Fine-grained Optimization of Deep Neural Networks

Mete Ozay

Introduction:

- DNNs trained using weights renormalized by the proposed method can achieve tighter bounds for theoretical generalization errors compared to using unnormalized weights. These DNNs do not have spurious local minima.

- In the proof of convergence theorems, we observe that gradients of weights should satisfy a particular normalization requirement and we employ this requirement for adaptive computation of step size of the FG-SGD. To this best of our knowledge, this is first result which also establishes the relationship between norms of weights and norms of gradients for training DNNs.

- We prove that loss functions of DNNs trained using the proposed FG-SGD converges to minima almost surely. To the best of our knowledge, our proposed FG-SGD is the first algorithm performing optimization on different collections of products of weight manifolds to train DNNs with convergence properties.

Table 1: Comparison of generalization bounds. $\mathcal{O}$ denotes big-O and $\tilde{\mathcal{O}}$ is soft-O. $\delta_{l,F}$, $\delta_{l,2}$, and $\delta_{l,2\to1}$ denotes upper bounds of the Frobenius norm $\|\omega_l\|_F \le \delta_{l,F}$, spectral norm $\|\omega_l\|_2 \le \delta_{l,2}$ and the sum of the Euclidean norms for all rows $\|\omega_l\|_{2\to1} \le \delta_{l,2\to1}$ ($\ell_{2\to1}$) of weights ω_l at the l^{th} layer of an L layer DNN using N samples. Suppose that all layers have the same width ϖ, weights have the same length $\mathcal{K}$ and the same stride $\mathfrak{s}$. Then, generalization bounds are obtained for DNNs using these fixed parameters by $\|\omega_l\|_2 = \frac{\mathcal{K}}{\mathfrak{s}}$, $\|\omega_l\|_F = \sqrt{\varpi}$ and $\|\omega_l\|_{2\to1} = \varpi$. We compute a concatenated weight matrix $\omega_{g,l} = (\omega^1_{g,l}, \omega^2_{g,l}, \dots, \omega^{|\mathfrak{g}|}_{g,l})$ for the g^{th} weight group of size $|\mathfrak{g}|, g = 1, 2, \dots, G_l, \forall l$ using a weight grouping strategy. Then, we have upper bounds of norms by $\|\omega_{g,l}\|_F \le \delta_{g,l,F} \le 1$, $\|\omega_{g,l}\|_2 \le \delta_{g,l,2} \le 1$ and $\|\omega_{g,l}\|_{2\to1} \le \delta_{g,l,2\to1} \le 1, g = 1, 2, \dots, G_l$, which are defined in Table 2

	DNNs (dynamic group scaling)
Neyshabur et al. [22]	$\mathcal{O}\Big(\frac{2^L \prod_{l=1}^{L}\prod_{g=1}^{G_l}\delta_{g,l,F}}{\sqrt{N}}\Big)$
Bartlett et al. [3]	$\tilde{\mathcal{O}}\Bigg(\frac{\prod_{l=1}^{L}\prod_{g=1}^{G_l}\delta_{g,l,2}}{\sqrt{N}}\Big(\sum_{l=1}^{L}\prod_{g=1}^{G_l}\big(\frac{\delta_{g,l,2\to1}}{\delta_{g,l,2}}\big)^{\frac{2}{3}}\Big)^{\frac{3}{2}}\Bigg)$
Neyshabur et al. [8]	$\tilde{\mathcal{O}}\Bigg(\frac{\prod_{l=1}^{L}\prod_{g=1}^{G_l}\delta_{g,l,2}}{\sqrt{N}}\sqrt{L^2\varpi\sum_{l=1}^{L}\prod_{g=1}^{G_l}\frac{\delta^2_{g,l,F}}{\delta^2_{g,l,2}}}\Bigg)$

Table 2: Comparison of norms of weights belonging to different weight manifolds. Suppose that weights $\omega^i_{g,l} \in \mathbb{R}^{A_l \times B_l}$ belonging to the g^{th} group of size $|\mathfrak{g}|, g = 1, 2, \dots, G_l, \forall l$ have the same size $A_l \times B_l$ for simplicity, and $\sigma(\omega^i_{g,l})$ denotes the top singular value of $\omega^i_{g,l}$. Let $\|\omega^i_{g,l}\|_F$, $\|\omega^i_{g,l}\|_2$, and $\|\omega^i_{g,l}\|_{2\to1}$, denote respectively the Frobenius, spectral and $\ell_{2\to1}$ norms of the weight $\omega^i_{g,l}$. Then, we have $\|\omega_{g,l}\|_F \ge (\prod_{i=1}^{|\mathfrak{g}|}\|\omega^i_{g,l}\|_F)^{1/|\mathfrak{g}|}$, $\|\omega_{g,l}\|_2 \ge (\prod_{i=1}^{|\mathfrak{g}|}\|\omega^i_{g,l}\|_2)^{1/|\mathfrak{g}|}$ and $\|\omega_{g,l}\|_{2\to1} \ge (\prod_{i=1}^{|\mathfrak{g}|}\|\omega^i_{g,l}\|_{2\to1})^{1/|\mathfrak{g}|}$.

