Low-rank Embedding of Kernels in Convolutional Neural Networks under Random Shuffling

Chao Li, Zhun Sun, Jinshi Yu, Ming Hou and Qibin Zhao

RIKEN Center for Advanced Intelligence Project
School of Automation, Guangdong University of Technology, Guangzhou 510006, China
{chao.li, qibin.zhao}@riken.jp

Motivation

1. Matrix/tensor decomposition is a promising tool for weight compression.
2. Previously, the authors claim that the efficiency is due to the structral similarity in training data.

In our paper, we argue:

In CNN,
the low-rank structure
of the kernels is inherent!

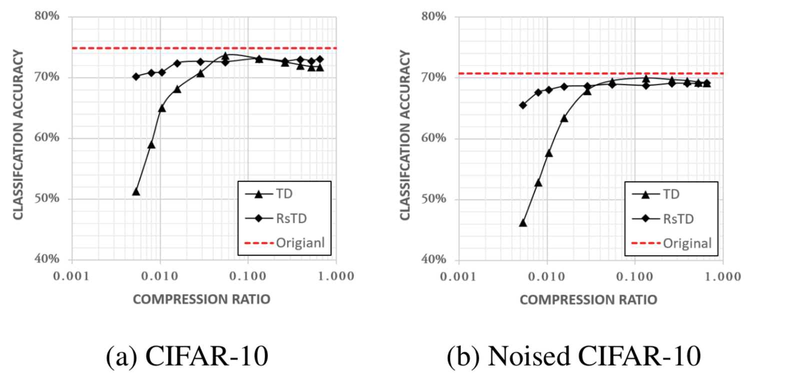
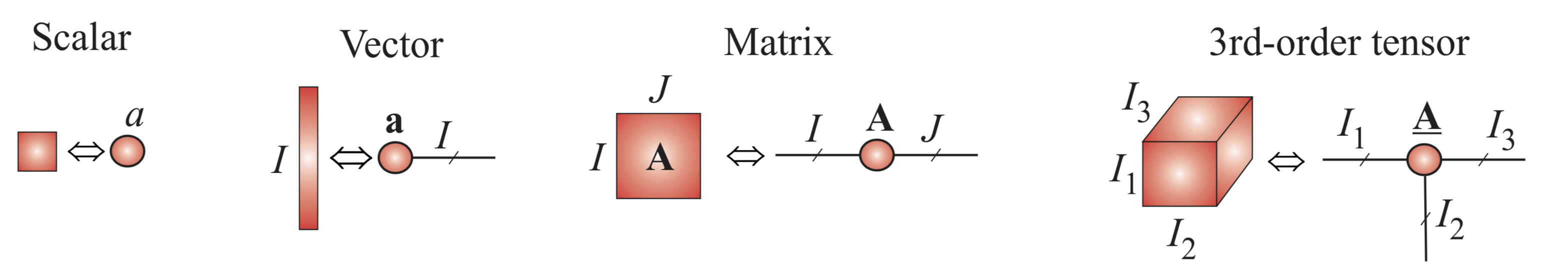


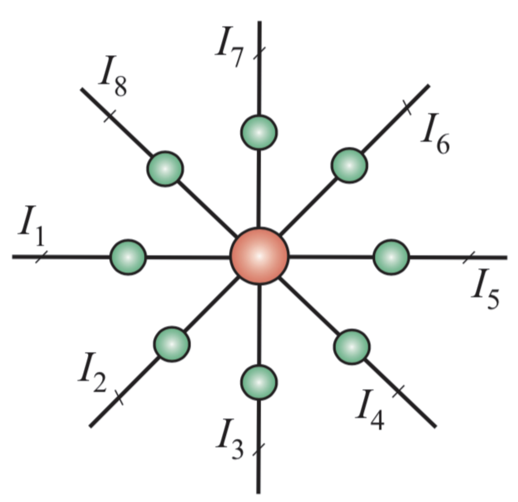
Fig. 1: Comparison of the classification accuracy of the CNNs in our experiments, where TD represents the conventional TD-based compression method (by tensor-train-matrix decomposition), RsTD denotes the proposed model in which the random shuffling operation is imposed on each kernel before TD, and the right line in the figure is the baseline by the uncompressed network.

A unified formulation for TD

Graphical representation (GR) of TDs



GR of Tucker decomp.:



TD can be describe by a graph.

$$\mathcal{X} = TD(\widehat{\mathcal{G}}; \mathbf{A})$$

Parameters

Adjacency matrix

Cichocki A, Lee N, Oseledets I, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions[J]. Foundations and Trends® in Machine Learning, 2016, 9(4-5): 249-429.

Experimental Results and Analysis

Experiment setting:

Input
conv – 3 × 3 – 256 – stride 1
conv – 3 × 3 – 256 – stride 1
conv – 3 × 3 – 256 – stride 2
conv – 3 × 3 – 256 – stride 1
conv – 3 × 3 – 256 – stride 1
conv – 3 × 3 – 256 – stride 2
conv – 3 × 3 – 256 – stride 1
global average pooling
fully connected-10
soft-max classifier

Table 1: CNN configurations. The convolution layer parameters are denoted by conv –<kernel size>–<number of output channels>–<stride option>.

