Supplementary Material for: Deep Learning Model Compression with Rank Reduction in Tensor Decomposition

Wei Dai, *Student Member, IEEE,* **Jicong Fan**, *Member, IEEE* **Yiming Miao**, *Member, IEEE* and **Kai Hwang**, *Life Fellow, IEEE,*

I. Introduction

This supplemental materials contain all detailed proofs in the original paper.

A. Low-Rank Convolution.

Proposition 1. *Suppose the kernel of a convolutional layer $\mathcal{W} \in \mathbb{R}^{q\times c\times h\times e}$ with a multilinear rank of (r_1, r_2, r_3, r_4), the low-rank convolution process can be expressed as:*

$$\mathcal{Y}_{(1)} = \boldsymbol{U}_1 \mathcal{G}_{(1)} (\boldsymbol{U}_2 \otimes \boldsymbol{U}_3 \otimes \boldsymbol{U}_4)^\intercal \cdot \textit{im2col}(\mathcal{X}), \tag{1}$$

where $\otimes$ is the Kronecker product, $\mathcal{G} \in \mathbb{R}^{r_1\times r_2\times r_3\times r_4}$, $\boldsymbol{U}_1 \in \mathbb{R}^{q\times r_1}$, $\boldsymbol{U}_2 \in \mathbb{R}^{c\times r_2}$, $\boldsymbol{U}_3 \in \mathbb{R}^{h\times r_3}$, and $\boldsymbol{U}_4 \in \mathbb{R}^{w\times r_4}$.

Proof. Mathematically, the convolution using im2col can be expressed as

$$\mathcal{Y}_{(1)} = \mathcal{W}_{(1)} \cdot \text{im2col}(\mathcal{X}), \tag{2}$$

where $\text{im2col}(\mathcal{X}) \in \mathbb{R}^{chw\times h_o w_o}$ and $\mathcal{W}_{(1)} \in \mathbb{R}^{q\times chw}$ is mode-1 unfold of tensor $\mathcal{W}$.

The kernel $\mathcal{W}$ with multilinear rank of (r_1, r_2, r_3, r_4) can be decomposed as $\mathcal{W} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 \times_4 \boldsymbol{U}_4$. Then, we can express the mode-1 unfolding of $\mathcal{W}$ as

$$\mathcal{W}_{(1)} = \boldsymbol{U}_1 \mathcal{G}_{(1)} (\boldsymbol{U}_2 \otimes \boldsymbol{U}_3 \otimes \boldsymbol{U}_4)^\intercal.$$

By plugging in Eq. (2), we prove the proposition. $\mathcal{Y}_{(1)} = \boldsymbol{U}_1 \mathcal{G}_{(1)} (\boldsymbol{U}_2 \otimes \boldsymbol{U}_3 \otimes \boldsymbol{U}_4)^\intercal \cdot \text{im2col}(\mathcal{X})$. Q.E.D.

Figure 3 illustrates the low-rank convolution. We avoid reconstructing the original kernel by performing matrix multiplication from right to left instead. We further obtain fast computation by applying some mathematical tricks. See the proof of Theorem 4theorem.4.

As for the fully-connected layer, consider the weight $\boldsymbol{W} \in \mathbb{R}^{q\times c}$, input vector $\boldsymbol{x} \in \mathbb{R}^c$, and the output vector $\boldsymbol{y} \in \mathbb{R}^q$, then $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$. We consider the weight in the fully-connected layer is also in high dimensional space, e.g. we convert $\boldsymbol{W}$ into 4-D space $\mathcal{W} \in \mathbb{R}^{q\times c\times 1\times 1}$. Then, the output vector can be calculated by $\boldsymbol{y} = \mathcal{W}_{(1)}\boldsymbol{x}$. Similar to the low-rank convolutional layer, we derive

Proposition 2. *Suppose the weight of a fully connected layer $\mathcal{W} \in \mathcal{R}^{q\times c\times 1\times 1}$ has multilinear rank of $(r_1, r_2, 1, 1)$, the low-rank forward process can be expressed as:*

$$\boldsymbol{y} = \boldsymbol{U}_1 \mathcal{G}_{(1)} \boldsymbol{U}_2^\intercal \boldsymbol{x},$$

where $\mathcal{G} \in \mathbb{R}^{r_1\times r_2}$, $\boldsymbol{U}_1 \in \mathbb{R}^{c\times r_1}$, and $\boldsymbol{U}_2 \in \mathbb{R}^{q\times r_2}$.

