Learning Low-Rank Tensor Cores with Probabilistic ℓ_0-Regularized Rank Selection for Model Compression

Tianxiao Cao[1] · Lu Sun[1] · Canh Hao Nguyen[2] · Hiroshi Mamitsuka[2]

[1]ShanghaiTech University, China
[2]Kyoto University, Japan

上海科技大学
ShanghaiTech University

京都大学
KYOTO UNIVERSITY

Background

- **Model compression:** To reduce the number of parameters of deep neural networks (DNNs) while keeping the same function and comparable performance.
- **Tensor decomposition (TD) for model compression:** To replace the large weight parameter tensor with smaller tensors that contract to the large tensor. Contraction is analog to matrix multiplication which eliminates the joint dimension of two tensors. Tensor diagram: A vertex is a tensor. An edge is a dimension. A connection between edges is a contraction.

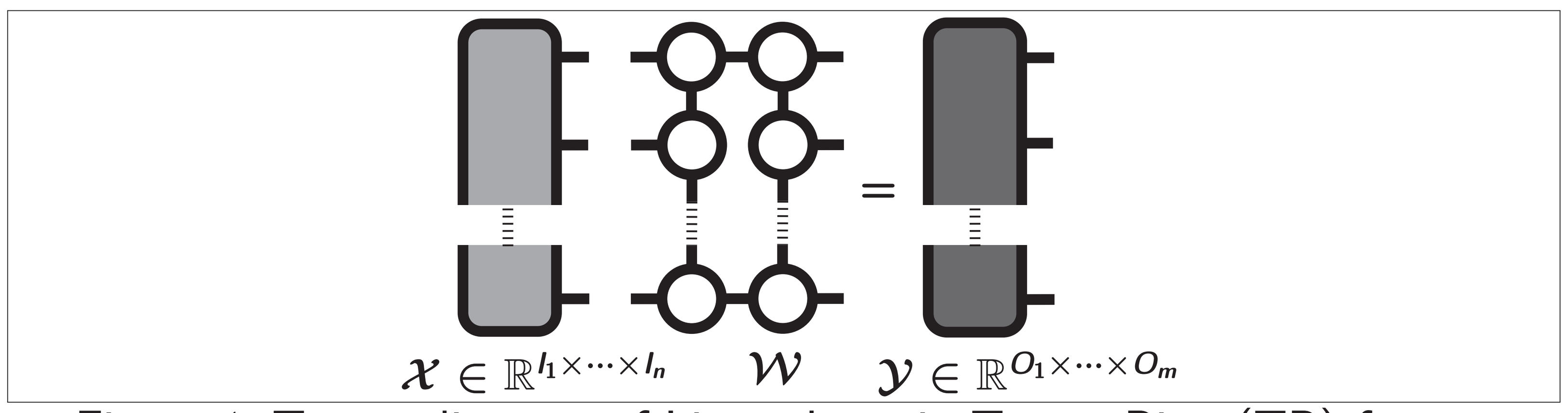

Figure 1. Tensor diagram of Linear layer in Tensor-Ring (TR) format

- **Ranks** (in TR, the intermediate dimensions, e.g. the dimensions on the rings) decide the trade-off between parameter number and expressive power, while many prototype compression methods with TD determine them uniformly as one hyper-parameter, which is sub-optimal![4]

Motivation

- **Rank selection:** To select the proper ranks of each compressed layer for better compression-accuracy trade-off.
- **Challenges:** The rank search space size grows exponentially with the increase of small tensors! Sometimes there are many small tensors (e.g. TR). Existing rank-selection-based methods may require large extra costs (e.g. re-training [1, 2], RL agents [1], genetic algorithms [2], designing Bayesian network [3]).
- **Goal of this paper:** To learn the ranks and model parameters jointly through gradient methods!

Method

We insert a diagonal matrix between each pair of tensor cores, and each zero in this diagonal matrix means that: in each of the two cores, one of its slices can be zeroed out and pruned, and rank decreases. Fig. 2 shows the modification.

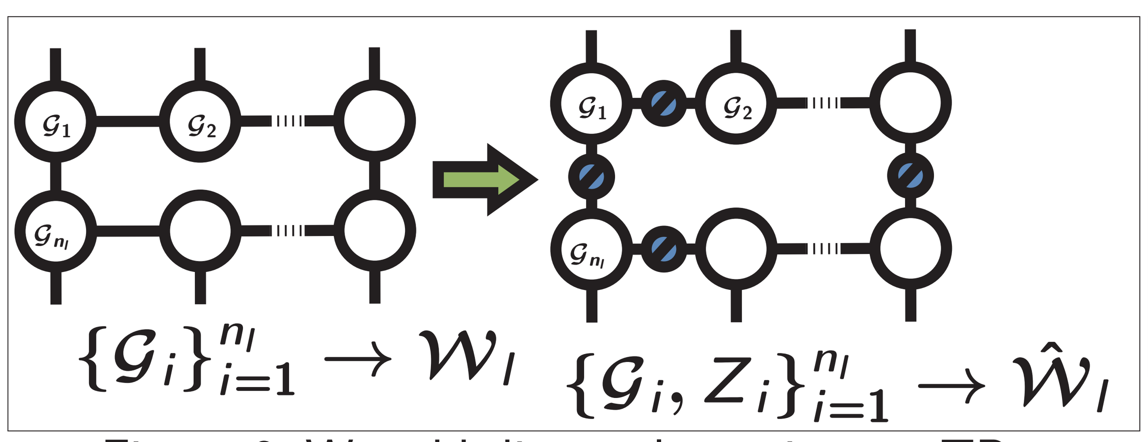

Figure 2. We add diagonal matrices to TDs.