Norms	(i) Sphere	(ii) Stiefel	(iii) Oblique
$\|\omega^i_{g,l}\|_2$	$\sigma(\omega^i_{g,l})$	1.0	$\sigma(\omega^i_{g,l})$
$\|\omega^i_{g,l}\|_F$	1.0	$(B_l)^{1/2}$	$(B_l)^{1/2}$
$\|\omega^i_{g,l}\|_{2\to1}$	1.0	$(B_l)^{1/4}$	$(B_l)^{1/4}$

Optimization using Fine-grained SGD

Algorithm 1 Optimization using FG-SGD on products manifolds of fine-grained weights.

```
Input: $T$ (number of iterations), $S$ (training set),
    $\Theta$ (set of hyperparameters), $\mathcal{L}$ (a loss function), $\mathcal{I}^l_g \subseteq \mathcal{I}_{\mathcal{G}_l}, \forall g, l$.
Initialization: Construct a collection of products of weight manifolds $\mathcal{G}_l$, initialize re-scaling
    parameters $\mathcal{R}^t_l$ and initialize weights $\omega^t_{g,l} \in \mathbb{M}_{g,l}$ with $\mathcal{I}^l_g \subseteq \mathcal{I}_{\mathcal{G}_l}, \forall m, l$.
for each iteration $t = 1, 2, \dots, T$ do
    for each layer $l = 1, 2, \dots, L$ do
        $\mathrm{grad}\mathcal{L}(\omega^t_{g,l}) := \Pi_{\omega^t_{g,l}}\Big(\mathrm{grad}_E\, \mathcal{L}(\omega^t_{g,l}), \Theta, \mathcal{R}^t_l\Big), \forall \mathcal{G}_l$.
        $v_t := h(\mathrm{grad}\mathcal{L}(\omega^t_{g,l}), r(t, \Theta)), \forall \mathcal{G}_l$.
        $\omega^{t+1}_{g,l} := \phi_{\omega^t_{g,l}}(v_t, \mathcal{R}^t_l), \forall \omega^t_{g,l}, \forall \mathcal{G}_l$.
    end for
end for
Output: A set of estimated weights $\{\omega^T_{g,l}\}_{l=1}^L, \forall g$.
```

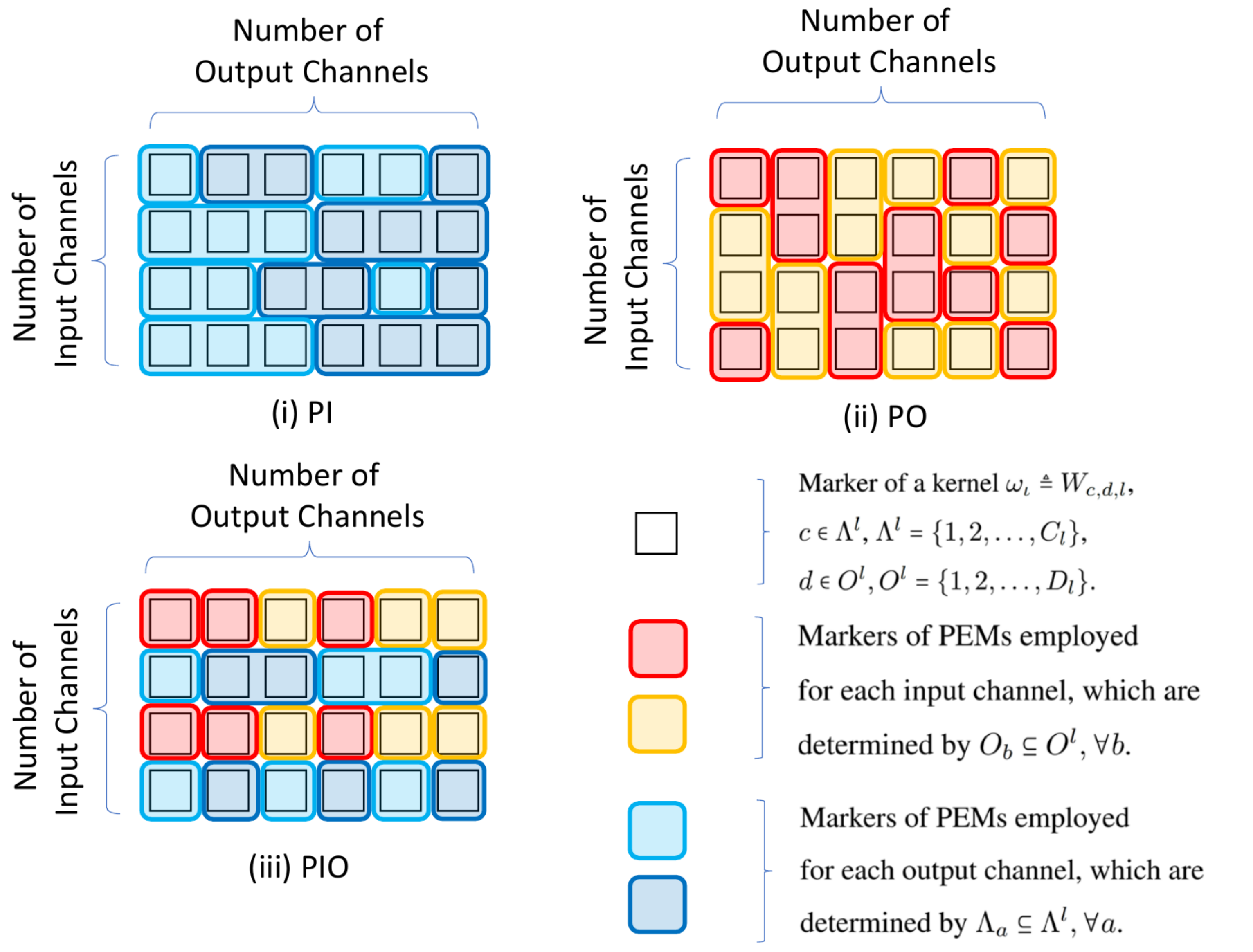
Number of
Output Channels
Number of
Input Channels
(i) PI
Number of
Output Channels
Number of
Input Channels
(ii) PO
Number of
Output Channels
Number of
Input Channels
(iii) PIO
□ Marker of a kernel $\omega_\iota \triangleq W_{c,d,l}$, $c \in \Lambda^l, \Lambda^l = \{1, 2, \dots, C_l\}$, $d \in O^l, O^l = \{1, 2, \dots, D_l\}$.
🟥 🟨 Markers of PEMs employed for each input channel, which are determined by $O_b \subseteq O^l, \forall b$.
🟦 🟦 Markers of PEMs employed for each output channel, which are determined by $\Lambda_a \subseteq \Lambda^l, \forall a$.

