A Novel Rank Selection Scheme in Tensor Ring Decomposition Based on Reinforcement Learning for Deep Neural Networks

Zhiyu (Edward) Cheng*, Baopu Li*, Yanwen Fan+, Yingze Bao*

*Baidu Research, Sunnyvale, California, USA

+Department of Computer Vision Technology (VIS), Baidu Inc, Beijing, China

The IEEE International Conference on Acoustics, Speech, and Signal Processing, May 4-8, 2020, Barcelona, Spain

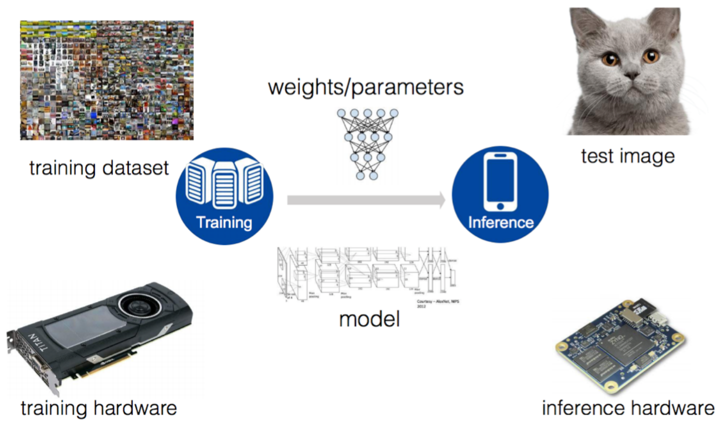
Outline

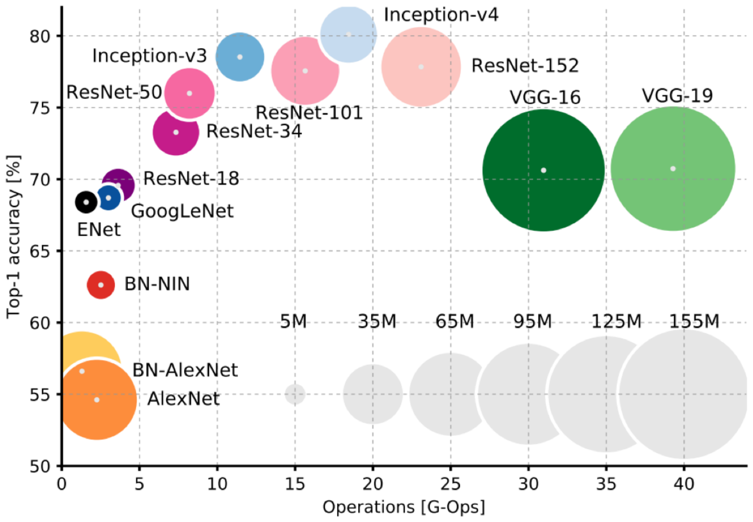
- Background & Motivation
- Tensor decomposition and its application in deep learning model compression
- Automate tensor ring rank selection via reinforcement learning
- Conclusion

Background & Motivation

- Tensor decomposition has been proved to be effective for solving many problems in signal processing and machine learning [Sidiropoulos et al, 2017].

- Modern deep learning models often contain millions of parameters and tend to be over-parameterized [Ba et al, 2014]



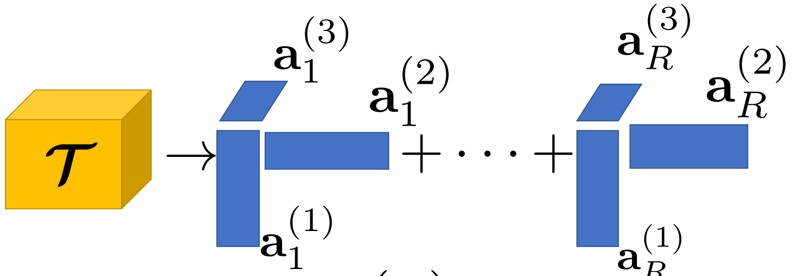


[Canziani et al 2017]

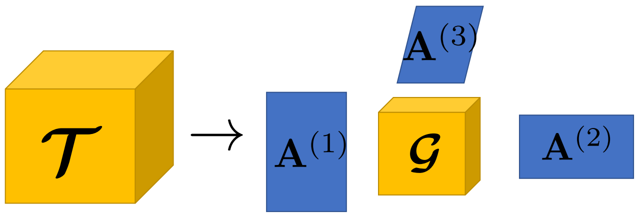
- Tensor decomposition is naturally suited for compressing deep neural networks to perform energy-efficient deep learning tasks.

