南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

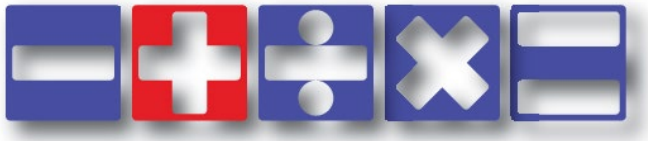
A Hardware-efficient Training-oriented Quadratic Approximation Softmax for Transformers

ARITH 2026

Zehao Cheng, Weichao Yi, Hui Chen, Chenggang Yan, Bi Wu, Ke Chen and Weiqiang Liu

College of Integrated Circuits

Nanjing University of Aeronautics and Astronautics



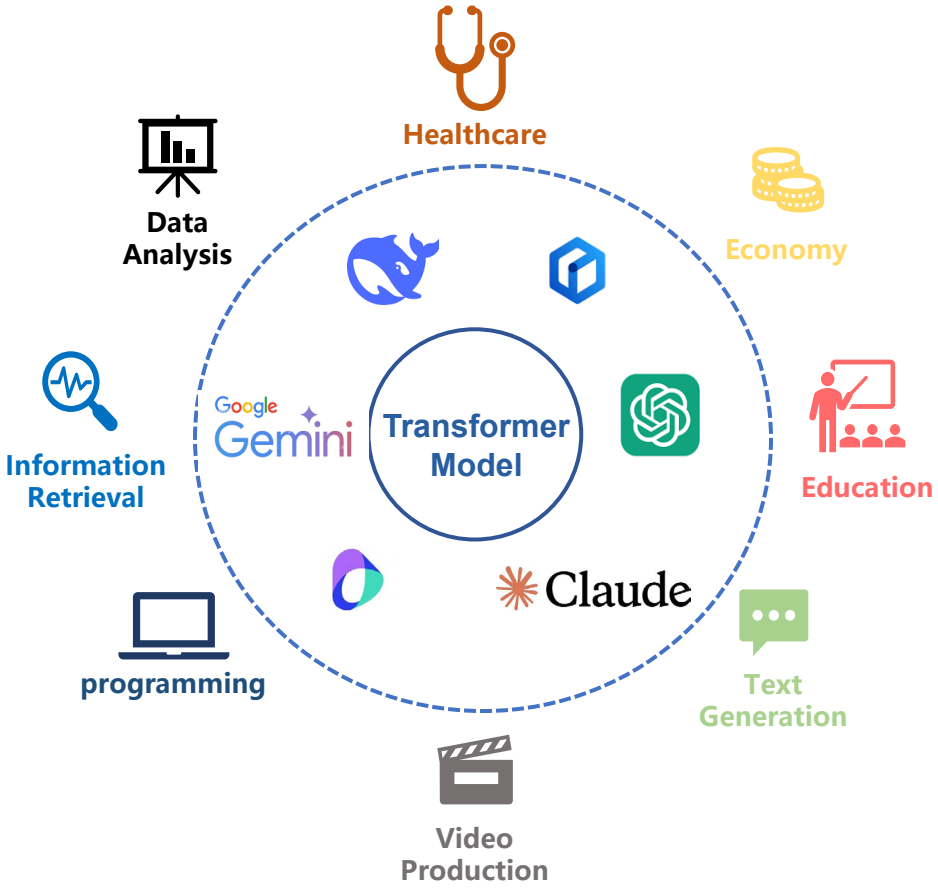
Fulda, Germany. June 28 - July 1, 2026.

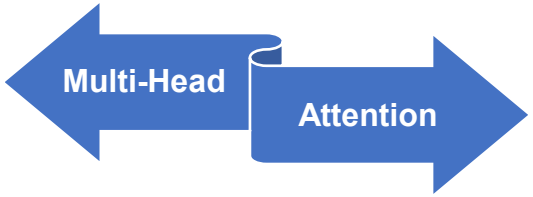
南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Content

Background: Softmax in Transformer Architectures

Softmax is the core computational component of the self-attention mechanism in Transformer networks.





