

STAR: An Efficient Softmax Engine for Attention Model with RRAM Crossbar

Yifeng Zhai¹, Bing Li¹, Bonan Yan², Jing Wang³

Capital Normal University¹, Peking University², Renmin University of China³,

Email: zyifeng25@163.com, bing.li@cnu.edu.cn¹, bonanyan@pku.edu.cn², jwang@ruc.edu.cn³

Abstract—RRAM crossbars have been studied to construct in-memory accelerators for neural network applications due to their in-situ computing capability. However, prior RRAM-based accelerators show efficiency degradation when executing the popular attention models. We observed that the frequent softmax operations arise as the efficiency bottleneck and also are insensitive to computing precision. Thus, we propose STAR, which boosts the computing efficiency with an efficient RRAM-based softmax engine and a fine-grained global pipeline for the attention models. Specifically, STAR exploits the versatility and flexibility of RRAM crossbars to trade off the model accuracy and hardware efficiency. The experimental results evaluated on several datasets show STAR achieves up to $30.63\times$ and $1.31\times$ computing efficiency improvements over the GPU and the state-of-the-art RRAM-based attention accelerators, respectively.

Index Terms—RRAM Crossbar, Attention Model, Softmax, Processing-in-memory

I. INTRODUCTION

Though some RRAM-based accelerators specialized for attention models have been discussed [1]–[3], they primarily focus on implementing the matrix multiplications on the RRAM crossbar. In this work, we observed the execution time of softmax operation grows quickly in attention models when the input sequence length increases. The latency of softmax exceeds that of matrix multiplication when the input sequence length is 512 in the BERT-base model, which reaches up to 59.20% of the whole execution time. Though our results are observed on a GPU platform, the softmax latency problem would be exacerbated on the RRAM-based accelerators because the matrix multiplication is significantly optimized by being implemented in RRAM crossbars [4] but the softmax still runs on the same circuits. Thus, it is of significance to tailor an efficient softmax engine in RRAM-based accelerators for attention models. To this end, we propose STAR, which features an RRAM-based softmax engine by exploring the versatility and flexibility of RRAM crossbars to balance to the computing precision and efficiency. Moreover, an enhanced pipeline to balance the matrix multiplication and softmax operation in the attention is introduced. The effectiveness of STAR is verified by the comparison results with the recent RRAM-based accelerators for attention models [3].

II. RRAM-BASED SOFTMAX ENGINE

STAR is primarily composed of two types of crossbar-based processing engines: *MatMul engine* for the VMM-dominated

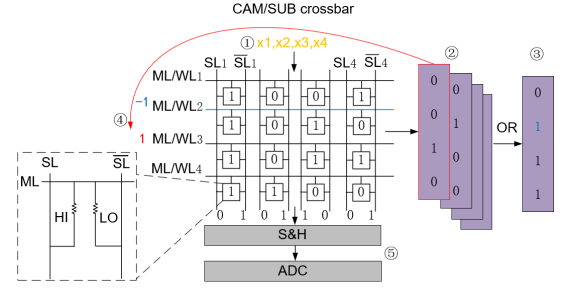


Fig. 1. The $x_i - x_{max}$ operation design.

operations and *Softmax engine* for the softmax operation, respectively. The MatMul engine follows the design in Re-Transformer [3]. As for the Softmax engine, different function units based on RRAM crossbars cooperate with each other to complete the softmax operation. The Softmax engine has two distinct stages, $x_i - x_{max}$ and the exponential operation, which desire crossbars having different functions.

1) $x_i - x_{max}$: The $x_i - x_{max}$ is achieved by one crossbar in a time-multiplex manner to complete the finding maximum and subtraction, respectively. Thus, the crossbar is denoted as CAM/SUB crossbar.

Fig. 1 shows the workflow of a 4×8 CAM/SUB crossbar to find out the max value in $[x_1 \cdots x_4]$. The crossbar works as a CAM first. For each x_i , all the WLs of the crossbar are searched in parallel and the matchlines output a one-hot vector in which ‘1’ denotes the matched line. For example, if the data stored in the WL₃ in Fig. 1 is consistent with x_1 , the output vector would be $[0, 0, 1, 0]$ (②). The outputs of matchlines cascade the OR gates that merge the search results of all input x_i (③). Because the data are stored in descending order in the CAM crossbar, the index of the first ‘1’ in the result vector corresponds to the row number of CAM storing the x_{max} . In the example of Fig. 1, x_{max} stores at WL₂. Next, the crossbar executes the subtraction $x_i - x_{max}$. The match vector outputs will be used as the input voltage vector. Instead, the input for the x_{max} row is a negative voltage (④). Thus, the output from the SLs represents the results of $x_i - x_{max}$ (⑤).

2) *Exponential Operation*: The exponential operation is implemented by CAM crossbar and LUT crossbar. A VMM crossbar collaborates with them to complete the summation in the softmax. All possible values of $x_i - x_{max}$ and their exponential results are preloaded in CAM crossbar and LUT crossbar, respectively. Since the $x_i - x_{max}$ is always negative,

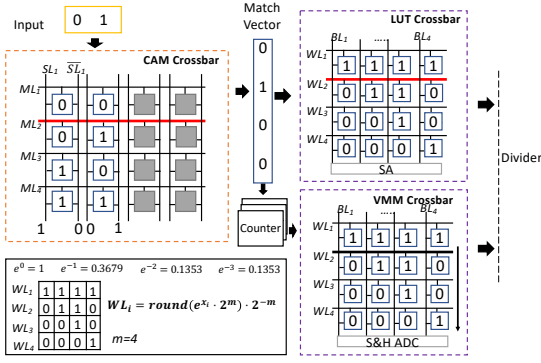


Fig. 2. The exponential operation design in our softmax engine.

