Error Preserving Correction: A Method for CP Decomposition at a Target Error Bound [1]
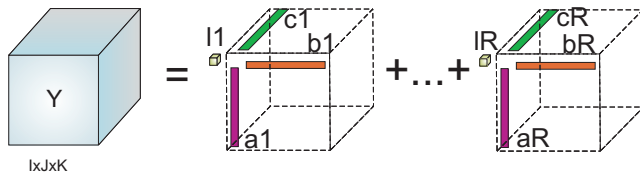
Anh-Huy Phan, Petr Tichavský and Andrzej Cichocki

Center for Computational and Data-Intensive Science and Engineering (CDISE)
Skolkovo Institute of Science and Technology (SKOLTECH), Moscow, Russia
Czech Academy of Sciences, Institute of Information Theory and Automation, Prague, Czech Republic

February 10, 2022

CANDECOMP/PARAFAC (CPD)

IxJxK

- ▶ Canonical Polyadic Decomposition (CPD): an extension of matrix factorization to multiway data, which represents higher-order tensors through multi-way operations over their latent components.
- ▶ CPD has been extended with various constraints with many efficient algorithms
- ▶ Despite the great successes, there are still many challenging problems in CPD
- ▶ *Degeneracy*

Convolutional Neural Networks I

Model	Params	SIZE(MB)
SimpleNet	6.4M	24.4
SqueezeNet	1.25M	4.7
Inception v4*	42.71M	163
Inception v3*	23.83M	91
Incep-Resv2*	55.97M	214
ResNet-152	60.19M	230
ResNet-50	25.56M	97.70
AlexNet	60.97M	217.00
GoogleNet	7M	40
VGG16	138.36M	512.2

- ▶ Deep neural networks show excellent performance in solving complex AI tasks in wide areas including computer vision, speech recognition and natural language processing,

- ▶ Most state-of-the-art NNs are Overparameterized, contain a huge number of parameters and thus computationally demanding

► Common approaches to redundancy reduction in NNs include
sparsification (prunning),
low-rank approximation to the weights within the convolutional and fully
connected layers,
prunning layers,
scaling
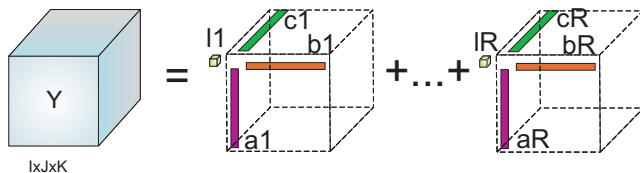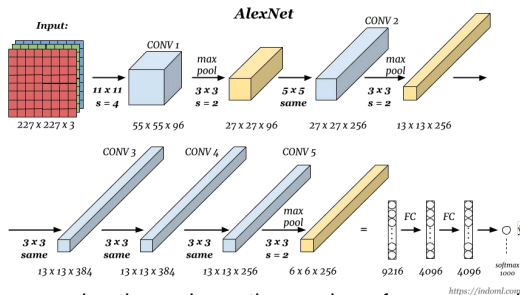building up a deep network from shallow networks with share parameters

Figure: CANDECOMP/PARAFAC (CPD) as a tool to extract rank-1 patterns from multiway data or compression of convolutional tensors.

Low-rank tensor factorization

Factorization of weights in convolutional and fully connected layers can replace the dense and large kernels by network of small kernels.

This method reduces the number of parameters in CNNs.

Canonical polyadic tensor decomposition is one of TDs.

AlexNet

https://indoml.com

- ▶ Low-rank approximation reduces the number of parameters in convolutional layers. Thereby accelerate the inference of the network (Jaderberg et. al, 2014)

- ▶ However Lebedev et. al, 2015 observed that the fine-tuning of the whole network was *unstable*.

- ▶ Tensor decomposition will not be much efficient if NNs with factorized layers is unstable and sensitive to the change of the parameters.

Indeed it is not straighforward to apply TDs to compress CNNs.
What is the main reason?

Degeneracy

▶ **Tensor Rank** the minimal rank-one tensor terms to fully represent the tensor
The following tensor of size $2 \times 2 \times 2$

$$\mathcal{A} = a_1 \circ a_2 \circ a_3 + a_1 \circ b_2 \circ b_3 - b_1 \circ a_2 \circ b_3 + b_1 \circ b_2 \circ a_3,$$

has rank-3 over the real and rank-2 and a complex rank 3

$$\mathcal{A} = \frac{1}{2}(\bar{z}_1 \circ z_2 \circ \bar{z}_3 + z_1 \circ \bar{z}_2 \circ z_3),$$

$z_k = a_k + i b_k$

▶ **Border Rank**
If there exists a sequence of tensors of rank at most $r < s$ whose limit is $\mathcal{A}$, the least value of s is the border rank of $\mathcal{A}$. The following tensor is of rank-3 but its border rank 2

$$\mathcal{A} = \boldsymbol{u} \circ \boldsymbol{u} \circ \boldsymbol{v} + \boldsymbol{u} \circ \boldsymbol{v} \circ \boldsymbol{u} + \boldsymbol{v} \circ \boldsymbol{u} \circ \boldsymbol{u}$$

with $\|\boldsymbol{u}\| = \|\boldsymbol{v}\| = 1$ and $\boldsymbol{u}^T \boldsymbol{v} \neq 1$,
$\mathcal{A}$ can be approximated with an arbitrary precision by rank-2 tensors $\mathcal{A}_n$ as $n \to \infty$

$$\begin{aligned}
\mathcal{A}_n &= n(\boldsymbol{u} + \frac{1}{n}\boldsymbol{v}) \circ (\boldsymbol{u} + \frac{1}{n}\boldsymbol{v}) \circ (\boldsymbol{u} + \frac{1}{n}\boldsymbol{v}) - n\boldsymbol{u} \circ \boldsymbol{u} \circ \boldsymbol{u} \\
&= \boldsymbol{u} \circ \boldsymbol{u} \circ \boldsymbol{v} + \boldsymbol{u} \circ \boldsymbol{v} \circ \boldsymbol{u} + \boldsymbol{v} \circ \boldsymbol{u} \circ \boldsymbol{u} \\
&\quad + \frac{1}{n}(\boldsymbol{u} \circ \boldsymbol{v} \circ \boldsymbol{v} + \boldsymbol{v} \circ \boldsymbol{u} \circ \boldsymbol{v} + \boldsymbol{v} \circ \boldsymbol{v} \circ \boldsymbol{u}) + \frac{1}{n^2}\boldsymbol{v} \circ \boldsymbol{v} \circ \boldsymbol{v}
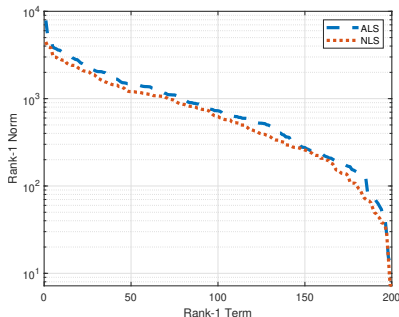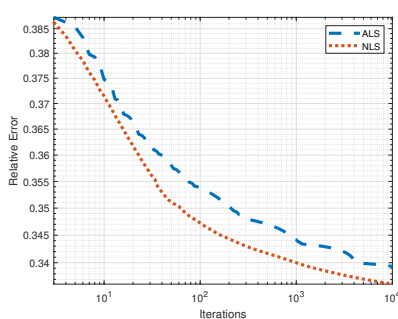\end{aligned}$$

▶ **Diverging component - Degeneracy**
Whenever $\mathcal{A}_n \to \mathcal{A}$ as $n \to \infty$, there should exist at least $1 \leq i \neq j \leq r$ such that

$$\|\mathbf{a}_{i,n}^1 \circ \mathbf{a}_{i,n}^2 \circ \cdots \circ \mathbf{a}_{i,n}^d\|_F \to \infty$$

Degeneracy is often encountered when attempting to approximate a tensor using numerical optimization algorithms
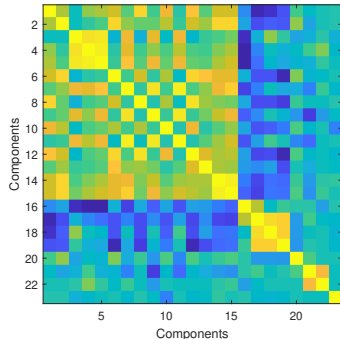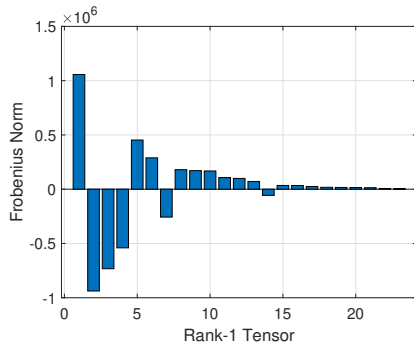
ALS: alternating least squares algorithm

NLS: nonlinear least squares algorithm (TensorLab)

Algorithms converge slowly to false local minima after 10000 iterations

while norms of the estimated rank-1 tensors are very high.