Proof. Since $\mathcal{W}$ has multilinear rank of $(r_1, r_2, 1, 1)$, it can be decomposed as $\mathcal{W} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 \times_4 \boldsymbol{U}_4$ with $\mathcal{G} \in \mathbb{R}^{r_1\times r_2\times 1\times 1}$, $\boldsymbol{U}_1 \in \mathbb{R}^{q\times r_1}$, $\boldsymbol{U}_2 \in \mathbb{R}^{c\times r_2}$, $\boldsymbol{U}_3 \in \mathbb{R}^{1\times 1}$, and $\boldsymbol{U}_4 \in \mathbb{R}^{1\times 1}$, we can see $\boldsymbol{U}_3$ and $\boldsymbol{U}_4$ are essentially scalar. By setting them to 1, the decomposition can be simplified as

$$\mathcal{W} = \mathcal{G} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2.$$

After tensor mode-1 unfolding and apply $\mathcal{W}_{(0)} = \boldsymbol{U}_1 \mathcal{G}_{(1)} \boldsymbol{U}_2$, we have $\boldsymbol{y} = \boldsymbol{U}_1 \mathcal{G}_{(1)} \boldsymbol{U}_2^\intercal \boldsymbol{x}$. Q.E.D.

B. Proof of Lemma 1.

Proof. We summarize the update rule of the proposed scheme in the following.

$$\begin{aligned}
\mathcal{W}^t &= g(\mathcal{H}^t), \\
\hat{\mathcal{W}}^t &= \mathcal{W}^t - \eta_1 \nabla l(\mathcal{W}^t), \\
\mathcal{H}^{t+1} &= c(\hat{\mathcal{W}}^t) - \eta_2 \nabla l(c(\hat{\mathcal{W}}^t)), \\
\mathcal{W}^{t+1} &= g(\mathcal{H}^{t+1}).
\end{aligned} \tag{3}$$

By rewriting the update rule in Eq. (3) as

$$\mathcal{W}^{t+1} = \mathcal{W}^t - \eta_1 \nabla l(\mathcal{W}^t) + \mathcal{E}^t, \tag{4}$$

where $\mathcal{E}^t = g(\mathcal{H}^{t+1}) - \mathcal{W}^t + \eta_1 \nabla l(\mathcal{W}^t)$ denotes the low-rank update error on the t-th iteration.

Then, we can bound the low-rank update error $\mathcal{E}^t$. We first explicitly derive $\|\mathcal{E}^t\|_F$ as the following.

$$\begin{aligned}
\|\mathcal{E}^t\|_F &= \|\mathcal{E}^t_{(1)}\|_F \\
&= \|g(\mathcal{H}^{t+1})_{(1)} - \mathcal{W}^t_{(1)} + \eta_1 \nabla l(\mathcal{W}^t)_{(1)}\|_F.
\end{aligned}$$

Wei Dai, Yiming Miao, and Kai Hwang are with The Chinese University of Hong Kong, Shenzhen. E-mail: weidai@link.cuhk.edu.cn, miaoyiming@cuhk.edu.cn, hwangkai@cuhk.edu.cn.

Jicong Fan is with The Chinese University of Hong Kong, Shenzhen, and also with Shenzhen Research Institute of Big data, Shenzhen, Guangdong, China. E-mail: fanjicong@cuhk.edu.cn.

