Supplementary Material to "An Embedded Feature Whitening Approach to Deep Neural Network Optimization"

Hongwei Yong and Lei Zhang

The Hong Kong Polytechnic University
{cshyong,cslzhang}@comp.polyu.edu.hk

In this supplementary file, we provide:

- The proof that the solution of Eq. (2) in the main paper is the ZCA whitening transformation (please refer to Section 3.1 in the main paper);
- Summary of the proposed W-SGDM and W-Adam algorithms (please refer to Section 4.2 in the main paper);
- Hyperparameter settings on CIFAR100/CIFAR10 and ImageNet, and the tuning of momentum α (please refer to Section 4.2 in the main paper).

1 ZCA Transformation

The solution of the following objective function

$$\min_{\mathbf{T}} ||\mathbf{X} - \Phi(\mathbf{X})||_2^2, \quad s.t. \quad \Phi(\mathbf{X}) = \mathbf{T}\mathbf{X}, \quad \frac{1}{N}\Phi(\mathbf{X})\Phi(\mathbf{X})^T = \mathbf{I} \tag{1}$$

is $\mathbf{T} = (\mathbf{X}\mathbf{X}^T/N)^{-\frac{1}{2}}$, which is just the ZCA whitening formulation.

Proof. Suppose that $\mathbf{U}\mathbf{D}\mathbf{U}^T$ is the SVD decomposition of $\mathbf{\Sigma} = \mathbf{X}\mathbf{X}^T/N$. From the constraint that $\frac{1}{N}\Phi(\mathbf{X})\Phi(\mathbf{X})^T = \mathbf{I}$, we know $\mathbf{T}\mathbf{\Sigma}\mathbf{T}^T = \mathbf{I}$, and consequently we have $\mathbf{T} = \mathbf{M}\mathbf{D}^{-\frac{1}{2}}\mathbf{U}^T$, where $\mathbf{M}$ is an arbitrary orthogonal matrix with $\mathbf{M}\mathbf{M}^T = \mathbf{M}^T\mathbf{M} = \mathbf{I}$.

The objective to be minimized in Eq. (1) is

$$\begin{aligned} & ||\mathbf{X} - \mathbf{T}\mathbf{X}||_2^2 \\ =& tr((\mathbf{X} - \mathbf{T}\mathbf{X})(\mathbf{X} - \mathbf{T}\mathbf{X})^T) \\ =& tr(\mathbf{X}\mathbf{X}^T - \mathbf{X}\mathbf{X}^T\mathbf{T}^T - \mathbf{T}\mathbf{X}\mathbf{X}^T + \mathbf{T}\mathbf{X}\mathbf{X}^T\mathbf{T}^T) \\ =& tr(\mathbf{X}\mathbf{X}^T) + tr(\mathbf{T}\mathbf{X}\mathbf{X}^T\mathbf{T}^T) - 2tr(\mathbf{T}\mathbf{X}\mathbf{X}^T) \\ =& N \cdot tr(\mathbf{\Sigma}) + N \cdot tr(\mathbf{I}) - 2N \cdot tr(\mathbf{T}\mathbf{\Sigma}). \end{aligned} \tag{2}$$

The first two terms of Eq. (2) are independent of $\mathbf{T}$. Therefore, minimizing the objective in Eq. (1) w.r.t. $\mathbf{T}$ is to maximize the last term in Eq. (2):

$$
\begin{aligned}
&\max_{\mathbf{T}} tr(\mathbf{T}\boldsymbol{\Sigma}) \\
&= \max_{\mathbf{M}\mathbf{M}^T=\mathbf{I}} tr(\mathbf{M}\mathbf{D}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{U}^T) \\
&= \max_{\mathbf{M}\mathbf{M}^T=\mathbf{I}} tr(\mathbf{M}\mathbf{D}^{\frac{1}{2}}\mathbf{U}^T) \\
&= \max_{\mathbf{M}\mathbf{M}^T=\mathbf{I}} tr(\mathbf{D}^{\frac{1}{2}}\mathbf{U}^T\mathbf{M}) \\
&= \max_{\mathbf{Q}=\mathbf{U}^T\mathbf{M},\mathbf{Q}\mathbf{Q}^T=\mathbf{I}} tr(\mathbf{D}^{\frac{1}{2}}\mathbf{Q}) \\
&= \max_{\mathbf{Q}=\mathbf{U}^T\mathbf{M},\mathbf{Q}\mathbf{Q}^T=\mathbf{I}} \sum_{i}^{C} \mathbf{D}_{ii}^{\frac{1}{2}}\mathbf{Q}_{ii},
\end{aligned}
\tag{3}
$$

where $\mathbf{D}_{ii}$ and $\mathbf{Q}_{ii}$ are the i^{th} diagonal elements of $\mathbf{D}$ and $\mathbf{Q}$, respectively. Please note that $\mathbf{D}_{ii}^{\frac{1}{2}}$ is positive definite. Since $\mathbf{Q}$ is an orthogonal matrix, its diagonal elements $\mathbf{Q}_{ii} \leq 1$. Therefore, $\sum_i^C \mathbf{D}_{ii}^{\frac{1}{2}}\mathbf{Q}_{ii} \leq \sum_i^C \mathbf{D}_{ii}^{\frac{1}{2}}$. When $\mathbf{Q} = \mathbf{I}$, the equality holds, and the maximum value $\sum_i^C \mathbf{D}_{ii}^{\frac{1}{2}}$ is reached. Meanwhile, according to $\mathbf{Q} = \mathbf{U}^T\mathbf{M} = \mathbf{I}$, we have $\mathbf{M} = \mathbf{U}$. Therefore, the optimal whitening matrix for Eq. (1) is $\mathbf{T} = \mathbf{U}\mathbf{D}^{-\frac{1}{2}}\mathbf{U}^T$. ■