Example 2: Multiplication of two matrices of size 3×3 I

$$\text{vec}(\mathbf{AB}) = \mathcal{Y} \times_1 \text{vec}\left(\mathbf{A}^T\right)^T \times_2 \text{vec}\left(\mathbf{B}^T\right)^T$$

where $\mathcal{Y}$ is a tensor of size $9 \times 9 \times 9$ comprises only **1**s and **0**s.

- ▶ CPD of $\mathcal{Y}$ with rank-$R = 23$ often gets stuck in false local minima.
- ▶ Loading components are highly collinear,
- ▶ Some rank-1 tensors have very large norm
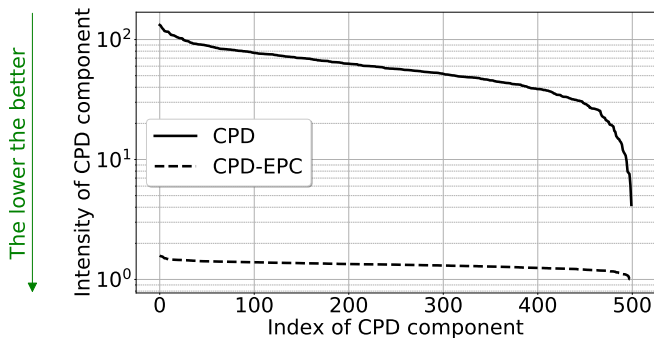
Figure: Intensity (Frobenius norm) of rank-1 tensors in CPDs of the kernel in the 4th layer of ResNet-18.
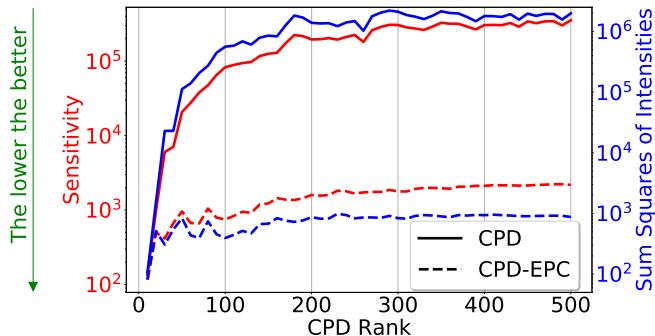
Figure: Sum of squares of the intensity and Sensitivity vs Rank of CPD.

Degeneracy occurs in most CPD of the convolutional kernels.

Degeneracy causes

- ▶ Highly collinear components which are uninterpretable
- ▶ Instable decomposition
- ▶ Algorithms are getting stuck in local minima
  running a huge number of iterations does not improve the
  convergence
- ▶ Instable finetuning and hard to converge in practice.

Lundy et al. (1989); Harshman and Lundy (1984); Mitchell and
Burdick (1994); Paatero (2000); Comon et al. (2009); Harshman
(2004); Krijnen et al. (2008a); Stegeman (2012).

- ▶ Improve stability and convergence by constraints,
  e.g., column-wise orthogonality Rayens and Mitchell (1997);
  Krijnen et al. (2008b),
  positivity or nonnegativity Paatero (1997); Lim and Comon
  (2009)

- ▶ Regularization, for example, a Tikhonov regularization

$$\min \quad \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 + \frac{\mu}{2} \sum_n \|\mathbf{U}^{(n)}\|_F^2$$

  μ is adaptively adjusted and should converge to zero.

- ▶ Iterated Tikhonov regularization Navasca et al. (2008)

$$\min \quad \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}_{[i]}\|_F^2 + \frac{\mu}{2} \sum_n \|\mathbf{U}_{[i]}^{(n)} - \mathbf{U}_{[i-1]}^{(n)}\|_F^2$$

The above-mentioned methods may be useful in certain situations,
but not always.

Two possible measures for the degeneracy degree in the CPD

Definition (Norms of the rank-1 tensorsPhan et al. (2019))

$$\mathrm{sn}(\llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket) = \sum_{r=1}^{R} \|\boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r\|_F^2 \tag{1}$$

Definition (Sensitivity Tichavský et al. (2019))

Given a tensor $\mathcal{T} = \llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket$, define the sensitivity as

$$\mathrm{ss}(\llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket) = \lim_{\sigma^2 \to 0} \frac{1}{R\sigma^2} E\{\|\mathcal{T} - \llbracket \mathbf{A} + \delta\mathbf{A}, \mathbf{B} + \delta\mathbf{B}, \mathbf{C} + \delta\mathbf{C} \rrbracket\|_F^2\} \tag{2}$$

where $\delta\mathbf{A}$, $\delta\mathbf{B}$, $\delta\mathbf{C}$ have random i.i.d. elements from $N(0, \sigma^2)$.

- ▶ Sensitivity of the tensor $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ is a measure with respect to perturbations in individual factor matrices.
- ▶ Decompositions with high sensitivity are usually useless.

Lemma

$$
\begin{aligned}
ss([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]) = & \ K \operatorname{tr}\{(\mathbf{A}^T\mathbf{A}) \circledast (\mathbf{B}^T\mathbf{B})\} + I \operatorname{tr}\{(\mathbf{B}^T\mathbf{B}) \circledast (\mathbf{C}^T\mathbf{C})\} \\
& + J \operatorname{tr}\{(\mathbf{A}^T\mathbf{A}) \circledast (\mathbf{C}^T\mathbf{C})\}.
\end{aligned} \tag{3}
$$

where $\circledast$ *denotes the Hadamard element-wise product.*

Proof.

$\square$

First, the perturbed tensor in (2) can be expressed as sum of 8 Kruskal terms

$$\llbracket \mathbf{A} + \delta\mathbf{A}, \mathbf{B} + \delta\mathbf{B}, \mathbf{C} + \delta\mathbf{C} \rrbracket = \llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket + \llbracket \delta\mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket + \llbracket \mathbf{A}, \delta\mathbf{B}, \mathbf{C} \rrbracket$$
$$+ \llbracket \mathbf{A}, \mathbf{B}, \delta\mathbf{C} \rrbracket + \llbracket \delta\mathbf{A}, \delta\mathbf{B}, \mathbf{C} \rrbracket + \llbracket \delta\mathbf{A}, \mathbf{B}, \delta\mathbf{C} \rrbracket$$
$$+ \llbracket \mathbf{A}, \delta\mathbf{B}, \delta\mathbf{C} \rrbracket + \llbracket \delta\mathbf{A}, \delta\mathbf{B}, \delta\mathbf{C} \rrbracket.$$

Since these Kruskal terms are uncorrelated and expectation of the terms composed by two or three factor matrices $\delta\mathbf{A}$, $\delta\mathbf{B}$ and $\delta\mathbf{C}$ are negligible, the expectation in (2) can be expressed in the form

$$E\{\|\mathcal{T} - \llbracket \mathbf{A} + \delta\mathbf{A}, \mathbf{B} + \delta\mathbf{B}, \mathbf{C} + \delta\mathbf{C} \rrbracket\|_F^2\} = E\{\|\llbracket \delta\mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket\|_F^2\} +$$
$$+ E\{\|\llbracket \mathbf{A}, \delta\mathbf{B}, \mathbf{C} \rrbracket\|_F^2\} + E\{\|\llbracket \mathbf{A}, \mathbf{B}, \delta\mathbf{C} \rrbracket\|_F^2\}. \tag{4}$$