Let

$$\begin{aligned}
&g(\boldsymbol{\mathcal{H}}^{t+1})_{(1)}\\
=&g\left(c(\hat{\boldsymbol{\mathcal{W}}}^t)-\eta_2\nabla l(c(\hat{\boldsymbol{\mathcal{W}}}^t))\right)_{(1)}\\
=&\left(\hat{\boldsymbol{U}}_1^t-\eta_2\nabla l(\hat{\boldsymbol{U}}_1^t)\right)\left(\hat{\boldsymbol{\mathcal{G}}}^t-\eta_2\nabla l(\hat{\boldsymbol{\mathcal{G}}}^t)\right)_{(1)}\\
&\left(\left(\hat{\boldsymbol{U}}_2^t-\eta_2\nabla l(\hat{\boldsymbol{U}}_2^t)\right)\otimes\left(\hat{\boldsymbol{U}}_3^t-\eta_2\nabla l(\hat{\boldsymbol{U}}_3^t)\right)\otimes\right.\\
&\left.\left(\hat{\boldsymbol{U}}_4^t-\eta_2\nabla l(\hat{\boldsymbol{U}}_4^t)\right)\right)^\intercal\\
=&\hat{\boldsymbol{U}}_1^t\hat{\boldsymbol{\mathcal{G}}}_{(1)}^t(\hat{\boldsymbol{U}}_2^t\otimes\hat{\boldsymbol{U}}_3^t\otimes\hat{\boldsymbol{U}}_4^t)^\intercal+\boldsymbol{R}^t\\
=&g(c(\hat{\boldsymbol{\mathcal{W}}}^t))_{(1)}+\boldsymbol{E}^t-\boldsymbol{E}^t+\boldsymbol{R}^t\\
=&\hat{\boldsymbol{\mathcal{W}}}_{(1)}^t+\boldsymbol{R}^t-\boldsymbol{E}^t\\
=&\boldsymbol{\mathcal{W}}_{(1)}^t-\eta_1\nabla l(\boldsymbol{\mathcal{W}}^t)_{(1)}+\boldsymbol{R}^t-\boldsymbol{E}^t,
\end{aligned}$$

where $\boldsymbol{E}^t=\hat{\boldsymbol{\mathcal{W}}}_{(1)}^t-g(c(\hat{\boldsymbol{\mathcal{W}}}^t))_{(1)}$, and

$$\begin{aligned}
\boldsymbol{R}^t=&-\eta_2\nabla l(\hat{\boldsymbol{U}}_1^t)\hat{\boldsymbol{\mathcal{G}}}_{(1)}^t(\hat{\boldsymbol{U}}_2^t\otimes\hat{\boldsymbol{U}}_3^t\otimes\hat{\boldsymbol{U}}_4^t)^\intercal-...\\
&+\eta_2^2\nabla l(\hat{\boldsymbol{U}}_1^t)\nabla l(\hat{\boldsymbol{\mathcal{G}}}^t)_{(1)}(\hat{\boldsymbol{U}}_2^t\otimes\hat{\boldsymbol{U}}_3^t\otimes\hat{\boldsymbol{U}}_4^t)^\intercal+...\\
&-\eta_2^3\nabla l(\hat{\boldsymbol{U}}_1^t)\nabla l(\hat{\boldsymbol{\mathcal{G}}}^t)_{(1)}(\nabla l(\hat{\boldsymbol{U}}_2^t)\otimes\hat{\boldsymbol{U}}_3^t\otimes\hat{\boldsymbol{U}}_4^t)^\intercal-...\\
&+\eta_2^4\nabla l(\hat{\boldsymbol{U}}_1^t)\nabla l(\hat{\boldsymbol{\mathcal{G}}}^t)_{(1)}(\nabla l(\hat{\boldsymbol{U}}_2^t)\otimes\nabla l(\hat{\boldsymbol{U}}_3^t)\otimes\hat{\boldsymbol{U}}_4^t)^\intercal+...\\
&-\eta_2^5\nabla l(\hat{\boldsymbol{U}}_1^t)\nabla l(\hat{\boldsymbol{\mathcal{G}}}^t)_{(1)}(\nabla l(\hat{\boldsymbol{U}}_2^t)\otimes\nabla l(\hat{\boldsymbol{U}}_3^t)\otimes\nabla l(\hat{\boldsymbol{U}}_4^t))^\intercal,
\end{aligned}$$

consists of 31 terms that are the permutation of low-rank weights and their gradients. Then, plugging it back, we have

$$\|\boldsymbol{\mathcal{E}}_{(1)}^t\|_F=\|\boldsymbol{R}^t-\boldsymbol{E}^t\|_F\leq\|\boldsymbol{R}^t\|_F+\|\boldsymbol{E}^t\|_F.$$

According to Equation (8Low-Rank Deep Learning Model Updateequation.5.8) in original paper, $\|\boldsymbol{E}^t\|_F$ is bounded that

$$\|\boldsymbol{E}^t\|_F\leq(1-\rho)\|\hat{\boldsymbol{\mathcal{W}}}^t\|_F\leq(1-\rho)\varphi.$$