TABLE I
COMPARISON WITH THE BASELINE CMOS-BASED SOFTMAX

Softmax Design	Area	Power
Softmax [5]	0.33×	0.12×
Ours (8-bit)	0.06×	0.05×

we remove the sign bit to save the area of CAM crossbar. Each input enters CAM crossbar and the output from the LUT crossbar is its exponential result. At the same time, the match vector for CAM crossbar is sent to the counter for accumulation. When all x_i complete the exponential computation, the results of the counter are sent to the VMM crossbar which stores exactly the same values as LUT crossbar to compute $\sum_{j=1}^d e^{x_j - x_{max}}$. Then the outputs of LUT crossbar and VMM crossbar enter the divider to complete the final division in the softmax.

Since the efficiency of the proposed softmax engine relates to the computing precision determined by the attention model, we analyzed the data range of all x_i across three popular datasets for the BERT-base model such that balances the computing precision and hardware efficiency with STAR. To achieve high model accuracy, the required bitwidth for CNEWS, MRPC, and CoLA are 8 bits (6-bit integer, 2-bit decimal), 9 bits (6-bit integer, 3-bit decimal), and 7 bits (5-bit integer, 2-bit decimal), respectively.

With the proposed RRAM-based Softmax engine, we introduce a vector-grained pipeline to improve the execution parallelism and efficiency for attention models. Thanks to the crossbar-based softmax engine, the complete attention mechanism operations could be in parallel in the vector granularity rather than the operand granularity in previous work.

III. EXPERIMENTAL RESULTS

We compared the proposed RRAM-based Softmax engine with an optimized CMOS-based softmax accelerator, Softmax [5] and a baselined CMOS-based softmax and compared STAR with a NVIDIA Titan RTX GPU platform and two ReRAM-based accelerators PipeLayer [6] and ReTransformer [3] to verify the collaboration of the proposed pipeline and Softmax engine.

The simulation of STAR is performed on NeuroSim [7] (for RRAM crossbar) and Synopsys Design Compiler (for the CMOS circuit), respectively. In the MatMul engine, the RRAM crossbar size is 128×128 and the precision of ADC is 5-bit by referring to [3]. In the proposed Softmax engine, the size of the CAM/SUB crossbar is 512×18 and the CAM (LUT, VMM)

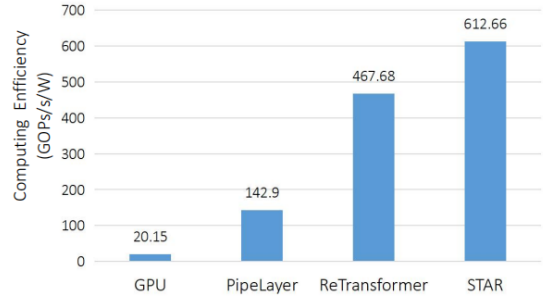


Fig. 3. Computing efficiency comparison results.

crossbar size is 256×18 to support 9-bit data and computing precision.

Table I is the comparison results of our Softmax engine with Softmax and the baseline CMOS-based softmax. Here, the evaluated model is the BERT-base model on the CNEWS dataset with a sequence length of 128. Compared to the baseline and Softmax, our Softmax engine is $0.06 \times$ and $0.20 \times$ smaller, respectively. As for power, it achieves $0.05 \times$ and $0.44 \times$ power efficient than baseline and Softmax, respectively. The results show our proposed Softmax engine offers a much better area efficiency and power efficiency than the baseline and Softmax. Fig. 3 compares the computing efficiency of GPU, Pipelayer [6], ReTransformer [3] and STAR. Computing efficiency here measures the number of operations that can be performed by a computing unit every unit time and every watt of power consumed. STAR achieves the computing efficiency of 612.66 GOPs/s/W. Compared to GPU, Pipelayer and ReTransformer, STAR improves the computing efficiency by $30.63 \times$, $4.32 \times$ and $1.31 \times$, respectively.

ACKNOWLEDGEMENT

This paper is supported by the National Natural Science Foundation of China (NSFC) under grant No. 62204164, 62222411.

REFERENCES

- [1] S. Lu, M. Wang, S. Liang, J. Lin, and Z. Wang, "Hardware accelerator for multi-head attention and position-wise feed-forward in the transformer," *arXiv preprint arXiv:2009.08605*, 2020.
- [2] H. Guo, L. Peng, J. Zhang, Q. Chen, and T. D. LeCompte, "Att: A fault-tolerant rram accelerator for attention-based neural networks," *IEEE International Conference on Computer Design: VLSI in Computers and Processors, ICCD*, 2020.
- [3] X. Yang, B. Yan, H. Li, and Y. Chen, "Retransformer: Reram-based processing-in-memory architecture for transformer acceleration," in *Proceedings of the 39th International Conference on Computer-Aided Design*, pp. 1–9, 2020.
- [4] B. Li, L. Song, F. Chen, X. Qian, Y. Chen, and H. H. Li, "Reram-based accelerator for deep learning," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 815–820, IEEE, 2018.
- [5] J. R. Stevens, R. Venkatesan, S. Dai, B. Khailany, and A. Raghunathan, "Softmax: Hardware/software co-design of an efficient softmax for transformers," *arXiv preprint arXiv:2103.09301*, 2021.
- [6] L. Song, X. Qian, H. Li, and Y. Chen, "Pipelayer: A pipelined rram-based accelerator for deep learning," in *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 541–552, IEEE, 2017.
- [7] P.-Y. Chen, X. Peng, and S. Yu, "Neurosim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures," in *2017 IEEE International Electron Devices Meeting (IEDM)*, pp. 6–1, IEEE, 2017.