Next we expand the Frobenius norm of the three Kruskal tensors

$$
\begin{aligned}
E\{\|\|[\![\delta\mathbf{A},\mathbf{B},\mathbf{C}]\!]\|\|_F^2\} &= E\{\|\,((\mathbf{C}\odot\mathbf{B})\otimes\mathbf{I})\,\mathrm{vec}(\delta\mathbf{A})\,\|^2\} \\
&= E\{\mathrm{tr}((\mathbf{C}\odot\mathbf{B})\otimes\mathbf{I})^T\,((\mathbf{C}\odot\mathbf{B})\otimes\mathbf{I})\,\mathrm{vec}(\delta\mathbf{A})\,\mathrm{vec}(\delta\mathbf{A})^T)\} \\
&= \sigma^2 I\,\mathrm{tr}((\mathbf{C}\odot\mathbf{B})^T(\mathbf{C}\odot\mathbf{B})\otimes\mathbf{I}) \\
&= RI\sigma^2\,\mathrm{tr}((\mathbf{C}^T\mathbf{C})\circledast(\mathbf{B}^T\mathbf{B})) \\
E\{\|\|[\![\mathbf{A},\delta\mathbf{B},\mathbf{C}]\!]\|\|_F^2\} &= RJ\sigma^2\,\mathrm{tr}((\mathbf{C}^T\mathbf{C})\circledast(\mathbf{A}^T\mathbf{A})) \\
E\{\|\|[\![\mathbf{A},\mathbf{B},\delta\mathbf{C}]\!]\|\|_F^2\} &= RK\sigma^2\,\mathrm{tr}((\mathbf{B}^T\mathbf{B})\circledast(\mathbf{A}^T\mathbf{A}))
\end{aligned}
$$

where $\odot$ and $\otimes$ are Khatri-Rao and Kronecker products, respectively.

Finally, we replace these above expressions into (4) to obtain the compact expression of sensitivity.
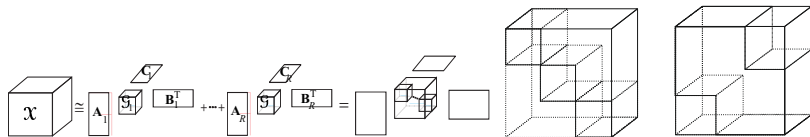
Figure: Illustration of non-overlapping and overlapping core tensors in BTD.

- ▶ CPD is not the only one tensor decomposition facing digeneracy, unstability
- ▶ Block term decompositionDe Lathauwer (2008), i.e., Tucker decompostion with block diagonal core tensor
- ▶ Most structured Tucker decomposition
- ▶ Tensor networks with loop, e.g., Tensor chain Landsberg (2012); Handschuh (2015).

$$\mathcal{T} \approx \mathcal{K} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

For Structured Tucker decomposition of order-3 tensors we define the sensitivity as

$$
\begin{aligned}
s(\mathcal{K}, \mathbf{A}, \mathbf{B}, \mathbf{C}) &= \lim_{\sigma^2 \to 0} \frac{1}{\sigma^2} \mathsf{E}\{\|[[\mathcal{K} + \delta\mathcal{K}; \mathbf{A} + \delta\mathbf{A}, \mathbf{B} + \delta\mathbf{B}, \mathbf{C} + \delta\mathbf{C}]] \\
&\quad - [[\mathcal{K}, \mathbf{A}, \mathbf{B}, \mathbf{C}]]\|_F^2\} \,,
\end{aligned}
\tag{5}
$$

where $\delta\mathcal{K}, \delta\mathbf{A}, \delta\mathbf{B}$ and $\delta\mathbf{C}$ are random Gaussian-distributed perturbations of the core tensor and the factor matrices with independently distributed elements of zero mean and variance σ^2 If some core tensor elements are constrained to be zero, the corresponding elements of $\delta\mathcal{K}$ are zero as well.

$$
\begin{aligned}
\mathrm{ss}(\Theta) &= \frac{1}{\sigma^2}\mathrm{E}\{\|[[\delta\mathcal{K}, \mathbf{A}, \mathbf{B}, \mathbf{C}]]\|_F^2\} + \frac{1}{\sigma^2}\mathrm{E}\{\|[[\mathcal{K}, \delta\mathbf{A}, \mathbf{B}, \mathbf{C}]]\|_F^2\} \\
&\quad + \frac{1}{\sigma^2}\mathrm{E}\{\|[[\mathcal{K}, \mathbf{A}, \delta\mathbf{B}, \mathbf{C}]]\|_F^2\} + \frac{1}{\sigma^2}\mathrm{E}\{\|[[\mathcal{K}, \mathbf{A}, \mathbf{B}, \delta\mathbf{C}]]\|_F^2\} \\
&= \mathbf{1}^T\mathrm{vec}\,[[\mathcal{L}, \mathbf{A}\star\mathbf{A}, \mathbf{B}\star\mathbf{B}, \mathbf{C}\star\mathbf{C}]] \\
&\quad + l_1\,\|[[\mathcal{K}, \mathbf{I}_{R_1}, \mathbf{B}, \mathbf{C}]]\|_F^2 + l_2\,\|[[\mathcal{K}, \mathbf{A}, \mathbf{I}_{R_2}, \mathbf{C}]]\|_F^2 \\
&\quad + l_3\,\|[[\mathcal{K}, \mathbf{A}, \mathbf{B}, \mathbf{I}_{R_3}]]\|_F^2\,,
\end{aligned}
\tag{6}
$$

where **1** is a vector composed of 1's, and $\mathcal{L}$ is the tensor indicating the nonzero elements of the core tensor $\mathcal{K}$.

$[S \times T \times D^2]$ $[S \times R_1]$ $[R_1 \times R_2 \times D^2]$ $[R_2 \times T]$

▶
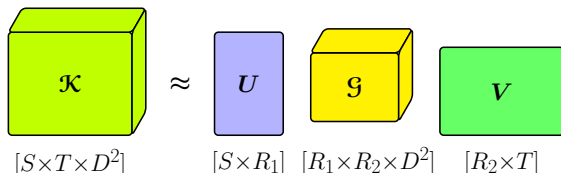
$$\mathfrak{T} \approx \mathfrak{K} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

We can always assume the TKD has orthonormal factor matrices

▶ Sensitivity for TKD depends only on the Frobenius norm of the core tensor, $\mathfrak{K}$

$$\mathrm{ss}_{TKD}(\boldsymbol{\theta}) = R_1 R_2 R_3 + (I_1 + I_2 + I_3)\|\mathfrak{K}\|_F^2. \tag{7}$$

▶ The HOOI algorithm for TKD minimizes the norm of the core tensor, i.e., it seeks a decomposition solution with minimal sensitivity.
Therefore, the orthogonal TKD is stable.

For CPD, the core tensor, $\mathcal{K}$, comprises only R nonzero elements $\lambda_1, \lambda_2, \ldots, \lambda_R$ on its diagonal.

$$
\begin{aligned}
\text{ss}_{\text{CPD}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) &= \sum_{r=1}^{R} \|\mathbf{a}_r\|^2 \|\mathbf{b}_r\|^2 \|\mathbf{c}_r\|^2 + I_1 \sum_{r=1}^{R} \lambda_r^2 \|\mathbf{b}_r\|^2 \|\mathbf{c}_r\|^2 \\
&\quad + I_2 \sum_{r=1}^{R} \lambda_r^2 \|\mathbf{a}_r\|^2 \|\mathbf{c}_r\|^2 + I_3 \sum_{r=1}^{R} \lambda_r^2 \|\mathbf{a}_r\|^2 \|\mathbf{b}_r\|^2
\end{aligned}
$$

Due to scaling ambiguity, $\mathbf{a}_r$, $\mathbf{b}_r$ and $\mathbf{c}_r$ can be normalized $\|\mathbf{a}_r\| = \|\mathbf{b}_r\| = \|\mathbf{c}_r\| = 1$.

