RankSearch: An Automatic Rank Search towards Optimal Tensor Compression for Video LSTM Networks on Edge

Changhai Man*, Cheng Chang†, Chenchen Ding‡, Ao Shen‡, Hongwei Ren‡, Ziyi Guan§,
Yuan Cheng¶, Shaobo Luo‡, Rumin Zhang‡, Ngai Wong§, Hao Yu‡
*Georgia Institute of Technology †University of California, Los Angeles
‡Southern University of Science and Technology §The University of Hong Kong ¶Shanghai Jiao Tong University

***Abstract*—Various industrial and domestic applications call for optimized lightweight video LSTM network models on edge. The recent tensor-train method can transform space-time features into tensors, which can be further decomposed into low-rank network models for lightweight video analysis on edge. The rank selection of tensor is however manually performed with no optimization. This paper formulates a rank search algorithm to automatically decide tensor ranks with consideration of the trade-off between network accuracy and complexity. A fast rank search method, called RankSearch, is developed to find optimized low-rank video LSTM network models on edge. Results from experiments show that RankSearch achieves a $4.84\times$ reduction in model complexity, and $1.96\times$ speed-up in run time while delivering a 3.86% accuracy improvement compared with the manual-ranked models.**

***Index Terms*—Tensor Rank Search, Complexity Aware Optimization, Tensorized Video LSTM, Edge Computing**

I. Introduction

It is still a grand challenge for real-time video analysis on edge devices. LSTM [1], [2] networks have been proposed and proved for extracting sequence-to-sequence information from videos. However, they are usually too expensive for edge deployment. Tensor decomposition such as tensor-train [3]–[6] has been applied to construct space-time features into tensors, followed by a low-rank approximation, which deeply compresses and accelerates video LSTM networks on edge devices. As the compression is based on an approximation which may degrade the accuracy, it is an optimization problem to select the optimal ranks in tensor decomposition to balance the accuracy and compression.

In this paper, we propose an in-training automatic rank search method, called RankSearch, which aims at lightweight tensorized video LSTM networks on edge devices with optimal ranks. Compared with manual [3]–[6] or post-training rank search methods [7]–[9], RankSearch formulated the rank search problem as an optimization, then transforms the discrete rank search into continuous space with search unit, finally integrate the rank search into the model's training process for the reduced searching overhead.

Results from experiments show that: by using RankSearch, the accuracy of the video LSTM model is improved from 79.85% to 83.71% (3.86% improvement) compared with the traditional manual methods, while the time and model complexity is reduced by $1.96\times$ and $4.84\times$, respectively.

II. RankSearch Problem Formulation

In the rank search problem, it has two goals: 1) High Accuracy, and 2) Low Complexity. The cost model consists of two parts L_{acc}, L_{comp} respectively, as $L = L_{\text{acc}} + \gamma L_{\text{comp}}$ where γ is the coefficient adjusting the preference of accuracy/complexity.

We employ the same cost function for the accuracy cost model as regular model training. Briefly, with decomposition caused error $\epsilon_{y,TT} = \epsilon_{W,TT}\mathbf{x}$, we can calculated the gradient as

$$\frac{\partial L_{\text{acc}}}{\partial \epsilon_{W,TT}} = \frac{2}{MN}(\epsilon_{W,TT} + \epsilon_{W,net})\mathbf{x}\mathbf{x}^T \tag{1}$$

where $\epsilon_{W,net}$ is the error caused by the un-decomposed network itself. Then we can prove the magnitude of $\epsilon_{W,TT}$ in terms of L2-norm will be smoothly reduced after the iteration of gradient descends with learning rate l, as follows:

$$||\epsilon_{W,TT}||^2 \approx ||\epsilon^{[0]}_{W,TT}(\mathbf{I} - \frac{2l}{MN}\mathbf{x}\mathbf{x}^T)||^2 < ||\epsilon^{[0]}_{W,TT}||^2 \tag{2}$$

For the complexity cost model, we can model the computational and storage complexity as follows:

$$\begin{aligned} \#\text{PARAMs} &= \sum_{k=1}^{d} m_k n_k r_k r_{k+1} \\ \#\text{FLOPs} &= \sum_{k=1}^{d}((\prod_{i=1}^{k} m_i)(\prod_{i=k}^{d} n_i r_i) r_{k+1}) \end{aligned} \tag{3}$$

with adjusting coefficient β, then we can complete the complexity cost as follows:

$$L_{\text{comp}} = \beta \cdot \#\text{PARAMs} + (1-\beta) \cdot \#\text{FLOPs} \tag{4}$$