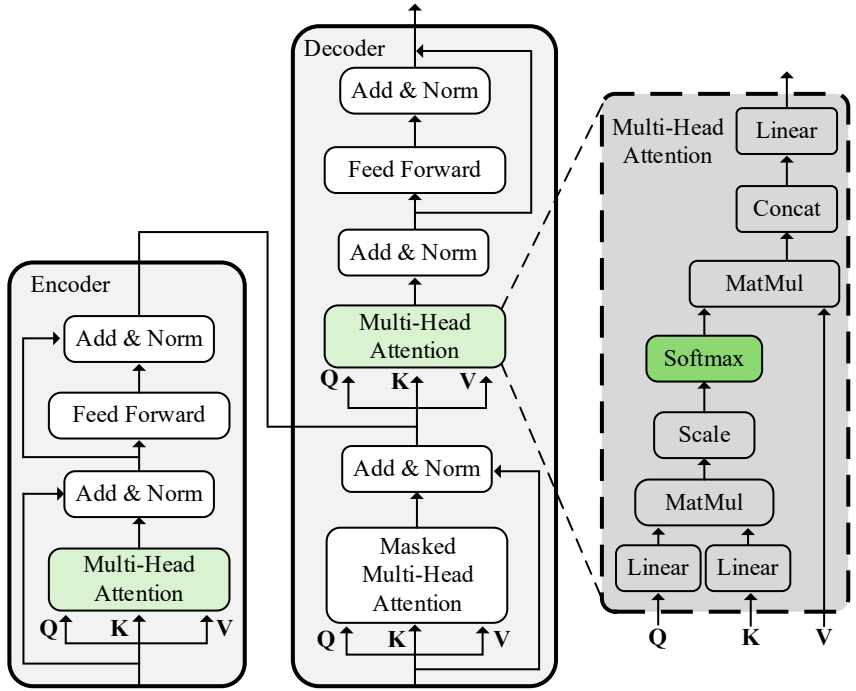


Fig.1 : Multi-Head Attention mechanism and the role of Softmax in Transformer blocks

$$softmax(x_i) = \frac{e^{x_i}}{\sum e^{x_i}}$$

Problem Statement: The Computational Bottleneck

南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Softmax has emerged as a primary computational bottleneck in Transformer-based network.