The sensitivity for CPD is then proportional to sum-of-square of λ_r, or norm of rank-1 tensors Phan et al. (2019)

$$
\text{ss}_{\text{CPD}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = R + (I_1 + I_2 + I_3) \sum_{r=1}^{R} \lambda_r^2,
$$

Another form of the sensitivity for CPD is when the core tensor does not explicitly exist

$$\mathrm{ss}(\mathbf{A}, \mathbf{B}, \mathbf{C}) \quad = \sum_{r=1}^{R} I_1 \, \|\boldsymbol{b}_r\|^2 \|\boldsymbol{c}_r\|^2 + I_2 \, \|\boldsymbol{a}_r\|^2 \|\boldsymbol{c}_r\|^2 + I_3 \, \|\boldsymbol{a}_r\|^2 \|\boldsymbol{b}_r\|^2 \, .$$

Stable CP Decomposition I

- ▶ Novel method to deal with degeneracy
  Seeking a new tensor, $\hat{\mathcal{Y}}$ with minimal sensitivity, while
  preserving the approximation error

- ▶ Correct the rank-1 tensors if the estimation encounters
  degeneracy

$$\min_{\{\mathbf{A},\mathbf{B},\mathbf{C}\}} \quad \text{ss}([\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]) \tag{8}$$
$$\text{s.t.} \quad \|\mathcal{K} - [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]\|_F^2 \leq \delta^2 .$$

The bound, δ^2, can represent the approximation error of the
decomposition with diverging components.

Phan, Tichavsky and Cichocki, Error Preserving Correction: A Method for CP Decomposition at a Target Error Bound, IEEE Transaction on Signal Processing, 67(5), 1175-1190,2019.

Phan, et al., Stable low-rank tensordecomposition for compression of convolutional neural network, in ECCV 2020.

- Update **A** while **B** and **C** are kept fixed

$$\min_{\mathbf{A}} \quad \mathrm{tr}\{(\mathbf{A}^T\mathbf{A}) \circledast \mathbf{W}\} = \|\mathbf{A}\,\mathrm{diag}(\boldsymbol{w})\|_F^2$$

$$\text{s.t.} \quad \|\mathbf{K}_{(1)} - \mathbf{A}\mathbf{Z}^T\|_F^2 \le \delta^2,$$

where $\mathbf{Z} = \mathbf{C} \odot \mathbf{B}$ and $\mathbf{W} = \mathbf{B}^T\mathbf{B} + \mathbf{C}^T\mathbf{C}$, $\boldsymbol{w} = [\sqrt{w_{1,1}}, \ldots, \sqrt{w_{R,R}}]$ is a vector of length R taken from the diagonal of $\mathbf{W}$.

Remark

The decomposition can be reformulated as regression with bound constraint which can be solved in closed form through the quadratic programming over a sphere Phan et al. (2019)

$$\min_{\widetilde{\mathbf{A}}} \quad \|\widetilde{\mathbf{A}}\|_F^2$$

$$\text{s.t.} \quad \|\mathbf{K}_{(1)} - \widetilde{\mathbf{A}}\widetilde{\mathbf{Z}}^T\|_F^2 \le \delta^2,$$

where $\widetilde{\mathbf{A}} = \mathbf{A}\,\mathrm{diag}(\boldsymbol{w})$ and $\widetilde{\mathbf{Z}} = \mathbf{Z}\,\mathrm{diag}(\boldsymbol{w}^{-1})$.

Remark

If **B** and **C** are normalized $\|\boldsymbol{b}_r\|_2 = \|\boldsymbol{c}_r\|_2 = 1$, then all entries of the diagonal of **W** are identical, we seek a matrix, **A**, with minimal norm

$$\min_{\mathbf{A}} \quad \|\mathbf{A}\|_F^2 \tag{9}$$
$$\text{s.t.} \quad \|\mathbf{K}_{(1)} - \mathbf{A}\mathbf{Z}^T\|_F^2 \leq \delta^2.$$

EPC
Correct the rank-1 tensors if their norms are relatively high

$$
\begin{aligned}
\min \quad & f(\boldsymbol{\theta}) = \|\boldsymbol{\eta}\|_2^2 = \sum_{r=1}^{R} \eta_r^2 \\
\text{s.t.} \quad & c(\boldsymbol{\theta}) = \|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \leq \delta^2,
\end{aligned}
$$

▶ where $\boldsymbol{\theta} = [\boldsymbol{u}_r^{(n)}, \boldsymbol{\eta}]$, $\|\boldsymbol{u}_r^{(n)}\| = 1$

$$
\mathcal{Y} \approx \hat{\mathcal{Y}} = \sum_{r=1}^{R} \eta_r \, \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(N)}
$$

▶ Constraint function is nonlinear in all the factor matrices,
   but linear in parameters in a single factor matrix,
   or in non-overlapping partitions of different factor matrices
   Tichavský et al. (2016); Phan et al. (2017).

Error Preserving Correction Method II

- ▶ Rewrite the objective function and especially the constraint function in a linear form.
- ▶ Then solve the problem using the alternating update scheme or the Sequential Quadratic Programming method Boggs and Tolle (1995); Fletcher (2000); Phan et al. (2019).

EPC

$$\min \quad f(\boldsymbol{\theta}) = \|\boldsymbol{\eta}\|_2^2 \quad \text{s.t.} \quad c(\boldsymbol{\theta}) = \|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \leq \delta^2,$$

▶ At each iteration, new estimate of $\mathbf{U}^{(n)}$ is found by solving a Spherical Constrained Quadratic Programming problem

▶ Let $\mathbf{U}_\eta^{(n)} = \mathbf{U}^{(n)} \operatorname{diag}(\boldsymbol{\eta})$, the objective function becomes

$$\|\boldsymbol{\eta}\|^2 = \|\mathbf{U}^{(n)} \operatorname{diag}(\boldsymbol{\eta})\|_F^2 = \|\mathbf{U}_\eta^{(n)}\|_F^2 \,.$$

▶ EPC becomes *linear regression with bound constraint*

$$\min \qquad \|\mathbf{U}_\eta^{(n)}\|_F^2$$
$$\text{s.t.} \qquad \|\mathbf{F}_n\Sigma^{\frac{-1}{2}} - \mathbf{U}_\eta^{(n)}\mathbf{V}_n\Sigma^{\frac{1}{2}}\|_F^2 \leq \delta_n^2$$

where $\delta_n^2 = \delta^2 + \|\mathbf{F}_n\Sigma^{\frac{-1}{2}}\|_F^2 - \|\mathcal{Y}\|_F^2$

$\mathbf{T}_n = \bigodot_{k \neq n} \mathbf{U}^{(k)}$,

$\mathbf{G}_n = \mathbf{Y}_{(n)}\mathbf{T}_n$ of size $I_n \times R$,

$\Gamma_{-n} = \mathbf{T}_n^T\mathbf{T}_n = \circledast_{k \neq n}(\mathbf{U}^{(k)T}\mathbf{U}^{(k)})$ of size $R \times R$

EVD of $\Gamma_{-n} = \mathbf{V}_n\Sigma\mathbf{V}_n^T$

$\mathbf{F}_n = \mathbf{G}_n\mathbf{V}_n$,

- The inequality constraint can be replaced by an equality constraint.