With the cost model for accuracy and complexity, we can define the problem as a minimization, then combines it with the model training, and finally formulate the online search as:

$$\min_{\mathbf{W},\mathbf{r}} \; L = L_{\text{acc}} + \gamma L_{\text{comp}} \tag{5}$$

III. RankSearch Problem Solution

To set the context for rank search, we would like briefly introduce how low-rank approximation works first. For any decomposed tensor $\mathcal{W}(l_1, \cdots, l_d) = \prod_d \mathcal{G}_i(l_i)$, each tensor core $\mathcal{G}_i \in \mathbb{R}^{r_i, l_1, r_{i+1}}$ can be represented as $\mathcal{G} = \mathcal{U} \times_3 \mathbf{\Sigma}$, the outer product of left singular bases $\mathcal{U}$ and corresponding singular values $\mathbf{\Sigma}$ on the third dimension (meanwhile, the right singular bases will be further decomposed in a different dimension for next tensor core). With low-rank approximation, we can select a threshold value σ_τ, where the tail singular values $\sigma_i < \sigma_\tau$ are rounded to zeros, like follows:

$$\hat{\mathbf{\Sigma}} = [\underbrace{\sigma_1, \sigma_2, \cdots, \sigma_{\hat{r}}}_{\hat{r}}, 0, \cdots, 0] \approx \mathbf{\Sigma} \tag{6}$$

In one word, the rank of the approximated tensor is related to the distribution of singular values σ_i under the same threshold. Thus, we can apply the magic of rank search by adding a

978-3-9819263-7-8/DATE23/© 2023 EDAA

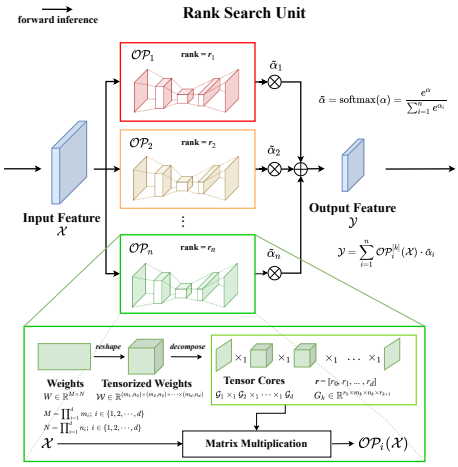

Fig. 1. Rank Search Unit

TABLE I
COMPARISON BETWEEN AUTO-RANKED AND MANUAL-RANKED MODELS

Model	Rank Method	Acc.	FLOPs[1]	PARAMs[1]	Rank
UCF11-LSTM1	Auto	83.71%	3.07M	3,864	(3, 8)
UCF11-LSTM2	Auto	81.28%	1.57M	2,640	(2, 4)
UCF11-LSTM1	Manual	79.85%	6.02M	18,701	(6, 6)
UCF11-LSTM2	Manual	77.37%	4.78M	10,280	(4, 4)
UCF11-LSTM3	Manual	61.34%	1.39M	1,808	(2, 2)

*Only shows part of results.

learnable coefficient p and slightly "re-curve" the singular values, as follows, and train p in the optimization,

$$\hat{\mathbf{\Sigma}}_{\text{wt}} = [\underbrace{p_1\sigma_1, p_2\sigma_2, \cdots, p_{\hat{r}}\sigma_{\hat{r}}}_{\hat{r}}, 0, \cdots, 0] \tag{7}$$

The selection of p is constrained by multiple rules: 1) the adjusting should not be strong, thus $p \approx 1$. 2) the adjusted singular values should be steeper, thus $p_i \leq p_{i+1}$. To ensure these constraints, we introduce search units and the equivalent singular values are as follows, where $\mathbf{\Sigma}_r$ is approximated with rank r and $\alpha_k = \sum_{i=1}^{k} p_i$

$$\mathbf{\Sigma}_{equiv} = \alpha_1\mathbf{\Sigma}_{r_1} + \alpha_2\mathbf{\Sigma}_{r_2} + \cdots + \alpha_n\mathbf{\Sigma}_{r_n} \tag{8}$$

To implement the RankSearch algorithm, here we propose a search unit, as shown in Fig. 1. Inside each search unit, there are multiple branches denoted as $\mathcal{OP}_i$ with a normalized branch weight α_k, and with rank r_k. The output of the search units equals the weighted sum of each branch's output, as follows,

$$\mathcal{Y} = \sum_{i=1}^{n} \mathcal{OP}_i(\mathcal{X}) \cdot \alpha_i \tag{9}$$

IV. EXPERIMENTS

To evaluate our proposed RankSearch framework, we evaluate our framework on CIFAR-10 [10], UCF-11 [11] and UCF-101 [12]. We first compared our searched model with manually ranked models in Table I, and showed a great advantage in both accuracy and complexity. In Table II, we compared our searched model with other SoTA models, which also shows great success in compressing the model with little accuracy loss.