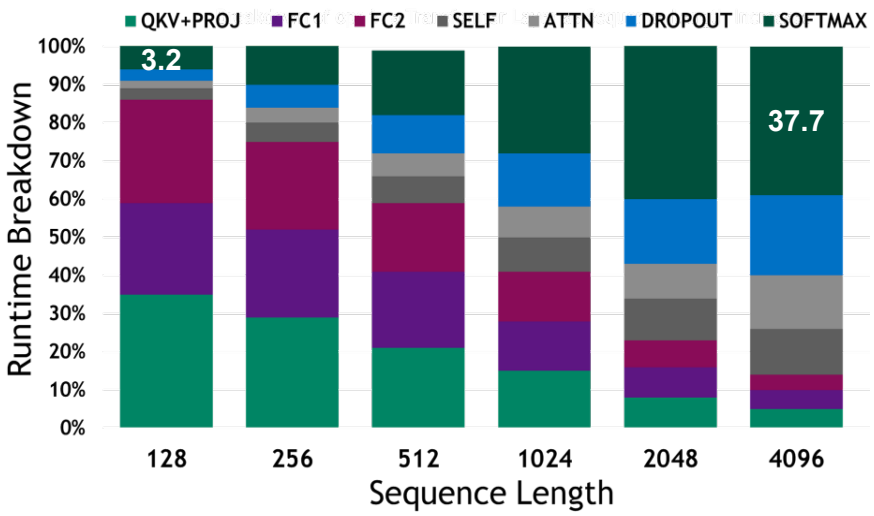
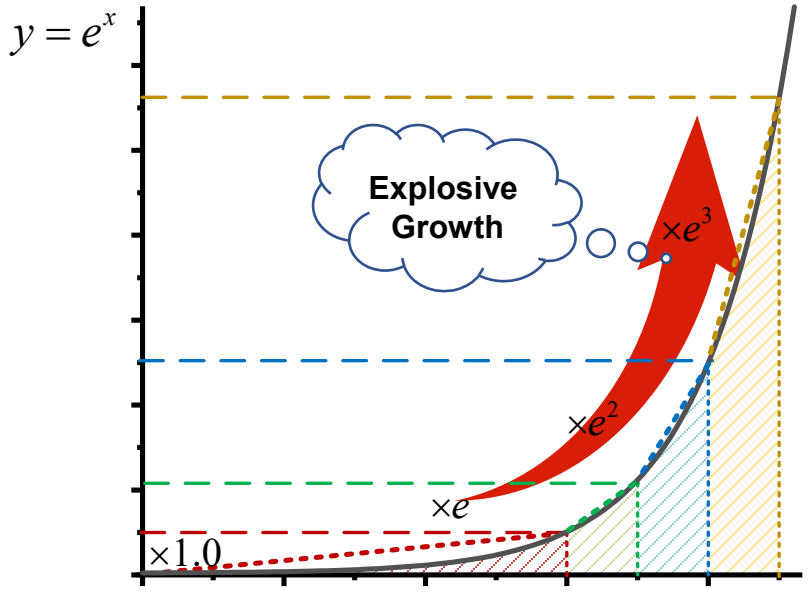
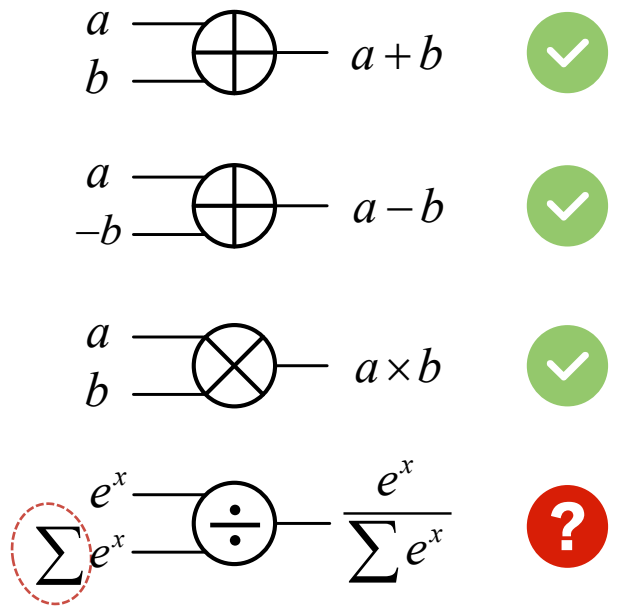


Fig. 2: Runtime breakdown for BERT-Large on a Volta GPU[10].



Growth of exponent & Nonlinearity



Global Reduction & expensive divisor operator

[10] J. R. Stevens, R. Venkatesan, S. Dai, B. Khailany, and A. Raghunathan, "Softermax: Hardware/software co-design of an efficient softmax for transformers," in 2021 58th ACM/IEEE Design Automation Conference (DAC). IEEE, 2021, pp. 469–474.