Lemma

The minimizer in

$$\min_{\mathbf{x}} \quad \|\mathbf{x}\|^2 \quad s.t. \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\| \leq \delta,$$

is equivalent to the minimizer of the following optimization problem

$$\min_{\mathbf{x}} \quad \|\mathbf{x}\|^2 \quad s.t. \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\| = \delta.$$

▶ Optimization problem for update of $\mathbf{U}_\eta^{(n)}$

$$\min \qquad \|\mathbf{U}_\eta^{(n)}\|_F^2$$
$$\text{s.t.} \qquad \|\mathbf{F}_n \Sigma^{\frac{-1}{2}} - \mathbf{U}_\eta^{(n)} \mathbf{V}_n \Sigma^{\frac{1}{2}}\|_F^2 = \delta_n^2$$

▶ Perform a reparameterization

$$\mathbf{Z}_n = \frac{1}{\delta_n} (\mathbf{F}_n \Sigma^{\frac{-1}{2}} - \mathbf{U}_\eta^{(n)} \mathbf{V}_n \Sigma^{\frac{1}{2}}),$$
$$\mathbf{U}_\eta^{(n)} = (\mathbf{F}_n \Sigma^{\frac{-1}{2}} - \delta_n \mathbf{Z}_n) \Sigma^{\frac{-1}{2}} \mathbf{V}_n^T,$$

and represent the Frobenius norm of $\mathbf{U}_\eta^{(n)}$ as

$$\|\mathbf{U}_\eta^{(n)}\|_F^2 = \|\mathbf{F}_n \Sigma^{-1}\|_F^2 + \delta_n^2 \operatorname{tr}(\mathbf{Z}_n \Sigma^{-1} \mathbf{Z}_n^T) - 2\delta_n \operatorname{tr}(\mathbf{F}_n \Sigma^{\frac{-3}{2}} \mathbf{Z}_n^T).$$

Spherical Constrained Quadratic Programming (SCQP)

Instead of updating $\mathbf{U}_\eta^{(n)}$, we seek $\mathbf{Z}_n$ of size $I_n \times R$ in an SCQP for matrix-variate

$$
\begin{aligned}
\min \quad & \delta_n \operatorname{tr}(\mathbf{Z}_n \Sigma^{-1} \mathbf{Z}_n^T) - 2 \operatorname{tr}(\mathbf{F}_n \Sigma^{\frac{-3}{2}} \mathbf{Z}_n^T) \\
\text{s.t.} \quad & \|\mathbf{Z}_n\|_F^2 = 1.
\end{aligned}
$$

- The above SCQP has $I_n R$ terms and can be solved in closed-form Gander et al. (1989); Phan et al. (2019).
- We next formulate another equipvalent SCQP problem but with smaller number of parameters

A SCQP for a matrix $\mathbf{X}$ of size $I \times R$ given in the form of

$$\min_{\mathbf{X}} \quad f(\mathbf{X}) = \frac{1}{2} \operatorname{tr}(\mathbf{X}^T \mathbf{Q} \mathbf{X}) + \operatorname{tr}(\mathbf{B}^T \mathbf{X}) \quad \textit{s.t.} \quad \|\mathbf{X}\|_F^2 = 1, \qquad (10)$$

where $\mathbf{Q}$ is a positive semidefinite matrix of size $I \times I$ and has EVD $\mathbf{Q} = \mathbf{U} \operatorname{diag}(\boldsymbol{\sigma}) \mathbf{U}^T$, and $\mathbf{B}$ is of size $I \times R$.
Then the optimal solution

$$\mathbf{X}^\star = \mathbf{B}^T \mathbf{U} \operatorname{diag}(\boldsymbol{z}^\star \oslash \boldsymbol{c})$$

can be derived from the minimizer $\boldsymbol{z}^\star$ of length I of an SCQP

$$\min \quad \frac{1}{2} \boldsymbol{z}^T \operatorname{diag}(\boldsymbol{\sigma}) \boldsymbol{z} + \boldsymbol{c}^T \boldsymbol{z} \quad \textit{s.t.} \quad \boldsymbol{z}^T \boldsymbol{z} = 1, \qquad (11)$$

where $\boldsymbol{c} = [c_1, \ldots, c_I]$, $c_i = \|\mathbf{B}^T \boldsymbol{u}_i\|$.

▶ The minimizer $\mathbf{Z}_n^\star$ can be derived from the minimizer $\mathbf{z}^\star = [z_1^\star, \ldots, z_R^\star]^T$ of an SCQP of a smaller scale

$$\min \quad \delta_n \mathbf{z}^T \Sigma^{-1} \mathbf{z}^T - 2\mathbf{c}^T \mathbf{z} \quad \text{s.t.} \quad \mathbf{z}^T \mathbf{z} = 1$$

where $\mathbf{c} = [\ldots, \sigma_r^{\frac{-3}{2}} \|\mathbf{f}_r^{(n)}\|, \ldots]$.

   ▶ For $c_r \neq 0$, the r-th column of $\mathbf{Z}_n^\star$ is the r-th column of $\mathbf{F}_n$ scaled by a factor of $\frac{z_r^\star}{\|\mathbf{f}_r^{(n)}\|}$

$$\mathbf{z}_r^{(n)\star} = \frac{z_r^\star}{\|\mathbf{f}_r^{(n)}\|} \mathbf{f}_r^{(n)}.$$

   ▶ For $c_r = 0$, $\mathbf{z}_r^{(n)\star}$ can be any vector of length $\|\mathbf{z}_r^{(n)\star}\|^2 = (z_r^\star)^2$.
   ▶ If $c_r = 0$ for $r > 1$, then $z_r^\star = 0$, hence $\mathbf{z}_r^{(n)\star}$ is a zero vector, Phan et al. (2019).

Algorithm 1: `Alternating Error Preserving Correction`

1

    Input: Data tensor $\mathcal{Y}$: $(I_1 \times I_2 \times \cdots \times I_N)$, and a rank R and error bound δ

    Output: $\mathcal{X} = [\![\boldsymbol{\eta}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$ s.t. $\min \|\boldsymbol{\eta}\|_2^2$ s.t. $\|\mathcal{Y} - \mathcal{X}\|_F^2 \le \delta^2$

2 begin

4   Initialize $\mathcal{X} = [\![\boldsymbol{\eta}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$ such that $\|\mathcal{Y} - \mathcal{X}\|_F^2 \le \delta^2$

5   **repeat**

6   **for** $n = 1, 2, \ldots, N$ **do**

8   Compute $\mathbf{G}_n = \mathbf{Y}_{(n)} \left(\bigodot_{k \ne n} \mathbf{U}^{(k)} \right)$

10   Compute EVD of $\Gamma_{-n} = \circledast_{k \ne n}(\mathbf{U}^{(k)T}\mathbf{U}^{(k)}) = \mathbf{V}_n\Sigma_n\mathbf{V}_n^T$ and $\mathbf{F}_n = \mathbf{G}_n\mathbf{V}_n$

12   Solve an SCQP: $\min \quad \delta_n \mathbf{z}^T \Sigma^{-1} \mathbf{z}^T - 2\mathbf{c}^T\mathbf{z}$ s.t. $\mathbf{z}^T\mathbf{z} = 1$

14 $\mathbf{Z}_n = [\ldots, \frac{z_r}{\|\mathbf{f}_r^{(n)}\|}\mathbf{f}_r^{(n)}, \ldots]$

16 $\mathbf{U}_\eta^{(n)} = (\mathbf{F}_n\Sigma^{\frac{-1}{2}} - \delta_n \mathbf{Z}_n) \Sigma^{\frac{-1}{2}} \mathbf{V}_n^T$

18   Update $\boldsymbol{\eta}$ and $\mathbf{U}^{(n)}$: $\eta_r = \|\boldsymbol{u}_{\eta,r}^{(n)}\|$, $\boldsymbol{u}_r^{(n)} = \frac{\boldsymbol{u}_{\eta,r}^{(n)}}{\eta_r}$

19   **end**

20   **until** *a stopping criterion is met*

21 end

- Assume the first column of $\mathbf{F}_n$ is non-zero, i.e., $c_1 \neq 0$, hence, $z_1^\star \neq 0$, and

$$\mathbf{Z}_n^\star = \bar{\mathbf{F}}_n \, \text{diag}([\ldots, z_r^\star, \ldots]), \qquad (12)$$

where columns of $\bar{\mathbf{F}}_n$ are $\frac{\boldsymbol{f}_r^{(n)}}{\|\boldsymbol{f}_r^{(n)}\|}$ for non zero columns $\boldsymbol{f}_r^{(n)}$, and zero vectors elsewhere.