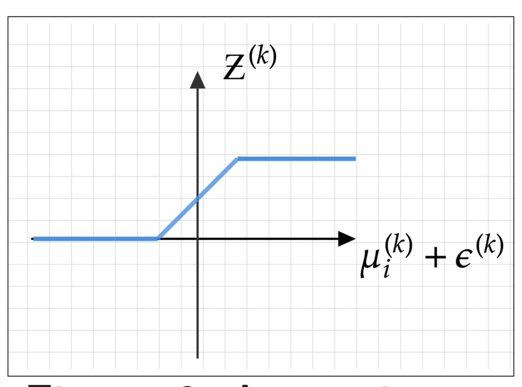

Figure 3. Approximate Bernoulli gate

We hope the diagonal Z to have Bernoulli distribution, indicating the availability of corresponding ranks. To induce rank selection, we aim to optimize weights and Bernoulli's:

$$\min_{\theta'} \quad \mathbb{E}_Z \mathbb{E}_{X,Y}\left[\mathcal{L}(f_{\theta'}(X), Y) + \lambda \sum_{l=1}^{L}\sum_{i=1}^{n_l} \|Z_i^{(l)}\|_0\right]. \qquad (1)$$

To solve using gradient methods, we adopt an Approximate Bernoulli gate on each entry of Z, shown in Fig. 3. σ is zero-centered Gaussian during training and set as zero during inference. Then ℓ_0 becomes Gaussian CDF in training and the objective is optimized with its Monte Carlo estimate:

$$\min_{\theta'} \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{K}\sum_{k=1}^{K}\mathcal{L}(f_{\{\mathcal{G}_i, Z_i^{[m]}\}_{i=1}^{n}}(x_k), y_k)\right) + \lambda\sum_{i=1}^{n}\sum_{j=1}^{R_i}\Phi\left(\frac{\mu_{i,j}}{\sigma}\right). \qquad (2)$$

Experimental Results

Method	Accuracy (%)	Compression
LeNet-5	**99.37**±0.06	1×
TRN ($r = 10$)	99.03±0.15	26×
TRN ($r = 15$)	99.16±0.06	11.71×
TRN ($r = 20$)	99.20±0.08	6.62×
Tucker [2018]	99.15	2×
MPO (TT) [2020]	99.17	20×
PSTRN [2021]	**99.4**	16.5×
MARS [2023]	99.0	10±0.8×
Ours (TR)	99.19±0.06	**61.39**±2.84×
	99.37±0.03	20.53±0.81×

Figure 4. LeNet-5 on MNIST

Method	Accuracy (%)	Compression
ResNet-32	**92.47**±0.17	1×
TRN ($r = 15$)	91.33±0.05	2.28×
TRN ($r = 10$)	90.45±0.10	4.95×
TT ($r = 13$) [2018]	88.3	4.8×
Tucker [2020]	87.7	5×
TR-RL [2020]	88.1	**15**×
PSTRN-M [2021]	90.6	5.1×
Ours (TR)	90.93±0.05	5×

Figure 5. ResNet-32 on CIFAR10

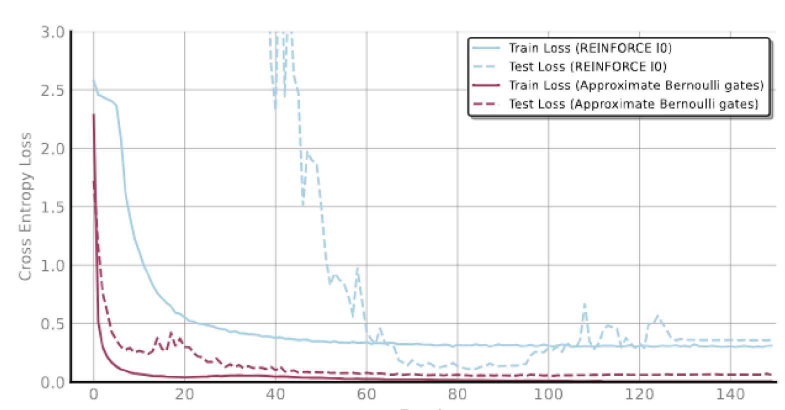

Figure 6. Loss of Approximate Bernoulli gate vs direct optimizing ℓ_0

Method	Accuracy	Comp. (Emb.)	Param.
Original Emb.	37.4%	1×	5.20M
TT ($r = 16$)	41.1%	182×	0.82M
MARS [2023]	**42.4%**	340×	0.81M
Ours (TT)	42.2%	**449**×	**0.80M**

Figure 7. Emb in LSTM on SST

Method	SacreBLEU	Comp. (Emb.)	Params.
Original Emb.	**32.36**	1×	65.13M
TR ($r = 32$)	32.14	21.33×	49.14M
TR ($r = 27$)	31.76	29.97×	48.91M
Ours (TR)	31.94	**30.78**×	**48.90M**

Figure 8. Emb in Transformer on IWSLT'14 De-en

The proposed method is:

- Better performance than uniform rank baselines
- Comparable to other rank selection methods but costs less in the search process
- More stable compared to directly optimizing ℓ_0

Bibliography

[1] CHENG, Zhiyu, et al. A novel rank selection scheme in tensor ring decomposition based on reinforcement learning for deep neural networks. In: *ICASSP 2020.*

[2] LI, Nannan, et al. Heuristic rank selection with progressively searching tensor ring network. *Complex Intelligent Systems*, 2021, 1-15.

[3] HAWKINS, Cole; ZHANG, Zheng. Bayesian tensorized neural networks with automatic rank selection. *Neurocomputing*, 2021, 453: 172-180.

[4] KODRYAN, Maxim; KROPOTOV, Dmitry; VETROV, Dmitry. Mars: Masked automatic ranks selection in tensor decompositions. In: *AISTATS 2023.*