Using the assumption (2) and (3), if $0\leq\eta_2\leq1$, $\mathbb{E}\left[\|\boldsymbol{R}^t\right]\|_F$ is bounded that

$$\begin{aligned}
&\mathbb{E}\left[\|\boldsymbol{R}^t\|_F\right]\\
\leq&\eta_2\left(\|\nabla l(\hat{\boldsymbol{U}}_1^t)\|_2\|\hat{\boldsymbol{\mathcal{G}}}_{(1)}^t\|_F\|\hat{\boldsymbol{U}}_2^t\|_2\|\hat{\boldsymbol{U}}_3^t\|_2\|\hat{\boldsymbol{U}}_4^t\|_2+...\right.\\
&+\|\nabla l(\hat{\boldsymbol{U}}_1^t)\|_2\|\nabla l(\hat{\boldsymbol{\mathcal{G}}}^t)_{(1)}\|_F\|\hat{\boldsymbol{U}}_2^t\|_2\|\hat{\boldsymbol{U}}_3^t\|_2\|\hat{\boldsymbol{U}}_4^t\|_2+...\\
&+\|\nabla l(\hat{\boldsymbol{U}}_1^t)\|_2\|\nabla l(\hat{\boldsymbol{\mathcal{G}}}^t)_{(1)}\|_F\|\nabla l(\hat{\boldsymbol{U}}_2^t)\|_2\|\hat{\boldsymbol{U}}_3^t\|_2\|\hat{\boldsymbol{U}}_4^t\|_2+...\\
&+\|\nabla l(\hat{\boldsymbol{U}}_1^t)\|_2\|\nabla l(\hat{\boldsymbol{\mathcal{G}}}^t)_{(1)}\|_F\|\nabla l(\hat{\boldsymbol{U}}_2^t)\|_2\|\nabla l(\hat{\boldsymbol{U}}_3^t)\|_2\|\hat{\boldsymbol{U}}_4^t\|_2+...\\
&+\|\nabla l(\hat{\boldsymbol{U}}_1^t)\|_2\|\nabla l(\hat{\boldsymbol{\mathcal{G}}}^t)_{(1)}\|_F\\
&\left.\quad\|\nabla l(\hat{\boldsymbol{U}}_2^t)\|_2\|\nabla l(\hat{\boldsymbol{U}}_3^t)\|_2\|\nabla l(\hat{\boldsymbol{U}}_4^t)\|_2\right)\\
\leq&\eta_2\left(\varphi\left(\sum_{i=1}^4\binom{4}{i}2^i\right)+G_2\left(\sum_{i=0}^4\binom{4}{i}2^i\right)\right)\\
=&\eta_2\left(80\varphi+81G_2\right),
\end{aligned}$$

where the matrix norm inequality that $\|\boldsymbol{A}\boldsymbol{B}\|_F\leq\|\boldsymbol{A}\|_2\|\boldsymbol{B}\|_F$ is applied. Then, by taking the expectation, it yields

$$\mathbb{E}\left[\|\boldsymbol{\mathcal{E}}^t\|_F\right]\leq\eta_2\left(80\varphi+81G_2\right)+(1-\rho)\varphi$$

Q.E.D.

C. Proof of Theorem 1

Proof. Since we assume the loss function $\mathcal{L}$ is with L-Lipschitz continuous gradient, we have

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^{t+1})]\leq&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]+\langle\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)],\boldsymbol{\mathcal{W}}^{t+1}-\boldsymbol{\mathcal{W}}^t\rangle\\
&+\frac{L}{2}\|\boldsymbol{\mathcal{W}}^{t+1}-\boldsymbol{\mathcal{W}}^t\|^2
\end{aligned}$$