2 Algorithms of W-SGDM and W-Adam

We apply EFW to the two commonly used DNN optimizers, *i.e.*, SGDM and Adam (or AdamW), and name them as W-SGDM and W-Adam, respectively. The detailed algorithms of W-SGDM and W-Adam are summarized in Algorithm 1 and Algorithm 2.

It can be seen that it is easy to embed EFW to common DNN optimizers. We first apply the EFW step to modify the gradient of weight, and then follow the updating rule of the optimizer to update the weight with the modified gradient.

3 Hyperparameter Setting

3.1 Learning rate and weight decay

As we mentioned in Section 3.5 of the main paper, with the gradient norm recovery operation, W-SGDM and W-Adam can directly adopt the default learning rate and weight decay of SGDM and Adam (or AdamW), respectively. The results of W-SGDM and W-Adam on CIFAR100/10 with the same learning rate and weight decay as SGDM and AdamW, respectively, are shown in Table 1. We can see that W-SGDM and W-Adam indeed achieve remarkable performance gains over SGDM and AdamW.

Of course, finetuning the learning rate (LR) and weight decay (WD) of W-SGDM and W-Adam around the default settings of SGDM and Adam/AdamW can further boost the performance. The finetuned settings of LR and WD, as well as the used weight decay methods (L_2 regularization or weight decouple), of

Algorithm 1: W-SGDM

Input: T_{xx}, T_{svd}, α, ϵ, β,$\mathbf{M}_{xx}^0$, $\mathbf{T}^0$
Output: $\mathbf{W}^{(T)}$

for t=1:T **do**
  $\mathbf{G}^t = \nabla_{\mathbf{W}^t}\mathcal{L}$;
  if $t\%T_{xx} = 0$ **then**
    $\mathbf{M}_{xx}^t = \alpha\mathbf{M}_{xx}^{t-1} + (1-\alpha)\mathbf{X}^t\mathbf{X}^{tT}$
  else
    $\mathbf{M}_{xx}^t = \mathbf{M}_{xx}^{t-1}$
  end
  if $t\%T_{svd} = 0$ **then**
    $\mathbf{U}\mathbf{D}\mathbf{U}^T = \mathbf{M}_{xx}^t$
    $\mathbf{T}^t = \mathbf{U}(\mathbf{D} + \epsilon d_{max}\mathbf{I})^{-1/2}\mathbf{U}^T$
  else
    $\mathbf{T}^t = \mathbf{T}^{t-1}$
  end
  $\widehat{\mathbf{G}}^t = \mathbf{G}^t\mathbf{T}^t$
  $\tilde{\mathbf{G}}^t = \widehat{\mathbf{G}}^t\frac{||\mathbf{G}^t||_2}{||\widehat{\mathbf{G}}^t||_2}$;
  $\mathbf{M}_G^t = \beta\mathbf{M}_G^{t-1} + (1-\beta)\tilde{\mathbf{G}}^t$;
  $\mathbf{W}^{t+1} = \mathbf{W}^t - \eta\mathbf{M}_G^t$;
end

Algorithm 2: W-Adam

Input: T_{xx}, T_{svd}, α, ϵ, β_1, β_2,$\mathbf{M}_{xx}^0$, $\mathbf{T}^0$
Output: $\mathbf{W}^{(T)}$

for t=1:T **do**
  $\mathbf{G}^t = \nabla_{\mathbf{W}^t}\mathcal{L}$;
  if $t\%T_{xx} = 0$ **then**
    $\mathbf{M}_{xx}^t = \alpha\mathbf{M}_{xx}^{t-1} + (1-\alpha)\mathbf{X}^t\mathbf{X}^{tT}$
  else
    $\mathbf{M}_{xx}^t = \mathbf{M}_{xx}^{t-1}$
  end
  if $t\%T_{svd} = 0$ **then**
    $\mathbf{U}\mathbf{D}\mathbf{U}^T = \mathbf{M}_{xx}^t$
    $\mathbf{T}^t = \mathbf{U}(\mathbf{D} + \epsilon d_{max}\mathbf{I})^{-1/2}\mathbf{U}^T$
  else
    $\mathbf{T}^t = \mathbf{T}^{t-1}$
  end
  $\widehat{\mathbf{G}}^t = \mathbf{G}^t\mathbf{T}^t$
  $\tilde{\mathbf{G}}^t = \widehat{\mathbf{G}}^t\frac{||\mathbf{G}^t||_2}{||\widehat{\mathbf{G}}^t||_2}$;
  $\mathbf{M}_G^t = \beta_1\mathbf{M}_G^{t-1} + (1-\beta_1)\tilde{\mathbf{G}}^t$;
  $\mathbf{V}_G^t = \beta_2\mathbf{V}_G^{t-1} + (1-\beta_2)\tilde{\mathbf{G}}^t \odot \tilde{\mathbf{G}}^t$;
  $\widehat{\mathbf{M}}_G^t = \frac{\mathbf{M}_G^t}{1-\beta_1^t}$, $\widehat{\mathbf{V}}_G^t = \frac{\mathbf{V}_G^t}{1-\beta_2^t}$;
  $\mathbf{W}^{t+1} = \mathbf{W}^t - \eta\frac{\widehat{\mathbf{M}}_G^t}{\sqrt{\widehat{\mathbf{V}}_G^t + \epsilon_2}}$;
end

