation. In order to show that the PCA relaxation can preserve the

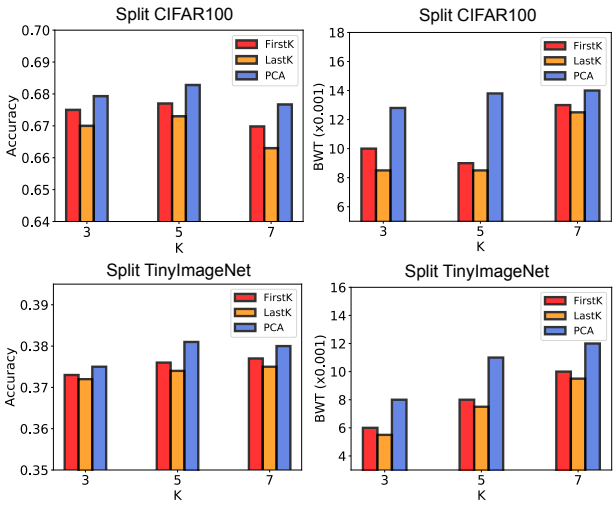

Figure 1: Average Accuracy and Backward Transfer with different relaxation methods. FirstK: preserving the first K elements of task-specific gradients matrix $\hat{G}$. LastK: preserving the last K elements of task-specific gradients matrix $\hat{G}$.

important information in the task-specific constraints, we compare PCA Relaxation with other two methods. **FirstK** is to relax the task-specific matrix by preserving its first K elements and **LastK** is to relax the task-specific matrix by

preserving its last K elements. Here, the first/last K elements correspond to the gradients of the first/last K episodic memory tasks.

As illustrated in Figure 1, the PCA relaxation method outperforms FirstK and LastK under different value of K. This phenomenon is partly due to that PCA relaxation preserves more knowledge than other methods. This explanation can be confirmed by larger BWT of PCA Relaxation at different K. In addition, We observe that FirstK outperforms LastK in Figure 1. This phenomenon may result from that tasks learnt earlier are easier to be forgotten than those tasks learnt later.

4. Proof of P is positive semidefinite

$P = \mathrm{I}_N - BB^\top$, where $B \in \mathbb{R}^{N\times r}$, $r < N$ and $B^\top B = \mathrm{I}_r$. The following proof indicates that P is positive semidefinite.

Proof. Following singular value decomposition [**?**], $B = U\Sigma V^\top$, where $U \in \mathbb{R}^{N\times N}$ and $V \in \mathbb{R}^{r\times r}$ are unitary matrices, $\Sigma \in \mathbb{R}^{N\times r}$ is rectangular diagonal matrix with r non-negative real numbers on the diagonal.

We first show that $\Sigma^\top\Sigma = \mathrm{I}_r$. Since $B^\top B = \mathrm{I}_r$, $V\Sigma^\top\Sigma V^\top = \mathrm{I}_r$. Since V is unitary, $\Sigma^\top\Sigma = \mathrm{I}_r$.

We then show that $\forall v \in \mathbb{R}^N$, $v^\top BB^\top v \le v^\top v$. Since Σ is rectangular diagonal and $\Sigma^\top\Sigma = \mathrm{I}_r$, $\Sigma\Sigma^\top = \begin{bmatrix}\mathrm{I}_r & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}$. Therefore,

$$\begin{aligned} v^\top BB^\top v &= v^\top U\Sigma\Sigma^\top U^\top v \\ &= (v^\top U\Sigma\Sigma^\top U^\top v) \\ &= (\Sigma\Sigma^\top U^\top vv^\top U) \\ &\le (U^\top vv^\top U) \\ &= (v^\top UU^\top v) \\ &= v^\top UU^\top v \\ &= v^\top v. \end{aligned} \tag{4}$$

Thus, $\forall v \in \mathbb{R}^N$, $v^\top Pv = v^\top v - v^\top BB^\top v \ge 0$, which completes the proof. □

5. Computational complexity reduction by LGU.

The main cost of our algorithm stems from Equ.(11) in the main text. We take (d) and (f) in Table 1 to analyze the time complexity of layerwise update. Denote T as the total task number and N as the number of model parameters. The total time cost of (d) is $O((T+1)N^2 + (2T^2+1)N)$: $O(2T^2N)$ for calculating B by Schmidt process, $O(TN^2)$ for calculating P, and $O(N^2+N)$ for getting w. Similarly, the total time cost of (f) is roughly $O((T+1)N^2/L + (2T^2+1)N)$, where L is the number of network layers. Since $L > 1$, the time cost of (f) is possibly less than that of (d). By implementing 3-epochs setting experiments on MNIST, we found the running time of (d) and (f) for 20 tasks was 286.3s and 253.4s respectively, which empirically verified our analysis.

References

[1] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 1