Then, plugging in Eq. 4, we have

$$\begin{aligned}
&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^{t+1})]\\
\leq&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]+\langle\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)],-\eta_1\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]\rangle\\
&+\langle\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)],-\boldsymbol{\mathcal{E}}^t\rangle+\frac{L}{2}\|-\eta_1\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)+\boldsymbol{\mathcal{E}}^t\|^2\\
=&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]-(\eta_1-\frac{L}{2}\eta_1^2)\|\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]\|^2\\
&+(1-L\eta_1)\langle\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)],\boldsymbol{\mathcal{E}}^t\rangle+\frac{L}{2}\|\boldsymbol{\mathcal{E}}^t\|^2\\
\leq&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]-(\eta_1-\frac{L}{2}\eta_1^2)\|\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]\|^2\\
&+\|\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]\|\|\boldsymbol{\mathcal{E}}^t\|+\frac{L}{2}\|\boldsymbol{\mathcal{E}}^t\|^2\\
\leq&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]+(\frac{L}{2}\eta_1^2-\eta_1+\frac{1}{2}\|\boldsymbol{\mathcal{E}}^t\|)\|\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]\|^2\\
&+\frac{1}{2}\|\boldsymbol{\mathcal{E}}^t\|+\frac{L}{2}\|\boldsymbol{\mathcal{E}}^t\|^2.
\end{aligned}$$

Since $\exists t_0,\eta_2^{(t),L}$, s.t. $t\geq t_0,\|\boldsymbol{\mathcal{E}}^t\|\leq\eta_1^2L$, we have

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^{t+1})]\leq&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]+\left(\eta_1(L\eta_1-1)\right)\|\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]\|^2\\
&+\frac{1}{2}\|\boldsymbol{\mathcal{E}}^t\|(1+L\|\boldsymbol{\mathcal{E}}^t\|)
\end{aligned}$$

If $\eta_1\in[0,\frac{1}{2L}]$, we have

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^{t+1})]\leq&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]-\frac{1}{2}\eta_1\|\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]\|^2\\
&+\frac{1}{2}\|\boldsymbol{\mathcal{E}}^t\|(1+L\|\boldsymbol{\mathcal{E}}^t\|)
\end{aligned}$$

By selecting proper $C\geq81L(\varphi+G)$, we have $\eta_2^{(t)}\leq\frac{1}{81L(\varphi+G)}$. Then,

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^{t+1})]\leq&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]-\frac{1}{2}\eta_1\|\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]\|^2\\
&+81\eta_2(\varphi+G).
\end{aligned}$$

By using the proposition in [1] A.31, hence we prove $\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^{t+1})]$ converges to a finite value and $\sum_{t=0}^{\infty}\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)\|^2]<\infty$. Q.E.D.

D. Proof of Theorem 2

Proof. From Theorem 1, we have

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]\leq\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^{t-1})]-\frac{1}{2}\eta_1\|\mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]\|^2+a\eta_2.$$

Plugging in the Polyak-Łojasiewicz inequality, we have

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]\leq&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^{t-1})]-\mu\eta_1\left(\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]-\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^*)]\right)\\
&+a\eta_2.
\end{aligned}$$

Then,

$$\begin{aligned}&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)] - \mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^*)]\\ \leq&(1-\mu\eta_1)\left(\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^{t-1})] - \mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^*)]\right) + a\frac{1}{C^t}.\end{aligned}$$

Hence, we have

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)]-\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^*)] \leq (1-\mu\eta_1)^t\left(\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^0)] - \mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^*)]\right)\\ + a(1-\mu\eta_1)^t\sum_{i=1}^{t}\left(\frac{1}{(1-\mu\eta_1)C}\right)^i\end{aligned}$$

If $C > \frac{1}{1-\mu\eta_1}$, we further have

$$\begin{aligned}&\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^t)] - \mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^*)]\\ \leq&(1-2\mu a)^t\left(\mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^0)] - \mathbb{E}[\mathcal{L}(\boldsymbol{\mathcal{W}}^*)] + a\frac{(1-\mu\eta_1)C}{(1-\mu\eta_1)C-1}\right)\end{aligned}$$

Since $\mu \leq L$, we have $0 \leq \eta_1 \leq \frac{1}{2L} \leq \frac{1}{\mu}$, and $1-\mu\eta_1 \in (0,1)$. When t goes to ∞, the RHS converges to 0. Therefore, our iteration will converge to the optimal with a linear convergence speed. Q.E.D.

E. Proof of Theorem 4

Proof. For the uncompressed case, the computation is essentially matrix-matrix multiplication between matrices with size $n \times n^3$ and $n^3 \times m$ according to Eq. (2). So, the computation complexity is $\Theta(mn^4)$.