Motivation: Training-oriented High-Precision Requirements

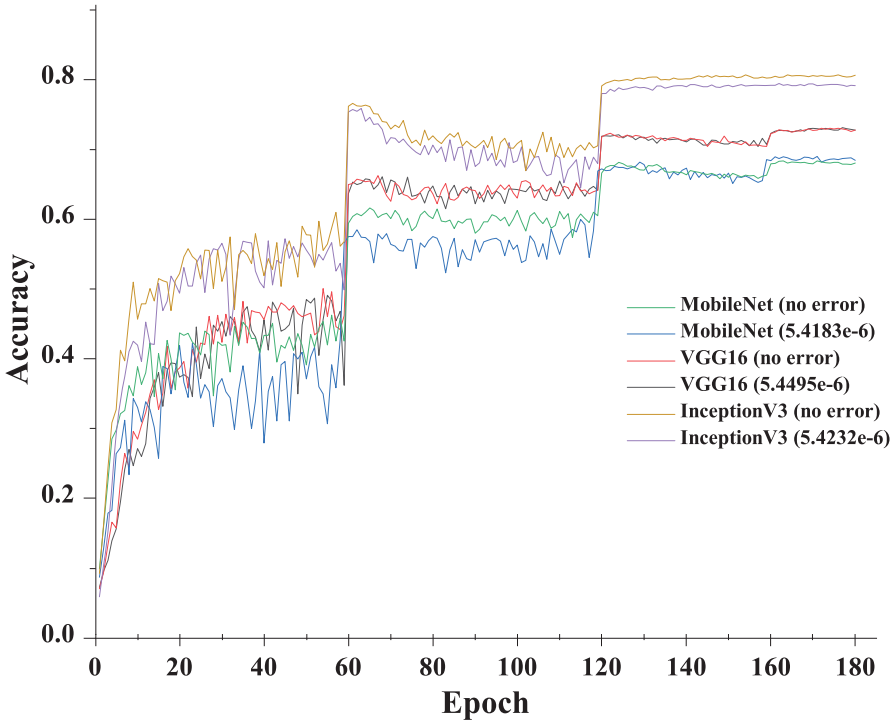


Fig3. Training accuracy of different networks with exact and approximate softmax[12].

High-Precision Requirement

Lookup Table : Excessive Storage

PiecewiseLinear : Too Many Segments

C O R D I C : Long Iteration Cycles

Error Analysis

More Analytic Approximation

Contributions:

- **TQA Error analysis**
- **TQA Softmax Hardware architecture**
- **Synthesis and Application Evaluation**

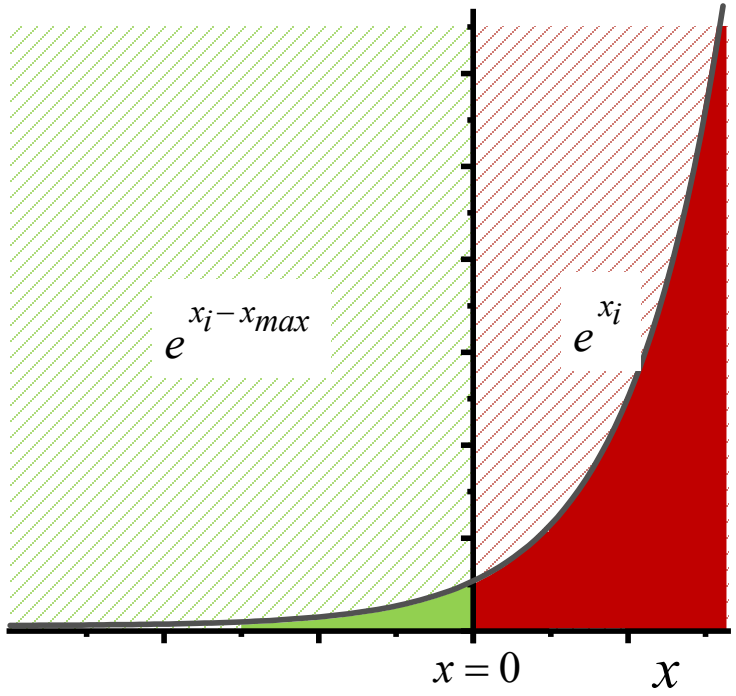
[12] Y. Zhang, L. Peng, L. Quan, Y. Zhang, S. Zheng, and H. Chen, “High precision method and architecture for base-2 softmax function in DNN training,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 70, no. 8, pp. 3268–3279, 2023.

Content

Base-2 Softmax and Down-scaling Strategy

南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

From a hardware perspective, Base-2 Softmax is a more suitable choice.



$$e^{m} = e^{m_1+m_2} = e^{m_1} \cdot e^{m_2} = e^{m_1} \gg / \ll m_2$$

$$2^{n} = 2^{n_1+n_2} = 2^{n_1} \cdot 2^{n_2} = 2^{n_1} \gg / \ll n_2$$

$e^m = 2^n$

$(n = m \times \log_2 e)$

$$softmax(x_i) = \frac{e^{x_i}}{\sum e^{x_i}} = \frac{e^{x_i} / e^{x_{max}}}{\sum e^{x_i} / e^{x_{max}}} = \frac{e^{x_i - x_{max}}}{\sum e^{x_i - x_{max}}}$$

Hardware Implementation

$$base-2\ softmax(x_i) = \frac{2^{x_i}}{\sum 2^{x_i}} = \frac{2^{x_i} / 2^{x_{max}}}{\sum 2^{x_i} / 2^{x_{max}}} = \frac{2^{x_i - x_{max}}}{\sum 2^{x_i - x_{max}}}$$

Log-Sum-Exp(LSE) and Computational Decomposition

LSE skill and computational decomposition are leveraged to optimize hardware design.