Tensor Decomposition

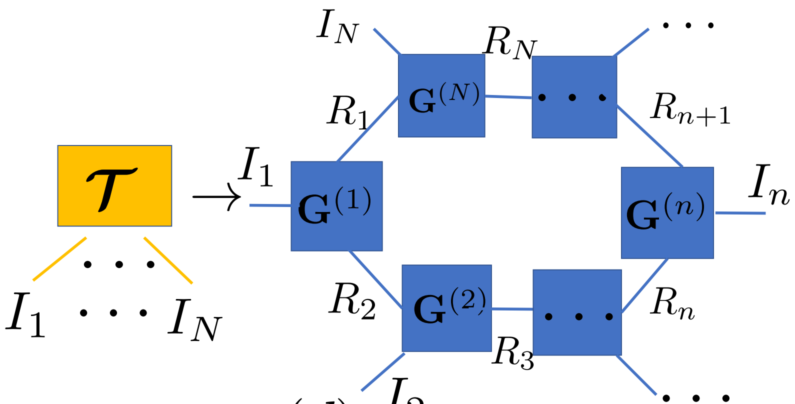
- CP decomposition [Hitchcock, 1927]
- Tucker decomposition [Tucker, 1966]
- Tensor train decomposition [Oseledets, 2011]
- Tensor ring decomposition [Zhao et al, 2016]



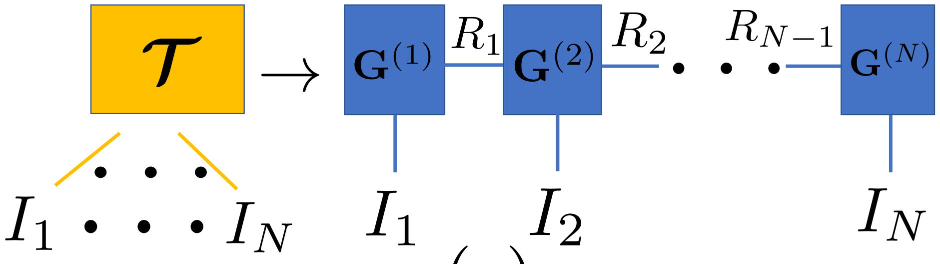
(a)



(b)



(d)



(c)

Tensor Decomposition in deep learning model compression

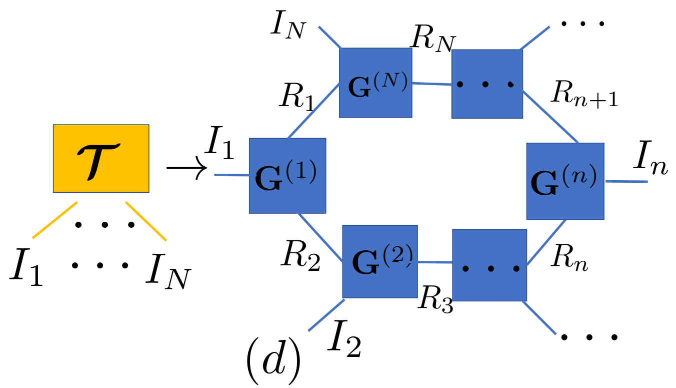
- Using CP decomposition, the 2nd convolutional layer of AlexNet was compressed, with 4x speed-up and ~1% Top5 error increase on ImageNet [Lebedev et al, 2015].
- With Tucker decomposition, various deep neural networks (AlexNet, VGG, GoogleNet) were compressed and decent reductions in model size, runtime and power, with small loss of accuracy were achieved [Kim et al, 2016].
- Tensor train decomposition were applied to compress both convolutional layers and fully connected layers in deep neural networks [Novikov et al, 2015][Garipov et al, 2016].
- ResNet-32 and Wide-ResNet-28 were compressed using tensor ring decomposition, and the results showed advantages over other forms of decomposition [Wang et al, 2018].
- An enhanced Tucker decomposition method which can adaptively adjust dimensions was proposed to compress modern deep learning models [Zhong et al, 2019].
- A tensor decomposition class specific to convolutional neural networks was characterized and analyzed [Hayashi et al, 2019].