In the case of low-rank convolutional layer, there is an efficient way to implement the Eq. (1). Denoting $\text{im2col}(\boldsymbol{\mathcal{X}}) = \left[\text{vec}(\mathcal{P}^{(1)}), \text{vec}(\mathcal{P}^{(2)}), ..., \text{vec}(\mathcal{P}^{(m)})\right]$ where $\text{vec}(\cdot)$ is the vectorization operation and $\mathcal{P}^{(m)} \in \mathbb{R}^{n\times n\times n}$ is the patch of image to be convolved, then by applying the matrix equations in Kronecker product [2], for each $k = 1, 2, ..., m$ we have

$$(\boldsymbol{U}_2 \otimes \boldsymbol{U}_3 \otimes \boldsymbol{U}_4)^\intercal \cdot \text{vec}(\mathcal{P}^{(k)}) = (\boldsymbol{U}_3 \otimes \boldsymbol{U}_4)^\intercal (\mathcal{P}^{(k)}_{(1)})^\intercal \boldsymbol{U}_2. \tag{5}$$

Furthermore, applying the matrix equations again, we have

$$(\boldsymbol{U}_3 \otimes \boldsymbol{U}_4)^\intercal (\mathcal{P}^{(k)}_{(1)})^\intercal_{:,i} = \boldsymbol{U}_4^\intercal (\mathcal{P}^{(k)}_{i,:,:})^\intercal \boldsymbol{U}_3,\ \forall i = 1,2,...,n. \tag{6}$$

Hence, to compute $(\boldsymbol{U}_2 \otimes \boldsymbol{U}_3 \otimes \boldsymbol{U}_4)^\intercal \cdot \text{im2col}(\boldsymbol{\mathcal{X}})$, we need to compute m times Eq. (5) and mn times Eq. (6). The computation complexity is $\Theta\left(m\left(n(n^2r + nr^2) + nr^3\right)\right)$. Then, by performing matrix multiplication like $\boldsymbol{U}_1\left(\boldsymbol{\mathcal{G}}_{(1)}\left((\boldsymbol{U}_2 \otimes \boldsymbol{U}_3 \otimes \boldsymbol{U}_4)^\intercal \cdot \text{im2col}(\boldsymbol{\mathcal{X}})\right)\right)$, the total complexity is $\Theta\left(m(n^3r + n^2r^2 + nr^3 + r^4 + nr)\right)$. Hence, the speed-up ratio E is lower bounded by $\Omega\left(\frac{n^4}{r^4+n^3r+n^2r^2+nr^3+nr}\right)$.

We want $E \geq \tau$. $E \geq \frac{n^4}{r^4+n^3r+n^2r^2+nr^3+nr} \geq \frac{n^4}{r^4(n^3+n^2+2n+1)}$. If $r \leq \left(\frac{n^4}{\tau(n^3+n^2+2n+1)}\right)^{\frac{1}{4}}$, then $\frac{n^4}{r^4(n^3+n^2+2n+1)} \geq \tau$. Therefore, $E \geq \tau$. Q.E.D.

F. Corollary for High-Dimension Case

Without losing generality, we derive the complexity for the higher dimensional case.

Corollary 1. *For high-dimension convolution operation with d-dimensional tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1\times d_2\times\ldots\times d_d}$ and $d_k = n, \forall k$, the computation complexity before compression is $\Theta(mn^d)$. The computation complexity of the low-rank convolution is $\Theta\left(m\sum_{i=0}^{d-1}n^ir^{d-i} + mnr\right)$. The speed-up ratio is $\Omega(\frac{n^d}{\sum_{i=0}^{d-1}n^ir^{d-i}+nr})$. The speedup ratio $\geq \tau$, if the multilinear rank $r \leq \left(\frac{n^d}{\tau(\sum_{i=0}^{d-1}n^i+n)}\right)^{\frac{1}{d}}$.*

The proof is similar to the 2D convolution in Theorem 4theorem.4 and is proved by mathematical induction. Hence, it is omitted.