Log-Sum-Exp skill

- Decompose Softmax into exponent and log domains using LSE skill.

$$\log_2(f_2(x)) = x_i^{'} - \log_2(\sum 2^{x_j^{'}})$$

$$(x_i^{'} = x_i - x_{max})$$

- Division operator is substituted by **subtraction** in the log domain

- Transform softmax into a sequence of **two exponent** and **one logarithm** operations for optimization

$$f_2(x_i) = \frac{A}{B}$$

LSE skill

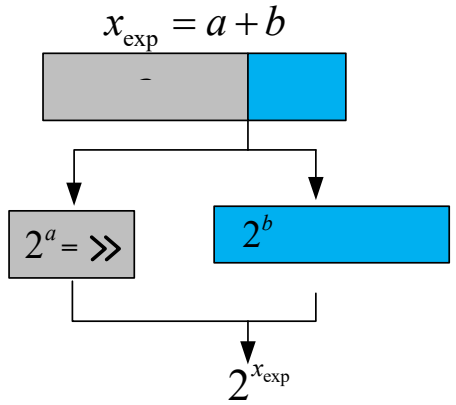
$$f_2(x_i) = 2^{A^{'} - B^{'}}$$

Exponent Decomposition

- Decompose input into
  a **negative integer a**
  and
  a **positive fraction b.**