- Since $\boldsymbol{f}_1^{(n)}$ is non-zero, the minimizer $\boldsymbol{z}^\star$ of the SCQP in (12) is given in closed-form as

$$z_r^\star = \frac{c_r}{\|\boldsymbol{c}\|(s_r - \lambda)} = \frac{\|\boldsymbol{f}_r(n)\|\sigma_r^{\frac{-3}{2}}}{\|\boldsymbol{c}\|(s_r - \lambda)} \qquad (13)$$

where $s_r = 1 + \frac{\delta_n}{\|\boldsymbol{c}\|}(\sigma_r^{-1} - \sigma_1^{-1})$, and λ is a unique root in $[0, 1)$ of a secular function Gander et al. (1989); Phan et al. (2019)

$$\sum_r (z_r^\star)^2 = 1$$

▶ The new update of $\mathbf{U}_\eta^{(n)}$ can be expressed in a compact form as

$$
\begin{aligned}
\mathbf{U}_\eta^{(n)} &= (\mathbf{F}_n \Sigma^{\frac{-1}{2}} - \delta_n \bar{\mathbf{F}}_n \operatorname{diag}([\ldots, z_r^\star, \ldots])) \Sigma^{\frac{-1}{2}} \mathbf{V}_n^T \\
&= \mathbf{F}_n \operatorname{diag}\left(1 - \frac{\delta_n}{\|\mathbf{c}\| \sqrt{\sigma_r}(s_r - \lambda)}\right) \Sigma^{-1} \mathbf{V}_n^T . \qquad (14)
\end{aligned}
$$

▶ Observed that only when $\delta_n = 0$, the above update (14) boils down to the ordinary ALS update for CPD

$$
\mathbf{U}_\eta^{(n)} = \mathbf{Y}_{(n)} \mathbf{T}_n \Gamma_{-n}^{-1}.
$$

Algorithm 2: CPD with EPC

1

Input: Data tensor $\mathcal{Y}$: $(I_1 \times I_2 \times \cdots \times I_N)$, and a rank R and rank-1 norm bound ε

Output: $\mathcal{X} = [\![\boldsymbol{\eta}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$ of rank R

2 begin

4   Initialize $\mathcal{X}_{[0]} = [\![\boldsymbol{\eta}_0; \mathbf{U}_0^{(1)}, \mathbf{U}_0^{(2)}, \ldots, \mathbf{U}_0^{(N)}]\!]$

5   **repeat**

7   **if** $\|\boldsymbol{\eta}_k\|_2^2 \geq \varepsilon^2$ **then**

9   Apply EPC to find $\mathcal{X}_{[k+1]}$: min $\|\boldsymbol{\eta}_{k+1}\|_2^2$ s.t $\|\mathcal{Y} - \mathcal{X}_{[k+1]}\|_F \leq \|\mathcal{Y} - \mathcal{X}_{[k]}\|_F$

10   **else**

12   Find $\mathcal{X}_{[k+1]}$ in CPD of $\mathcal{Y}$ initialized by $\mathcal{X}_{[k]}$

13   **end**

14   **until** *a stopping criterion is met*

15 end

▶ Instead of correcting solutions with divegerging loading components, we can seek a decomposition with bounded sensitivity

$$\min \quad \|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2$$
$$\text{s.t.} \quad \mathrm{ss}(\llbracket \mathbf{A}, \mathbf{B}, \mathbf{C} \rrbracket) \leq \delta.$$
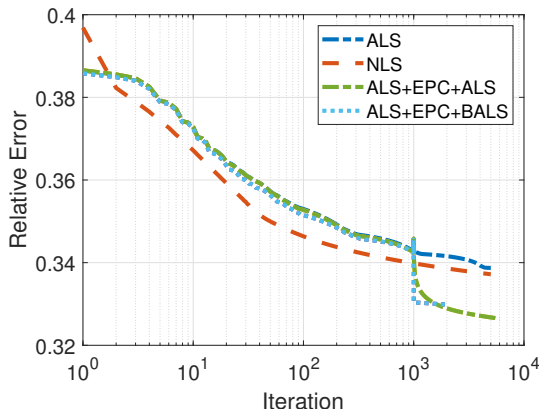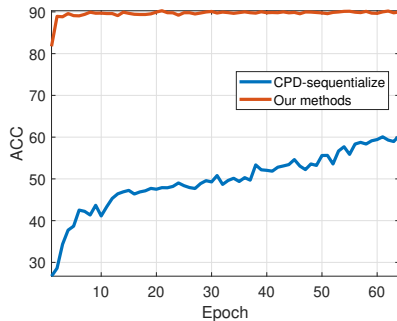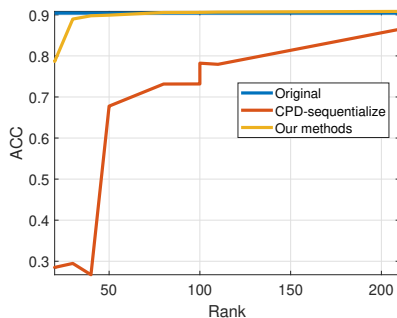
Figure: Convolution tensor at Layer-10 in AlexNet of size $(3 \times 3) \times 256 \times 384$. Both ALS and NLS converge slowly because of the high norm of rank-1 tensors. We apply EPC after 1000 iterations of ALS, then continue CPD. The algorithm attained significantly lower approximation error.

Table: Performance comparison for algorithms.

	ALS +line search	NLS	ALS+EPC +ALS	ALS+EPC +BALS
Relative Error	0.3387	0.3372	0.3254	0.3240
Running time (s)	254.2	1087.7	283.5	1080.8
Norm of Rank-one tensors	2.08×10^8	4.75×10^8	3.14×10^4	4.43×10^3

Example: Stable CPD of convolutional kernels in Alexnet I

- We used four convolutional kernels in the Alexnet. These tensors have the order 4, but were reshaped to order-3 tensors by folding their first two dimensions.
- This example demonstrates an application of the BTD with sensitivity control in compression of CNN.
- Four tensors are decomposed by BTD models with 15 blocks of the size $2 \times 2 \times 2$, so that the core tensor had the size $30 \times 30 \times 30$.

Example: Stable BTD of convolutional kernels in Alexnet CNN II

Table: Comparison of KLM-bnd and SDF-NLS algorithms on example with real-world tensors.

layer no./ size	Algorithm	Sensitivity	Relative fitting error
2	KLM-bnd	**6.5307e+06**	0.75058
$25 \times 48 \times 256$	SDF-NLS	1.8305e+12	0.75302
3	KLM-bnd	**1.7378e+08**	0.85923
$9 \times 256 \times 384$	SDF-NLS	1.0404e+12	0.85952
4	KLM-bnd	**1.3588e+08**	0.90215
$9 \times 192 \times 384$	SDF-NLS	1.055e+11	0.90224
5	KLM-bnd	**2.1769e+07**	0.85361
$9 \times 192 \times 256$	BTD-NLS	6.2318e+10	0.85397

- NLS for BTDSorber et al. (2016) and KLM with bounded sensitivity achieve approximately the same fitting error, but our algorithm provides the estimate with a much smaller sensitivity measure.
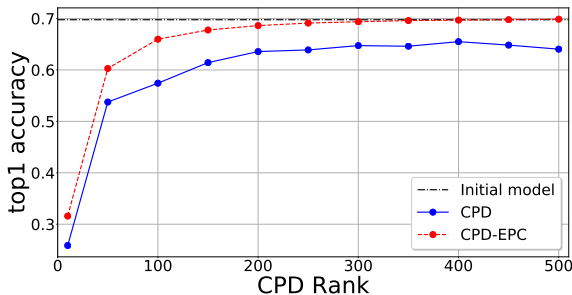
Figure: Performance evaluation of ResNet-18 on ILSVRC-12 dataset after replacing `layer4.1.conv1` by its approximation using CPD and stable CPD. The networks are fine-tuned after compression

Example: ResNet-18 with Stable Factorized-Layers II

Figure: Top-1 accuracy and sensitivity of the models estimated using CPD (blue) and stable CPD(red). Each model has a single decomposed layer with the best CP rank and was fine-tuned after compression. StableCPD outperforms CPD in terms of accuracy/sensitivity trade-off.
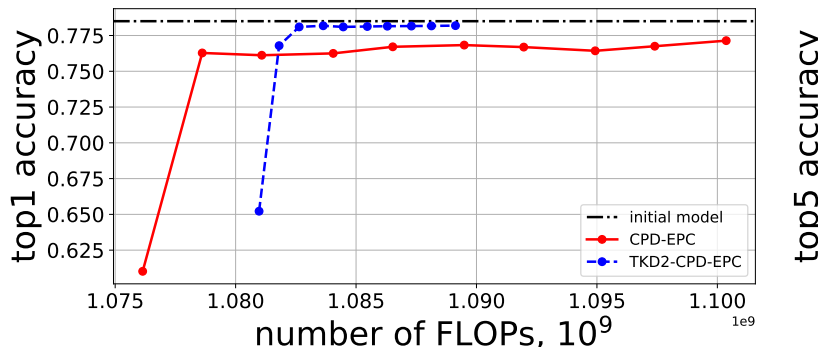
Figure: Top1 accuracy using stable CPD and stable TKD-CPD in compression of the layer `4.0.conv1` in the pre-trained ResNet-18 on ILSVRC-12 dataset. Initial model has $\approx 1.11 \times 10^9$ FLOPs.