V. CONCLUSION

In this paper, a rank search method, called RankSearch, is proposed to automatically select tensor ranks for video LSTM networks. By constructing a continuous optimization and online search strategy, RankSearch delivers performance as well as accuracy. Experiment results show that RankSearch achieves $4.84\times$ reduction in model size, $1.96\times$ speed-up in run time, and 3.86% accuracy improvement compared with the manual search-based methods.

TABLE II
COMPARISON BETWEEN STATE-OF-THE-ARTS AND RANKSEARCH OPTIMIZED MODELS

Approach	#Parameters	Accuracy
UCF-11		
Chen et al. [13]	47 M	85.21%
Yang et al. [3]	**3,360**	81.30%
Ours best TT-LSTM	**3,864**	83.71%
Ours best CNN+TT-LSTM	**4.21 M**	**95.94%**
UCF-101		
Zhou et al. [14]	57.32M	58.70%
Luo et al. [15]	177.88M	**80.30%**
Ours best CNN+TT-LSTM	**4.23M**	62.31%
Ours best 3DCNN+TT-LSTM	23.27M	**82.14%**

*Only shows part of results.

REFERENCES

[1] J. Donahue et al., "Long-term recurrent convolutional networks for visual recognition and description," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2625-2634, 2015.

[2] J. Liu, Y. Li, S. Song, J. Xing, C. Lan, and W. Zeng, "Multi-modality multi-task recurrent neural network for online action detection," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 9, pp. 2667-2682, Sept 2019.

[3] Y. Yang, D. Krompass, V. Tresp, "Tensor-Train Recurrent Neural Networks for Video Classification," 2017 International Conference on Machine Learning (ICML), pp. 3891–3900, 2017.

[4] Y. Cheng, G. Li, N. Wong, H. -B. Chen and H. Yu, "DEEPEYE: A Deeply Tensor-Compressed Neural Network for Video Comprehension on Terminal Devices," in ACM Transcations on Embedded Computing Systems, vol. 19, no. 18, pp. 1-25, May 2020.

[5] Y. Yang, H. Ren, C. Li, C. Ding and H. Yu, "An Edge-device Based Fast Fall Detection Using Spatio-temporal Optical Flow Model," 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 5067-5071, 2021.

[6] Y. Cheng, G. Huang, P. Zhen, B. Liu, H. -B. Chen, N. Wong and H. Yu, "An Anomaly Comprehension Neural Network for Surveillance Videos on Terminal Devices," 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1396-1401, 2020.

[7] M. Imaizumi, T. Maehara and K. Hayashi, "On Tensor Train Rank Minimization: Statistical Efficiency and Scalable Algorithm," 2017 Conference on Neural Information Processing Systems (NIPS), pp. 3930-3939, 2017.

[8] C. Hawkins and Z. Zhang, "Bayesian tensorized neural networks with automatic rank selection," in Neurocomputing, vol. 453, pp. 174-180, Apr. 2021.

[9] M. Yin, Y. Sui, S. Liao and B. Yuan, "Towards Efficient Tensor Decomposition-Based DNN Model Compression with Optimization Framework," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 10669-10678, doi: 10.1109/CVPR46437.2021.01053.

[10] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," Handbook of Systemic Autoimmune Diseases, 2009.

[11] J. Liu, Jiebo Luo and M. Shah, "Recognizing realistic actions from videos "in the wild"," 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1996-2003, 2009.

[12] K. Soomro, A. R. Zamir, M. Shah, "UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild," arXiv preprint, arXiv:1212.0402, 2012.

[13] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "Multi-fiber networks for video recognition," 2018 European Conference on Computer Vision, pp. 352-367, 2018.

[14] Y. Zhou, X. Sun, Z. -J. Zha and W. Zeng, "MiCT: Mixed 3D/2D Convolutional Tube for Human Action Recognition," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 449-458, 2018.

[15] Z. Luo, J. Hsieh, L. Jiang, J. Niebles, and F. Li, "Graph distillation for action detection with privileged modalities," 2018 European Conference on Computer Vision (ECCV), pp. 166-183, 2018.