$$x_{\exp} = a + b,\ a \in \mathbb{Z}^{-}, b \in [0,1)$$

$$2^{x_{\exp}} = 2^{b} \gg (-a)$$

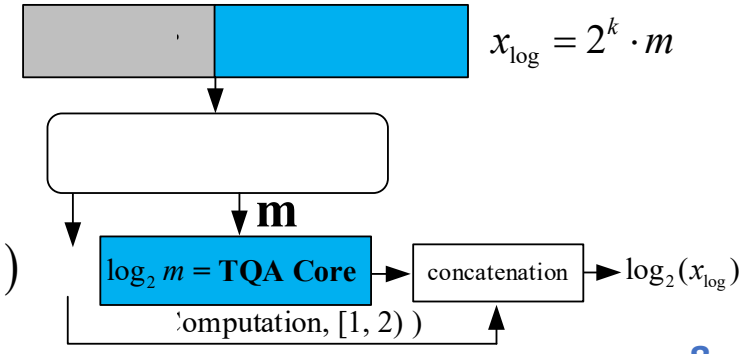


Logarithm Decomposition

- Decompose input into
  a **integer part k**
  and
  a **mantissa part b.**

$$x_{\log} = 2^{k} \cdot m,\ k \in \mathbb{Z},\ m \in [1,2)$$

$$\log_2(x_{\log}) = k + \log_2(m)$$

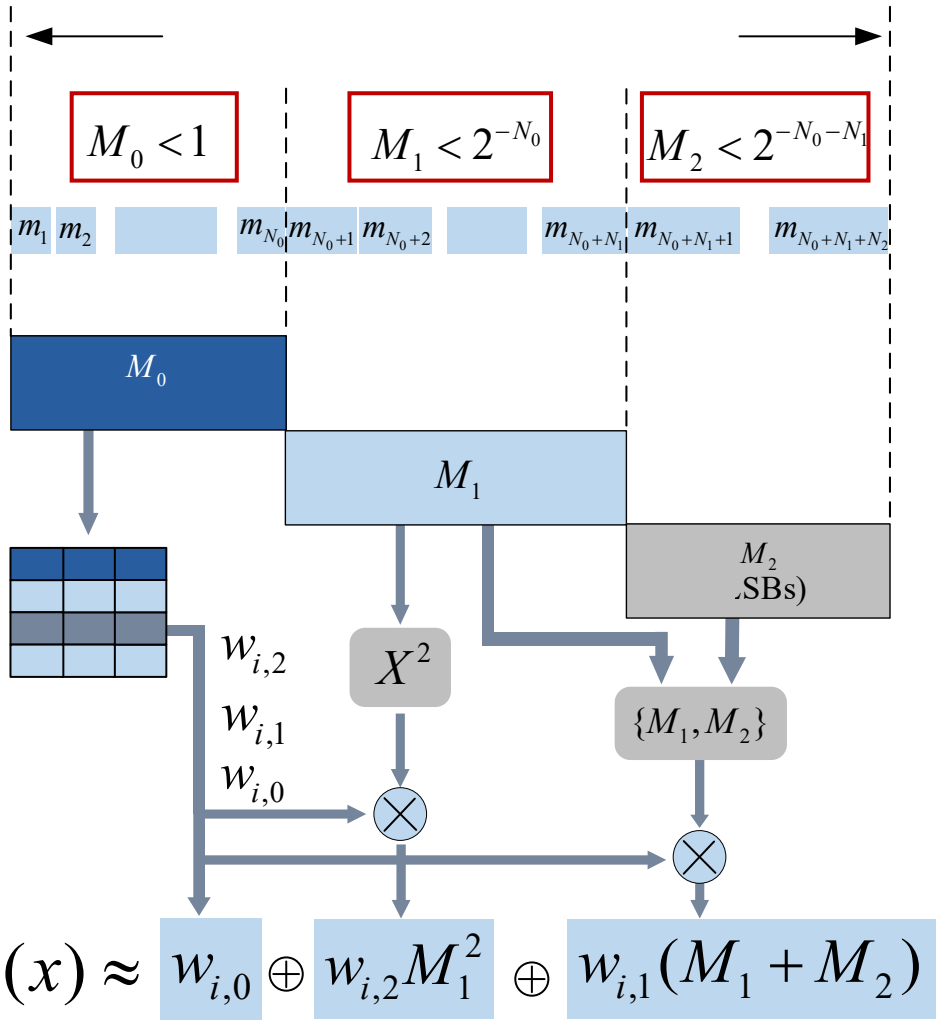


Content

TQA approximation For 2^x and $\log_2(1+x),\ x \in [0,1)$ (1)

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Quadratic Approximation and Fine-Grained Part



To meet the demand for **high-precision** computation, quadratic polynomial approximation was used.

- Input is split into **three** parts:

$$x = \{M_0, M_1, M_2\}$$

$$\begin{cases} M_0 < 1 \\ M_1 < 2^{-N_0} \\ M_2 < 2^{-N_0-N_1} \end{cases}$$

- Uniform Segments:

M_0 for fitted coefficients $w_{i,2}$

- only M_1 for quadratic term :

$w_{i,2} \cdot M_1^2$ (to avoid wasting hardware resouce on LSBs)

- (M_1+M_2) for linear term :

$w_{i,1} \cdot (M_1+M_2)$

TQA approximation For 2^x and $\log_2(1+x),\ x \in [0,1)$ (2)

Error analysis is necessary for coefficient quantization, truncation error, and curve fitting.

Variable	Bit-width
$w_{i,2}$	N_{w2}
$w_{i,1}$	N_{w1}
$w_{i,0}$	N_{w0}
M_0	N_0
M_1	N_1
M_1+M_2	N_{12}
M_1^2	N_{a1}
$w_{i,2}M_1^2$	N_{a2}
$w_{i,1}(M_1+M_2)$	N_{a3}

Overall Approximation Error (*Error*)

Coefficient Quantization Error: (e_c)

$$w_{i,2} \dashrightarrow w_{i,2}+e_{c,2}$$

$$w_{i,1} \dashrightarrow w_{i,1}+e_{c,1}$$

$$w_{i,0} \dashrightarrow w_{i,0}+e_{c,0}$$

Truncation Error: (e_t)

$$(M_1+M_2)^2 \dashrightarrow M_1^2+e_{t1}$$

$$(e_{t1}=2M_1M_2+M_2^2 \approx 2M_1M_2)$$

$$(M_1+M_2) \dashrightarrow M_1+M_2+e_{t2}$$

Model Error: (e_m)

$$y_{exact} \dashrightarrow y_{fit}+e_m$$

(More segments, Smaller model error.)