Tensor ring decomposition

Given a tensor

$$\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$$

Decompose to circularly multiplied 3rd-order tensors

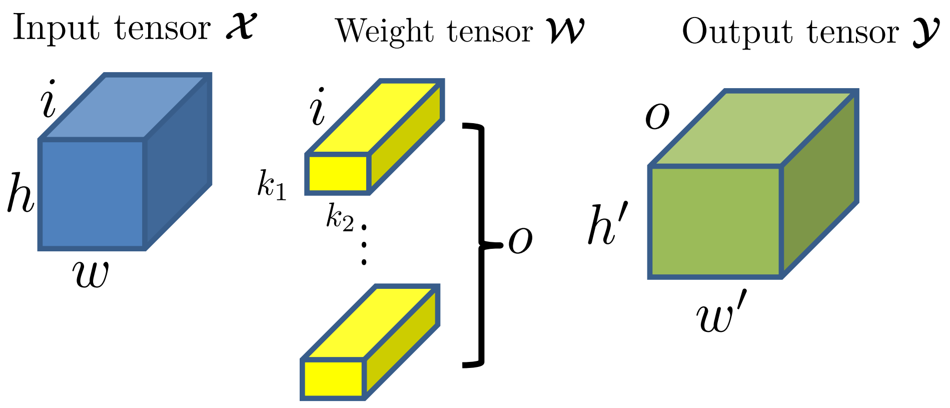


$$\mathcal{T}_{i_1,i_2,\ldots,i_N} \approx \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_N=1}^{R_N} \mathcal{G}^1_{r_1,i_1,r_2} \mathcal{G}^2_{r_2,i_2,r_3} \cdots \mathcal{G}^N_{r_N,i_N,r_{N+1}}$$
$$= \mathrm{Tr}\{\mathbf{G}^{(1)}[i_1] \cdot \mathbf{G}^{(2)}[i_2] \cdot \ \ldots \ \cdot \mathbf{G}^{(N)}[i_N]\}$$

where $\{\mathcal{G}^n\}_{n=1}^N$ is a collection of cores with $\mathcal{G}^n \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}$. Note the last tensor core is of size $R_N \times I_N \times R_1$, i.e., $R_{N+1} = R_1$, which relaxes the rank constraint of $R_{N+1} = R_1 = 1$ in tensor train decomposition. Tr denotes trace operation.

Tensor ring decomposition in convolutional layer

A typical convolutional layer in deep neural networks:



$$\mathcal{Y}_{h',w',o} = \sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}\sum_{i=1}^{I} \mathcal{W}_{k_1,k_2,i,o}\mathcal{X}_{h,w,i}$$

The convolution operation can be described by tensor ring decomposed tensors as follows:

$$\mathcal{M}_{h,w,r_2,r_3} = \sum_{i=1}^{I} \mathcal{X}_{h,w,i}\mathcal{G}^{(2)}_{r_2,i,r_3}$$

$$\mathcal{N}_{h',w',r_3,r_1} = \sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}\sum_{r_2}^{R} \mathcal{M}_{h,w,r_2,r_3}\mathcal{G}^{(1)}_{r_1,k_1,k_2,r_2}$$

$$\mathcal{Y}_{h',w',o} = \sum_{r_1}^{R}\sum_{r_3}^{R} \mathcal{N}_{h',w',r_3,r_1}\mathcal{G}^{(3)}_{r_3,o,r_1}$$

$\mathcal{M}$, $\mathcal{N}$ are Intermediate tensors.
Assume all tensor cores $\mathcal{G}$ have the same tensor ring rank R

Instead of having $\prod_{i=1}^{N} d_i$ parameters, with tensor ring decomposition we only have $\sum_{i=1}^{N} d_i R^2$ parameters.
Note d_i is one of the N factors used to factorize the weight tensor.