different optimizers on CIFAR100/CIFAR10 are shown in Table 2. These settings are applied to all backbone networks.

The fintuned settings of LR and WD on ImageNet are shown in Table 3. On ImageNet, we tune the LR and WD on ResNet18 and ResNet50, respectively. We also give the the training and validation accuracy curves on ImageNet with ResNet50 in Fig. 1. It can been seen that both the training and validation accuracies are improved by W-SGDM and W-AdamW over their original counterparts.

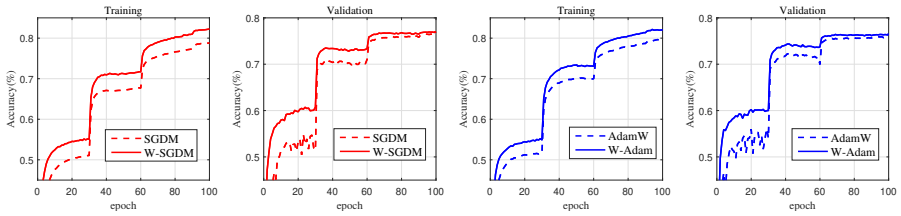

Fig. 1. The training and validation accuracy curves on ImageNet with ResNet50 backbone.

Table 1. Testing accuracies (%) of W-SGDM and W-Adam on CIFAR100/CIFAR10 by adopting the same learning rate and weight decay as SGDM and AdamW, respectively. The numbers in red color indicate the improvement of W-SGDM/W-Adam over SGDM/AdamW, respectively.

CIFAR100				
Model	SGDM	AdamW	W-SGDM	W-Adam
R18	77.20±.30	77.23±.10	78.73 ±.25(↑1.53)	78.35±.18(↑1.12)
R50	77.78±.43	78.10±.17	80.58 ±.58(↑2.80)	80.13±.12(↑2.03)
V11	70.80±.29	71.20±.29	72.90 ±.18(↑2.10)	72.93±.43(↑1.73)
D121	79.53±.19	78.05±.26	81.30 ±.19(↑1.77)	80.25±.08(↑2.20)
CIFAR10				
R18	95.10±.07	94.80±.10	95.27±.04 (↑0.17)	95.20±.12 (↑0.40)
R50	94.75±.30	94.72±.10	95.67±.22 (↑0.92)	95.57±.04 (↑0.85)
V11	92.17±.19	92.02±.08	92.98±.24 (↑0.81)	92.83±.17 (↑0.81)
D121	95.37±.17	94.80±.07	95.87±.19 (↑0.50)	95.50±.08 (↑0.70)

Table 2. The learning rate (LR), weight decay (WD) and weight decay methods for different optimizers on CIFAR100 and CIFAR10. The weight decay methods include L_2 regularization weight decay (WD1) and weight decouple (WD2).

Optimizer	SGDM	AdamW	RAdam	Ranger	Adabelief	AdaHessian	Apollo	W-SGDM	W-Adam
LR	0.1	0.001	0.001	0.001	0.001	0.15	0.01	0.05	0.0005
WD	0.0005	0.5	0.5	0.5	0.5	0.0005	0.05	0.001	1
WD method	WD1	WD2	WD2	WD2	WD2	WD2	WD2	WD1	WD2

Table 3. The learning rate (LR), weight decay (WD) and weight decay methods and for different optimizers on ImageNet.The weight decay methods include L_2 regularization weight decay (WD1) and weight decouple (WD2).

	Optimizer	SGDM	AdamW	RAdam	Ranger	Adabelief	AdaHessian	Apollo	W-SGDM	W-Adam
R18	LR	0.1	0.001	0.001	0.001	0.001	0.15	1	0.1	0.001
	WD	0.0001	0.1	0.1	0.1	0.05	0.0005	0.0001	0.0001	0.1
R50	LR	0.1	0.001	0.001	0.001	0.001	-	1	0.1	0.001
	WD	0.0001	0.1	0.05	0.1	0.1	-	0.0001	0.002	0.2
	WD method	WD1	WD2	WD2	WD2	WD2	WD2	WD1	WD1	WD2