TQA approximation For 2^x and $\log_2(1+x),\ x \in [0,1)$ (3)

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Second-order polynomial expression:

$$y(x) \approx w_{i,2}M_1^2 + w_{i,1}(M_1 + M_2) + w_{i,0}$$

$$w_{i,2} \dashrightarrow w_{i,2} + e_{c,2}$$
$$w_{i,1} \dashrightarrow w_{i,1} + e_{c,1}$$
$$w_{i,0} \dashrightarrow w_{i,0} + e_{c,0}$$
$$(M_1 + M_2)^2 \dashrightarrow M_1^2 + e_{t1}$$
$$(M_1 + M_2) \dashrightarrow M_1 + M_2 + e_{t2}$$

Expression with error:

$$y(x) \approx (w_{i,2} + e_{c,2})(M_1^2 + e_{t1}) + (w_{i,1} + e_{c,1})(M_1 + M_2 + e_{t2}) + (w_{i,0} + e_{c,0})$$

subtraction

$$Error = 2w_{i,2}M_1M_2 + w_{i,1}e_{t2} + M_1^2e_{c,2} + (M_1 + M_2)e_{c,1} + e_{c,0} + e_{c,2}e_{t1} + e_{c,1}e_{t2}$$

small enough to ignore them

$$Error \approx 2w_{i,2}M_1M_2 + w_{i,1}e_{t2} + M_1^2e_{c,2} + (M_1 + M_2)e_{c,1} + e_{c,0}$$

$$Error < w_{i,2} \times 2^{-(2N_0+N_1+1)} + w_{i,1} \times 2^{-(N_0+N_{12})} + 2^{-(2N_0+Nw_2)} + 2^{-(N_0+Nw_1)} + 2^{-(Nw_0)}$$

$a+b<c$

$|a|+|b|<c$

$$Error < |w_{i,2}| \times 2^{-(2N_0+N_1+1)} + |w_{i,1}| \times 2^{-(N_0+N_{12})} + 2^{-(2N_0+Nw_2)} + 2^{-(N_0+Nw_1)} + 2^{-(Nw_0)} \le 2^{-N_{ulp}}$$

TQA approximation For 2^x and $\log_2(1+x),\ x \in [0,1)$ (4)

$$|w_{i,2}|\times 2^{-(2N_0+N_1+1)} + |w_{i,1}|\times 2^{-(N_0+N_{12})} + 2^{-(2N_0+Nw_2)} + 2^{-(N_0+Nw_1)} + 2^{-(Nw_0)} \le 2^{-N_{ulp}}$$

allocate $2^{-N_{ulp}}$ among these five terms

$$|w_{i,2}|\times 2^{-(2N_0+N_1+1)} \le \frac{1}{5}\times 2^{-N_{ulp}} \Rightarrow N_1 \ge N_{ulp} + \log_2(5) + \log_2(|w_{i,2}|) - 2N_0 + 1$$

$$|w_{i,1}|\times 2^{-(N_0+N_{12})} \le \frac{1}{5}\times 2^{-N_{ulp}} \Rightarrow N_{12} \ge N_{ulp} + \log_2(5) + \log_2(|w_{i,1}|) - N_0$$

$$2^{-(2N_0+Nw_2)} \le \frac{1}{5}\times 2^{-N_{ulp}} \Rightarrow N_{w2} \ge N_{ulp} + \log_2(5) - 2N_0$$