The tensor ring rank R therefore controls the trade-off between the model size and the model accuracy.

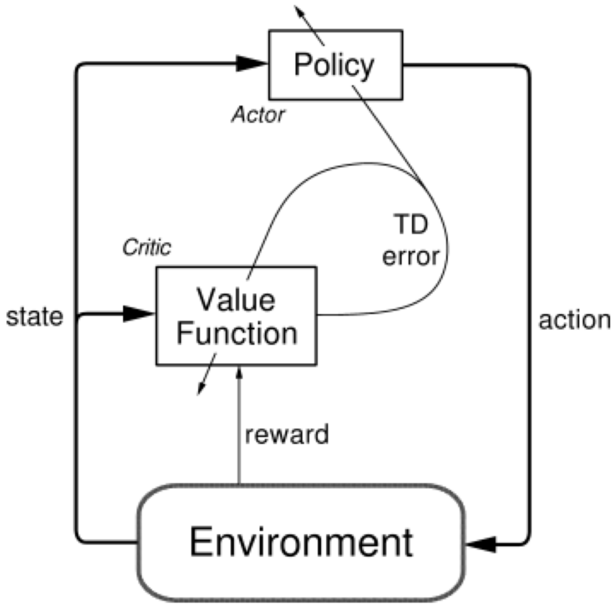
How to select the tensor ring rank R ?

Tensor ring rank selection

Existing works manually select ranks via heuristics to compress deep neural networks:

- Require long engineering hours to fine-tune the ranks for each layer of the deep neural networks.
- May not achieve satisfied compression ratio and accuracy tradeoff.

Ours: automatically select ranks via reinforcement learning



The actor-critic architecture [Sutton & Barto, "Reinforcement Learning: An Introduction", 1998]

Deep deterministic policy gradient (DDPG) [Lillicrap et al, 2015]:

- Off-policy
- Actor-critic
- Continuous action space

DDPG was used to learn pruning ratio for compressing deep neural networks [He et al, 2018].

Contributions

- We proposed a reinforcement learning based rank selection scheme for tensor ring decomposition to compress all convolutional layers in deep neural networks.

- We applied deep deterministic policy gradient (DDPG) for continuous control of the tensor ring rank, and designed state space and action space.

- Experimental results using standard benchmark datasets validated the proposed scheme, which achieved decent improvement over hand-crafted rank selection heuristics, i.e., learned ranks are better.

Tensor ring rank selection via Reinforcement Learning

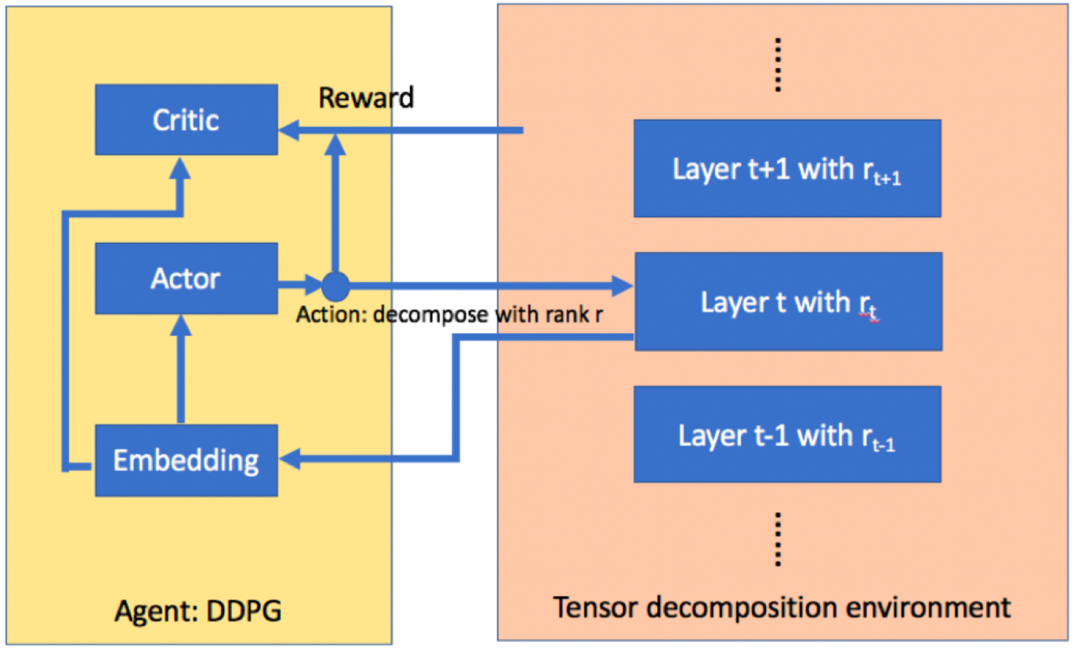
Embeddings:

layer index, input tensor dimension, weight tensor dimension, stride size, kernel size, parameter size, action of the previous layer.

$$\{i, n, c, h, w, s, k, params(i), a_{i-1}\}$$

Reward function:

$$\text{reward} = \text{accuracy}/(S_{\text{model}}^{[\mathbf{r}]} * \alpha)$$

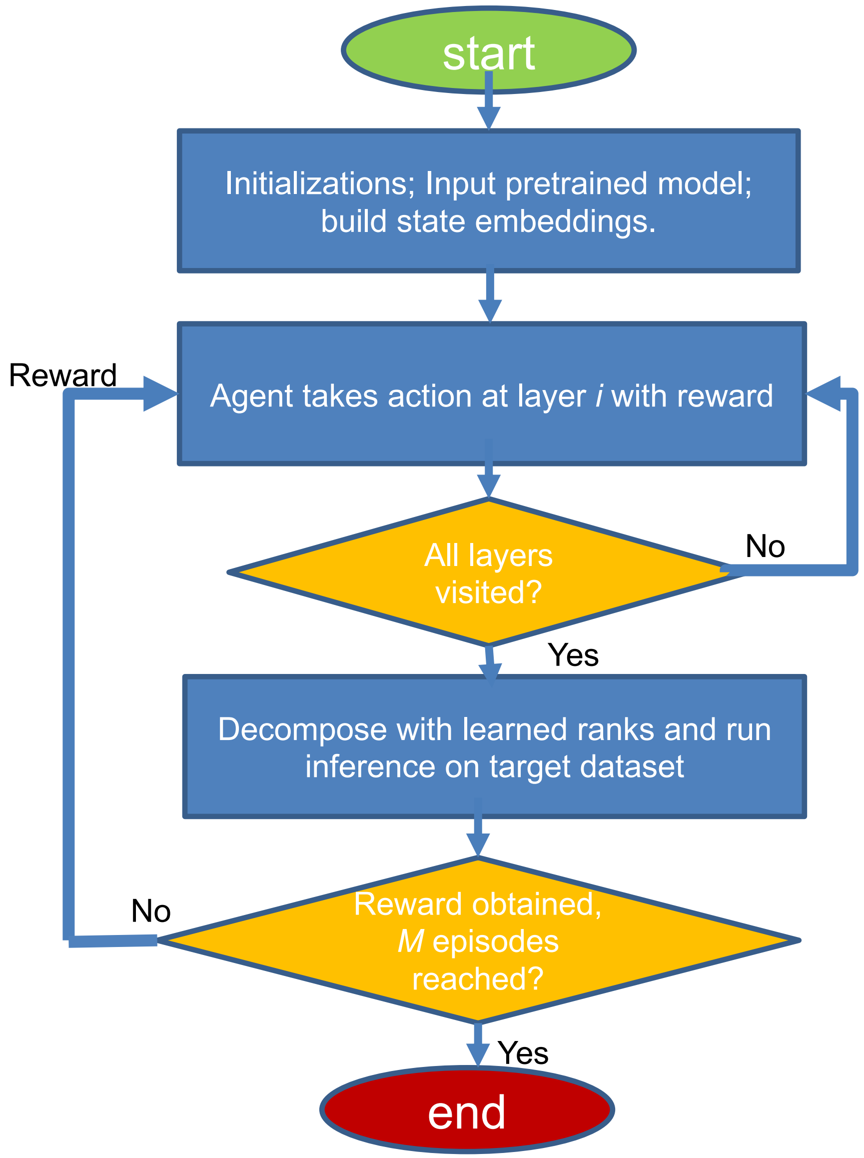


Overview of the rank selection scheme based on reinforcement learning for tensor ring decomposition in deep neural networks.

$S_{\text{model}}^{[\mathbf{r}]}$ denotes model size with tensor decomposition ranks $[\mathbf{r}]$

α controls the balance of the incentives provided by higher accuracy and smaller model size.

Rank search procedure



Algorithm 1 TR rank search based on DDPG

Require: Pretrained model with N convolutional layers
Ensure: Build state embedding such that each element is normalized within [0,1], which are inputs to the agent.
Ensure:
  Initialize action for each layer with a preset value
  repeat
    Visit each of N convolutional layers, and agent takes an action with rank r.
    If all N layers are visited, decompose each layer with learned rank and run model inference on target dataset, and get a reward which is a function of inference accuracy and compressed model size.
    Agent takes the reward and starts new search.
  until preset M episodes reached
  Output learned ranks with the best reward.

Experimental results

Image classification on two benchmark datasets:

Datasets	# images	# classes	# images/class	resolution
CIFAR-10	60,000	10	6,000	32x32x3
CIFAR-100	60,000	100	600	32x32x3

Deep neural networks: ResNet-20, ResNet-32 [He et al, 2015]

Table 1: Tensor ring decomposition on ResNet20

Method	Params	CR	Error (%)
CIFAR10			
Original	0.27M	1	9.6
TRN(ranks=6)[Wang et al, 2018]	0.02M	14x	16.9