G. Pseudo-Code of Tucker Convolutional Layer Implementation

We present a Pytorch-like pseudo-code for the Tucker convolution introduced in Theorem 4.

```
def tucker_conv2d(img, tucker_weights, dilation,
                  padding, stride):
    g, a, b, c, d = tucker_weights
    # im2col
    col_img = nn.Unfold((h, w),
                        dilation, padding,
                        stride)(img)

    r2, in_c = b.T.shape
    r3, h = c.T.shape
    r4, w = d.T.shape
    batch_size, in_cxhxw, m = col_img.shape

    p = torch.movedim(col_img, (0, 1, 2), (0, 2, 1))
    p = p.view(batch_size, m, in_c, h, w)
    # (r3 h,nm in_c h w,fe->nm in_c r3 r4)
    tem_re = torch.einsum("cd,nmbdf,fe->nmbce", c.T,
                          p, d)
    # (n, m, b, r3xr4)
    tem_re = tem_re.view(batch_size, m, in_c,
                         r3 * r4)
    tem_re = torch.matmul(b.T, tem_re)
    tem_re = tem_re.view(batch_size, m,
                         r3 * r4 * r2)
    tem_re = torch.movedim(tem_re, (0, 1, 2),
                           (0, 2, 1))

    g_0 = tl.unfold(g, 0)
    tem_result = torch.matmul(g_0, tem_re)
    y = tem_result.view(-1, out_c, H_out, W_out)
    return y
```

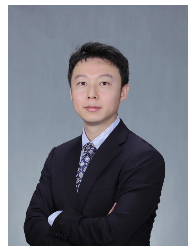
```
def tucker2_conv2d(img, tucker_weights, dilation,
                   padding, stride):
    tucker2_core, a, b = tucker_weights
    conv1_w = b.T.view(r2, in_channels, 1, 1)
    conv2_w = tucker2_core
    conv3_w = a.view(out_channels, r1, 1, 1)

    y = F.conv2d(x, conv1_w, bias=None)
    y = F.conv2d(y, conv2_w, dilation=dilation,
                 padding=padding,
                 stride=stride, bias=None)
    y = F.conv2d(y, conv3_w, bias=bias)
    return y
```

REFERENCES

[1] D. P. Bertsekas, Nonlinear programming, 3rd edition. Athena Scientific, 2016.

[2] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” SIAM review, vol. 51, no. 3, pp. 455–500, 2009.

Wei Dai received B.S. degree from the Chinese University of Hong Kong, Shenzhen in 2019 and the Master degree from University of Minnesota, Twin Cities in 2020. He is currently pursuing Ph.D. degree in Computer and Information Engineering at the Chinese University of Hong Kong, Shenzhen. His research interests include distributed cloud/edge computing, federated learning, and artificial intelligence.



Jicong Fan received his B.E and M.E degrees in Automation and Control Science & Engineering, from Beijing University of Chemical Technology, Beijing, P.R., China, in 2010 and 2013, respectively. From 2013 to 2015, he was a research assistant at the University of Hong Kong. He received his Ph.D. degree in Electronic Engineering, from City University of Hong Kong, Hong Kong S.A.R. in 2018. From 2018.01 to 2018.06, he was a visiting scholar at the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, USA. From 2018.10 to 2020.07, he was a Postdoc Associate at the School of Operations Research and Information Engineering, Cornell University, Ithaca, USA. Currently, he is a Research Assistant Professor at the School of Data Science, The Chinese University of Hong Kong (Shenzhen) and Shenzhen Research Institute of Big Data, Shenzhen, China. His research interests include statistical process control, signal processing, computer vision, optimization, and machine learning.

Yiming Miao received her Ph.D. degree in Computer Architecture from Huazhong University of Science and Technology, Wuhan, China in 2021. She also received her B.Sc. degree in Computer Science and Technology from Qinghai University, Xining, China in 2016. She is currently a Research Assistant Professor with the School of Data Science, The Chinese University of Hong Kong, Shenzhen, China. Her research interests include the internet of things, edge computing, and communication system.

Kai Hwang is a Presidential Chair Professor at the Chinese University of Hong Kong (CUHK), Shenzhen, China. He received the Ph.D. in EECS from the University of California at Berkeley. He has worked at Purdue University and University of Southern California for many years prior joining the CUHK in 2018.

Dr. Hwang has published 10 scientific books and over 280 scientific papers. An IEEE Life Fellow, He has received the Outstanding Achievement Award in 2005 from China Computer Federation and the Lifetime Achievement Award from IEEE CloudCom 2012. In 2020, he received the Tenth Wu Wenjun Artificial Intelligence Natural Science Award from China's Artificial Intelligence Association for his recent work on AI-oriented clouds/datacenters.