Stable TKD-CPD-EPC shows better accuracy recovery with a relatively low number of FLOPs.

Example: Full model compression I

Table: Comparison of different model compression methods on ILSVRC-12 validation dataset. The baseline models are taken from Torchvision.

Model	Method	↓ FLOPs	Δ top-1	Δ top-5
VGG-16	Asym. **?**	≈ 5.00	-	-1.00
	TKD+VBMF **?**	4.93	-	-0.50
	Our (EPS[1]=0.005)	**5.26**	**-0.92**	**-0.34**
ResNet-18	Channel Gating **?**	1.61	-1.62	-1.03
	Channel Pruning **?**	1.89	-2.29	-1.38
	FBS **?**	1.98	-2.54	-1.46
	MUSCO **?**	2.42	-0.47	-0.30
	Our (EPS[1]=0.00325)	**3.09**	**-0.69**	**-0.15**
ResNet-50	**Our** (EPS[1]=0.0028)	**2.64**	**-1.47**	**-0.71**

[1] EPS: accuracy drop threshold. Rank of the decomposition is chosen to maintain the drop in accuracy lower than EPS.

ResNet-50 for Image Classification I

We use a pretrained model shipped with Torchvision as a baseline for ILSVRC-12.

Table: ResNet-50 models for image classification on ImageNet. Original model has 25.6M parameters and 8.2B FLOPs. Rank search over the grid with rank step 1 and 16.

Model	FLOPs	No.param.	Top-1 acc., %	Top-5 acc., %
Original	1.00×	1.00×	76.13	92.86
Compressed 1^0	2.64×	2.51×	74.66(-1.47)	92.15(-0.71)
Compressed 16^0	2.32×	2.39×	74.66(-1.47)	92.35(-0.51)

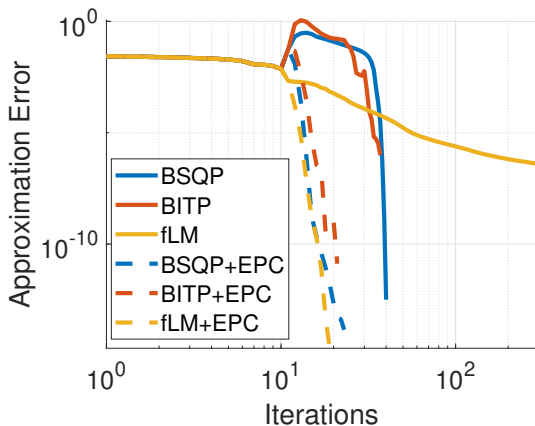Table: Inference time and acceleration for ResNet-50 on different platforms.

Platform	Model inference time	
	Original	Compressed
Intel® Xeon®Silver 4114 CPU 2.20 GHz	3.92 ± 0.02 s	2.84 ± 0.02 s
NVIDIA®Tesla®V100	102.3 ± 0.5 ms	89.5 ± 0.2 ms
Qualcomm®Snapdragon™845	221 ± 4 ms	171 ± 4 ms
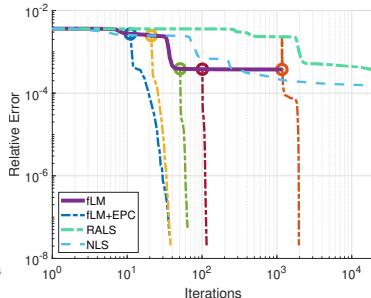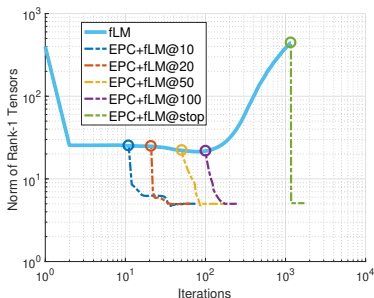
- ▶ Small tensor of size $4 \times 4 \times 4$ and rank-$R = 5$. The first four loading components were highly collinear, with $\boldsymbol{u}_r^{(n)T} \boldsymbol{u}_s^{(n)} = 0.99$ for all n and $1 \leq r \neq s \leq R - 1$.
- ▶ fLM, RALS and NLS do not converge, while the norm of rank-1 tensors increased dramatically (dashed blue curve in (left)).
- ▶ After running EPC, fLM converges quickly (dashed curves).
- ▶ EPC+fLM@10: apply EPC after 10 iterations of fLM

- Decomposition of the multiplicative tensors of size $9 \times 9 \times 9$ with rank of $R = 23$
- EPC after 10 iterations of fLM boots the convergence of CPD. EPC+fLM converges after 20 iterations

- When $\mathcal{Y}$ is corrupted by e.g., additive Gaussian noise, the constraint error bound in EPC $\delta^2 = \|\mathcal{E}\|_F^2$.

- Assuming that $\hat{\mathcal{Y}}^\star$ is with the smallest rank $R^\star$ such that $\|\mathcal{Y} - \hat{\mathcal{Y}}^\star\|_F^2 = \delta^2$.

When the estimated rank exceeds the true trank $R^\star$, the decomposition explains the noise, The higher the estimated tensor rank, the more noise the tensor $\hat{\mathcal{Y}}$ will explain.

When the noise is dominant to the data, approximation of $\mathcal{Y}$ with its true rank often yields an overestimate.
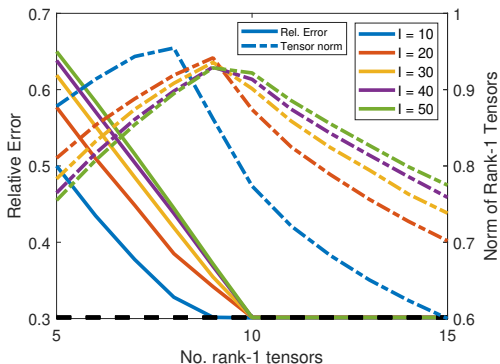
Determine numerical rank R which holds the error bound

Figure: Tensors of rank 10 corrupted by Gaussian noise at SNR = 10dB.

EPC can maintain the approximation at a target error bound, i.e., $\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 = \delta^2$ when $R \geq R^\star$, the true rank.

- ▶ When EPC does not reach the bound, the estimated rank is relatively low, $R < R^\star$
- ▶ When EPC attains the error bound, the rank of $\hat{\mathcal{Y}}$ may not be minimal. Try a smaller rank.

Image denoising I

- Color images $\mathcal{T}$ degraded by additive Gaussian noise at SNR = 10 dB
- Constructed tensors, $\mathcal{Y}_{r,c}$, of a size $w \times w \times 3 \times (2d+1) \times (2d+1)$

$$\mathcal{Y}_{r,c}(:,:,:,d+1+i,d+1+j) = \mathcal{T}_{r+i,c+j}$$

comprising $(2d+1)^2$ blocks, around the block $\mathcal{T}_{r,c} = \mathcal{T}(r : r+w-1, c : c+w-1, :)$, where $i, j = -d, \ldots, 0, \ldots, d$, and d represents the neighbour width.
- Each tensor $\mathcal{Y}_{r,c}$ was then approximated by EPC, where $\delta^2 = 3\sigma^2 w^2 (2d+1)^2$, and σ the noise level.
- DCT spatial filtering used as a preprocessing.
- EPC with the tensor rank of 8, and adjusted it to attain the error bound.
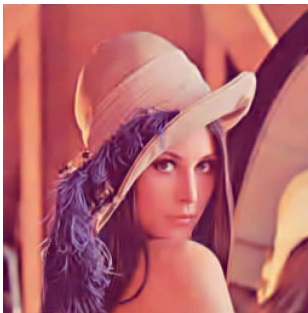
Image denoising II