TRN(ranks=10)[Wang et al, 2018]	0.05M	5x	12.5
Ours(learned ranks)	0.02M	14x	13.3
Ours(learned ranks)	0.04M	6x	11.7
CIFAR100			
Original	0.28M	1	34.6
TRN(ranks=8)[Wang et al, 2018]	0.03M	8x	41.6
Ours(learned ranks)	0.03M	8x	38.7

Table 2: Tensor decompositions on ResNet32

Method	Params	CR	Error (%)
CIFAR10			
Original	0.46M	1	7.5
Tucker[Kim et al, 2015]	0.09M	5x	12.3
TT(ranks=13)[Garipov et al, 2016]	0.1M	5x	11.7
TRN(ranks=6)[Wang et al, 2018]	0.03M	15x	19.2
Ours(learned ranks)	0.03M	15x	11.9
CIFAR100			
Original	0.47M	1	31.9
Tucker[Kim et al, 2015]	0.09M	5x	42.2
TT(ranks=13)[Garipov et al, 2016]	0.1M	5x	37.1
TRN(ranks=6)[Wang et al, 2018]	0.04M	12x	36.6
Ours(learned ranks)	0.04M	12x	35.5

As expected, our results with learned ranks outperform existing works, i.e., we are able to achieve either higher compression ratio (CR) or lower error rate, or both in some cases.

Conclusion

- We proposed a novel rank selection scheme in tensor ring decomposition based on reinforcement learning to compress deep neural networks.

- We applied deep deterministic policy gradient (DDPG) for continuous control of the tensor ring rank, and designed state space and action space.

- Experimental results using standard benchmark datasets validated the proposed scheme: learned ranks are better than hand-crafted ranks.

- Usability and future directions:
  - Apply the proposed framework to other forms of tensor decomposition.
  - Study tensor decomposition with learned ranks on other applications such as video understanding, natural language processing.
  - Hardware related: when implementing on hardware, considering resources limitation, learn the ranks to achieve better compute and memory balance.

Thank you!

Zhiyu (Edward) Cheng, Ph.D.
zhiyucheng@baidu.com