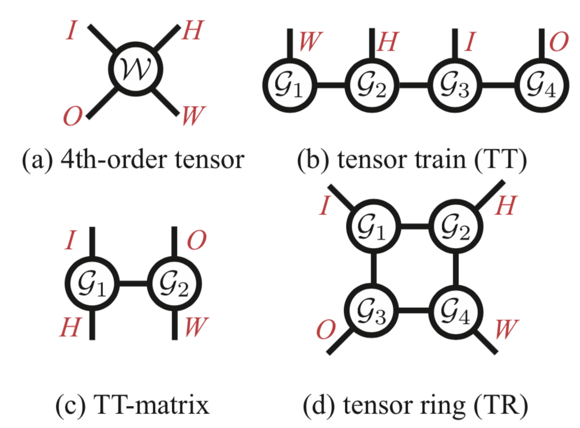
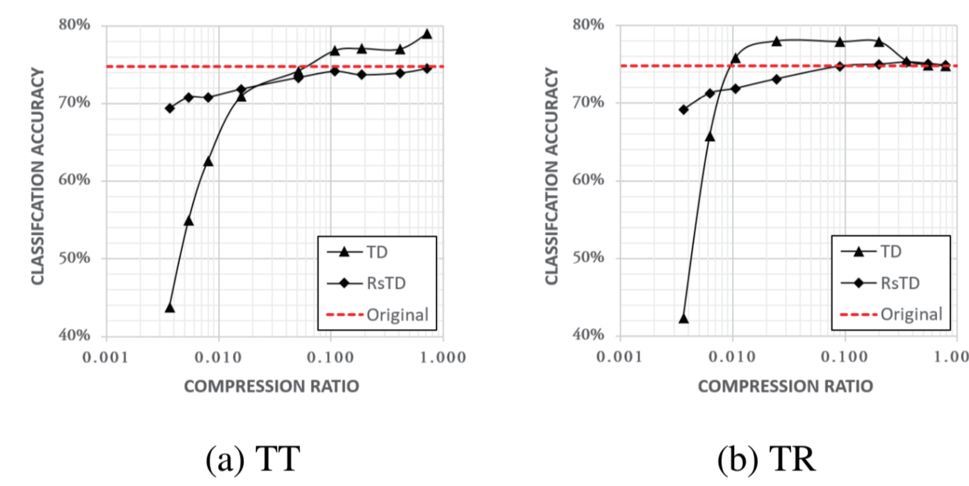


Fig. 2: Graphical representation for decomposing a kernel (4th-order tensor) by using tensor train (TT), TT-matrix and tensor ring (TR) decomposition, respectively.

Experimental results:

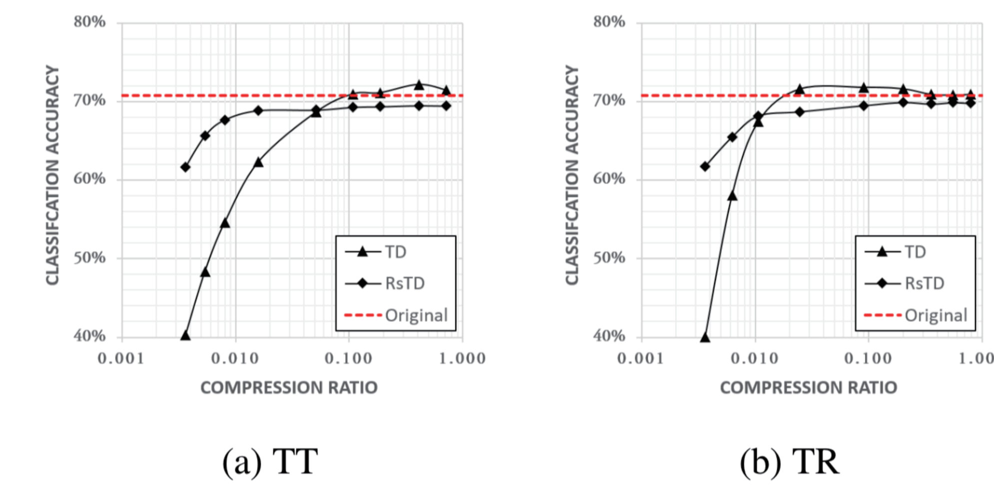
CIFAR-10



(a) TT (b) TR

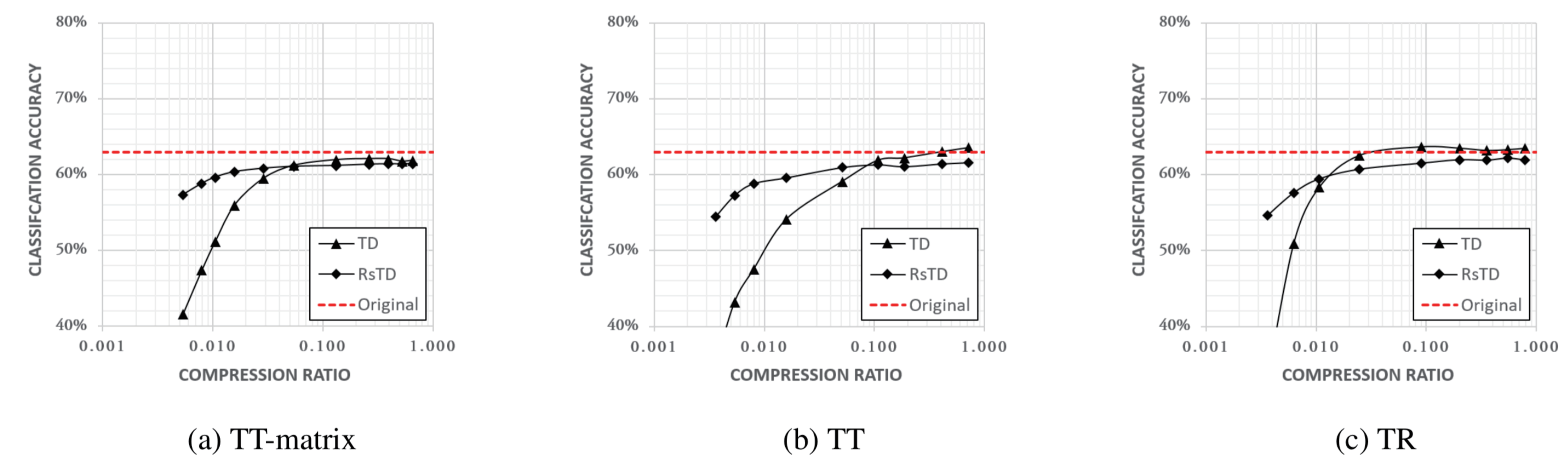
Noised CIFAR-10

$dev = 0.4$



(a) TT (b) TR

$dev = 0.8$



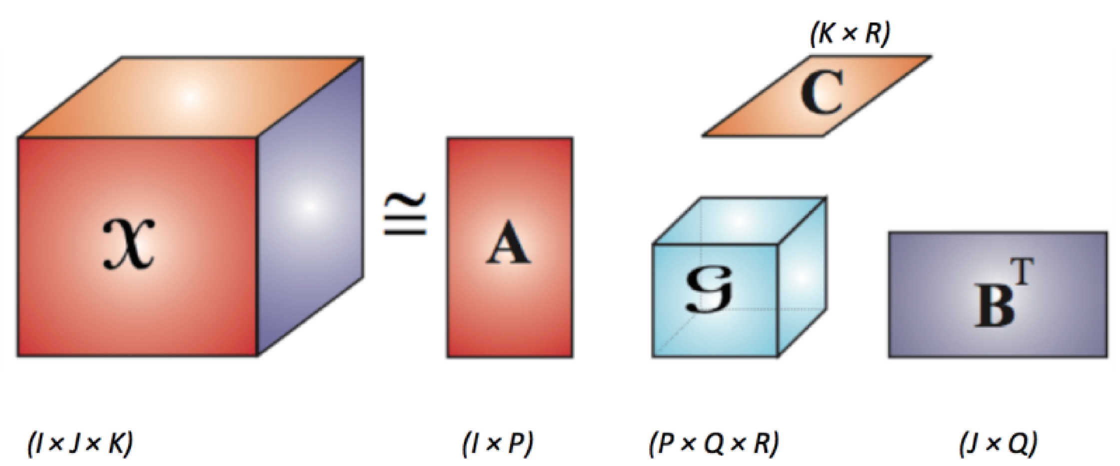
(a) TT-matrix (b) TT (c) TR

Tensor Decomposition (TD)

TD is to represent the high-dimensional problem by a low parametric form. Roughly speaking,

$$TD : \mathbb{R}^{M^D} \rightarrow \mathbb{R}^{m^d} \times \mathbb{R}^{m^d} \times \cdots \times \mathbb{R}^{m^d}$$

An example: 3rd-order Tucker decomposition



Extension of matrix multiplication

The magic of TD
comes from the tensor contraction operator

Cichocki A, Zdunek R, Phan A H, et al. Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation[M]. John Wiley & Sons, 2009.

Randomly-shuffled TD (RsTD) Layer

Random-shuffling (Rs) operator: $R : \mathbb{R}^{M^D} \rightarrow \mathbb{R}^{M^D}$

Rs is to randomly change the index for each entry.

RsTD = Rs + TD:
$$\mathcal{X} = R \cdot TD(\widehat{\mathcal{G}}; \mathbf{A})$$

RsTD Layer for CNN:
$$\mathcal{Y} = f\left(R \cdot TD(\widehat{\mathcal{G}}; \mathbf{A}) \otimes \mathcal{X} + \mathcal{B}\right)$$

Activation function

Convolutional product