Table: The performance comparison of algorithms considered in terms of PSNR (dB) and SSIM for image denoising when SNR = 10 dB.

| Algorithms	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
	Lena		Tiffany		Barbara	
EPC	**33.34**	**0.924**	**36.01**	**0.932**	**33.73**	**0.927**
TT-SVD	32.68	0.892	35.47	0.913	33.04	0.901
Tucker	32.74	0.919	35.47	0.926	32.92	0.919
BRTF	32.07	0.840	35.07	0.888	33.10	0.899
K-SVD	32.72	0.908	35.61	0.928	32.64	0.908
	Pepper		Pens		House	
EPC	**33.15**	**0.926**	**32.20**	**0.896**	**35.41**	**0.895**
TT-SVD	32.07	0.861	31.61	0.884	34.38	0.877
Tucker	32.23	0.917	31.27	0.884	34.40	0.885
BRTF	31.42	0.825	31.82	0.877	33.67	0.823
K-SVD	32.60	0.918	31.14	0.862	34.67	0.881

Image denoising III
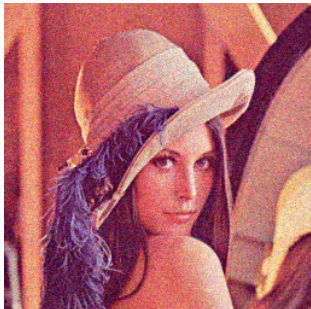
(a) Noisy image at SNR = 10 dB
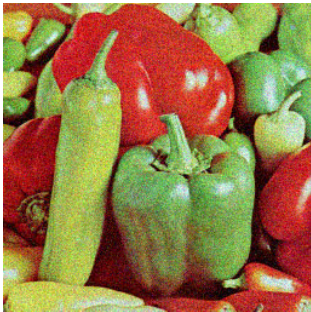
(b) EPC, SSIM = 0.924
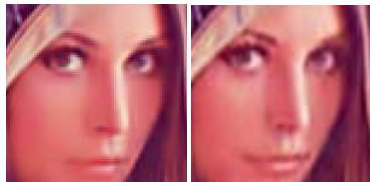
(c) From left to right, EPC(SSIM = 0.924), Tucker(0.919), TT-SVD(0.892) and K-SVD(0.908)
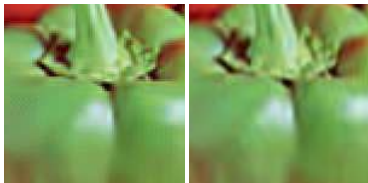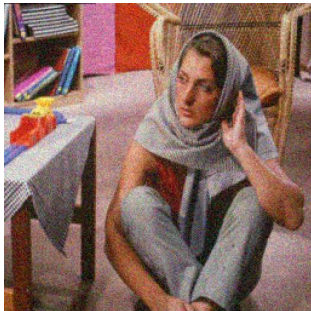
(d) Noisy image at SNR = 10 dB   (e) EPC, SSIM = 0.926

(f) From left to right, EPC(SSIM = 0.926), Tucker(0.917), TT-SVD(0.861) and K-SVD(0.918)

(g) Noisy image at SNR = 10 dB (h) EPC, SSIM = 0.927

(i) From left to right, EPC(SSIM = 0.927), Tucker(0.919), TT-SVD(0.901) and K-SVD(0.908)

Conclusions I

- ▶ Most tensor decompositions encounter diverging components (CPD, BTD, Tensor network with loop)
- ▶ Factorized layers with nonlinear ReLu layers work like a TD with nonnegativity constraints and can suffer the degeneracy as CPD or BTD.
- ▶ Stable tensor decomposition with bounded sensititivy make the decomposition solution interpretable, CNN with factorized layers easy to finetune and attain high accuracy
- ▶ Experiment results confirm superiority of the stable CPD over standard CPD in compression of CNNs.

References I

Boggs, P. T. and Tolle, J. W. (1995). Sequential quadratic programming. *Acta Numerica*, 4:1–51.

Comon, P., Luciani, X., and de Almeida, A. L. F. (2009). Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23(7-8):393–405.

De Lathauwer, L. (2008). Decompositions of a higher-order tensor in block terms – Part I and II. *SIAM Journal on Matrix Analysis and Applications (SIMAX)*, 30(3):1022–1066. Special Issue on Tensor Decompositions and Applications.

Fletcher, R. (2000). *Nonlinear Programming*, pages 229–258. John Wiley & Sons, Ltd.

Gander, W., Golub, G. H., and von Matt, U. (1989). A constrained eigenvalue problem. *Special Issue Dedicated to Alan J. Hoffman, Linear Algebra and its Applications*, 114:815 – 839.

Handschuh, S. (2015). *Numerical Methods in Tensor Networks*. PhD thesis, Facualty of Mathematics and Informatics, University Leipzig, Germany, Leipzig, Germany.

Harshman, R. and Lundy, M. (1984). *Data preprocessing and the extended Parafac model*. Praeger, New York, USA.

Harshman, R. A. (2004). The problem and nature of degenerate solutions or decompositions of 3-way arrays. In *Paper presented at the American Institute of Mathematics Tensor Decomposition Workshop*, Palo Alto, CA.

Krijnen, W., Dijkstra, T., and Stegeman, A. (2008a). On the non-existence of optimal solutions and the occurrence of "degeneracy" in the Candecomp/Parafac model. *Psychometrika*, 73:431–439.

Krijnen, W. P., Dijkstra, T. K., and Stegeman, A. (2008b). On the Non-Existence of Optimal Solutions and the Occurrence of Degeneracy in the CANDECOMP/PARAFAC Model. *Psychometrika*, 73:431–439.

Landsberg, J. M. (2012). *Tensors: Geometry and Applications*, volume 128. American Mathematical Society, Providence, RI, USA.

Lim, L.-H. and Comon, P. (2009). Nonnegative approximations of nonnegative tensors. *Journal of Chemometrics*, 23(7-8):432–441.

Lundy, M., Harshman, R., and Kruskal, J. (1989). A two-stage procedure incorporating good features of both trilinear and quadrilinear models. pages 123–130.

Mitchell, B. C. and Burdick, D. S. (1994). Slowly converging PARAFAC sequences: Swamps and two-factor degeneracies. *Journal of Chemometrics*, 8:155–168.

Navasca, C., Lathauwer, L. D., and Kindermann, S. (2008). Swamp reducing technique for tensor decomposition. In *Proc. 16th European Signal Processing Conference (EUSIPCO 2008)*.

Paatero, P. (1997). A weighted non-negative least squares algorithm for three-way PARAFAC factor analysis. *Chemometrics Intelligent Laboratory Systems*, 38(2):223–242.

Paatero, P. (2000). Construction and analysis of degenerate PARAFAC models. *J. Chemometrics*, 14(3):285–299.

Phan, A.-H., Tichavský, P., and Cichocki, A. (2017). Partitioned hierarchical alternating least squares algorithm for CP tensor decomposition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2542–2546.

Phan, A.-H., Tichavský, P., and Cichocki, A. (2019). Error preserving correction: A method for CP decomposition at a target error bound. *IEEE Transactions on Signal Processing*, 67(5):1175–1190.

Phan, A.-H., Yamagishi, M., and Cichocki, A. (2019). Quadratic programming over ellipsoids and its applications to linear regression and tensor decomposition. *Neural Computing and Applications*.

Rayens, W. and Mitchell, B. (1997). Two-factor degeneracies and a stabilization of PARAFAC. *Chemometrics Intelliggence Laboratory Systems*, 38(2):173–181.

Sorber, L., Van Barel, M., and De Lathauwer, L. (2016). Tensorlab v3.0.

Stegeman, A. (2012). Candecomp/parafac: From diverging components to a decomposition in block terms. *SIAM J. Matrix Analysis Applications*, 33(2):291–316.

Tichavský, P., Phan, A.-H., and Cichocki, A. (2016). Partitioned alternating least squares technique for canonical polyadic tensor decomposition. *IEEE Signal Processing Letters*, 23(7):993–997.

Tichavský, P., Phan, A.-H., and Cichocki, A. (2019). Sensitivity in tensor decomposition. *IEEE Signal Processing Letters*, 26(11):1653–1657.