Figure 2: An illustration of employment of the proposed PI, PO and PIO collection strategies at the l^{th} layer of a CNN. In Section 4.2, we randomly selected indices of weights, i.e. subsets of input and output channels, according to the uniform distribution. In this example, we suppose that there are four input and six output channels. Then, 24 convolution weights are computed on in two different POMs.

Main Results:

Convergence properties of the proposed FG-SGD used to train DNNs are summarized as follows:

Convergenge to local minima: The loss function of a non-linear DNN, which employs the proposed FG-SGD, converges to a local minimum, and the corresponding gradient converges to zero almost surely (a.s.). The formal theorem and proof are given in Theorem 2 in the supplemental material.

Convergenge to global minima: Loss functions of particular DNNs such as linear DNNs, one-hidden-layer CNNs, one-hidden-layer Leaky Relu networks, nonlinear DNNs with specific network structures (e.g. pyramidal networks), trained using FG-SGD, converge to a global minimum a.s. under mild assumptions on data (e.g. being distributed from Gaussian distribution, normalized, and realized by DNNs). The formal theorem and proof of this result are given in Corollary 1 in the supp. mat. The proof idea is to use the property that local minima of loss functions of these networks are global minima under these assumptions, by employing the results given in the recent works [27–36].

An example for adaptive computation of step size: Suppose that $\mathbb{M}_\iota$ are identified by $n_\iota \ge 2$ dimensional unit sphere, or the sphere scaled by the proposed scaling method. If step size is computed using (3) with

$$\imath(\omega^t_{G^m_l}) = (\max\{1, (R^t_{G^m_l})^2(2 + R^t_{G^m_l})^2\})^{\frac{1}{2}}, \quad (7)$$

then the loss function converges to local minima for a generic class of nonlinear DNNs, and to global minima for DNNs characterized in Corollary 1. The formal theorem and proof of this result are given in Corollary 2 in the supp. mat.

Table 3: Mean ± standard deviation of classification error (%) are given for results obtained using Resnet-50/101, SENet-Resnet-50/101, and 110-layer Resnets with constant depth (RCD) on Imagenet.

Model	**Imagenet(Resnet-50)**	**Imagenet(SENet-Resnet-50)**
Euc.	24.73 ± 0.32	23.31± 0.55
St	23.77 ± 0.27	23.09 ± 0.41
POMs of St	23.61 ± 0.22	22.97 ± 0.29
PIO (Sp+Ob+St)	23.04 ± 0.10	22.67 ± 0.15
PIO (Sp+Ob+St+Euc.)	22.89 ± 0.08	22.53 ± 0.11
(Additional results)	**Imagenet(Resnet-101)**	**Imagenet(SENet-Resnet-101)**
Euc.	23.15 ± 0.09	22.38 ± 0.30
PIO (Sp+Ob+St)	22.83 ± 0.06	21.93 ± 0.12
PIO (Sp+Ob+St+Euc.)	22.75 ± 0.02	21.76 ± 0.09

https://github.com/meteozay/fgo_dnns