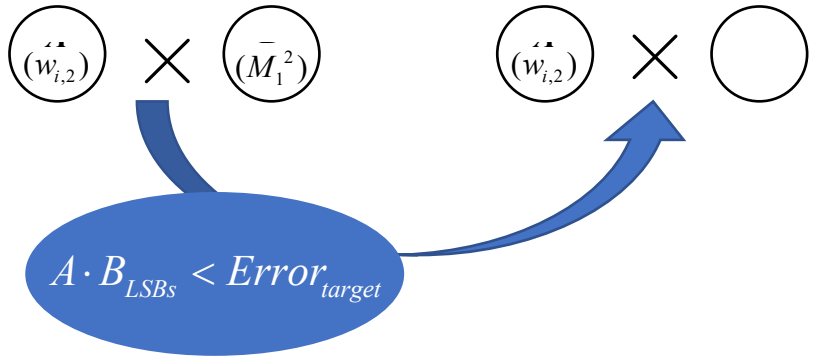
$$2^{-(N_0+Nw_1)} \le \frac{1}{5}\times 2^{-N_{ulp}} \Rightarrow N_{w1} \ge N_{ulp} + \log_2(5) - N_0$$

$$2^{-(Nw_0)} \le \frac{1}{5}\times 2^{-N_{ulp}} \Rightarrow N_{w0} \ge N_{ulp} + \log_2(5)$$

Variable	Bit-width
$w_{i,2}$	N_{w2}
$w_{i,1}$	N_{w1}
$w_{i,0}$	N_{w0}
M_0	N_0
M_1	N_1
$M_1 + M_2$	N_{12}
M_1^2	N_{a1}
$w_{i,2}M_1^2$	N_{a2}
$w_{i,1}(M_1+M_2)$	N_{a3}

TQA approximation For 2^x and $\log_2(1+x),\ x \in [0,1)$ (5)

The bit-width of $M_1^{\,2}$ (N_{a1}) can be decided through below.



$$w_{i,2}M_1^{\,2} = w_{i,2}\left[(M_1^{\,2})_{LSBs} + (M_1^{\,2})_{MSBs}\right] \approx w_{i,2} \cdot (M_1^{\,2})_{LSBs}$$

$$w_{i,2}M_1^{\,2} \rightarrow w_{i,2}M_1^{\,2} + e_{a1}$$

$$e_{a1} = w_{i,2} \times 2^{-N_{a1}} < 2^{-N_{ulp}} \rightarrow N_{a1} > N_{ulp} + \log_2(\max(w_{i,2}))$$

The bit-width of $w_{i,2}M_1^{\,2}$, $w_{i,1}(M_1+M_2)$, $w_{i,0}$ (N_{a2}, N_{a3}, N_{w0}) can be decided through below.

$$y(x) \approx [w_{i,2}M_1^2 + e_{a2}] + [w_{i,1}(M_1+M_2) + e_{a3}] + [w_{i,0} + e_{a4}]$$

allocate $2^{-N_{ulp}}$

$$e_{a2} < 2^{-N_{a2}} \le \tfrac{1}{3} \times 2^{-N_{ulp}} \rightarrow N_{a2} \ge \log_2(3) + N_{ulp}$$

$$e_{a3} < 2^{-N_{a3}} \le \tfrac{1}{3} \times 2^{-N_{ulp}} \rightarrow N_{a3} \ge \log_2(3) + N_{ulp}$$

$$e_{a4} < 2^{-N_{a4}} \le \tfrac{1}{3} \times 2^{-N_{ulp}} \rightarrow N_{w0} \ge \log_2(3) + N_{ulp}$$

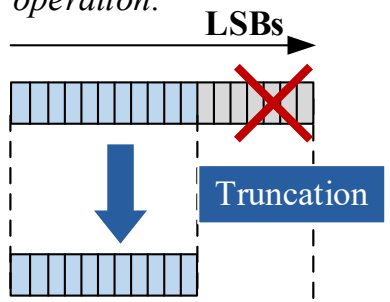
Variable	Bit-width
M_1^2	N_{a1}
$w_{i,2}M_1^2$	N_{a2}
$w_{i,1}(M_1+M_2)$	N_{a3}

Hardware-Oriented Algorithmic Optimization

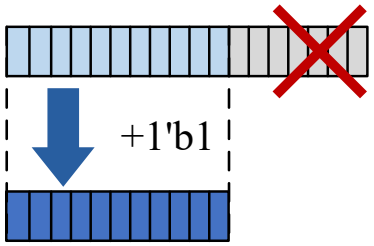
Coefficient Quantization

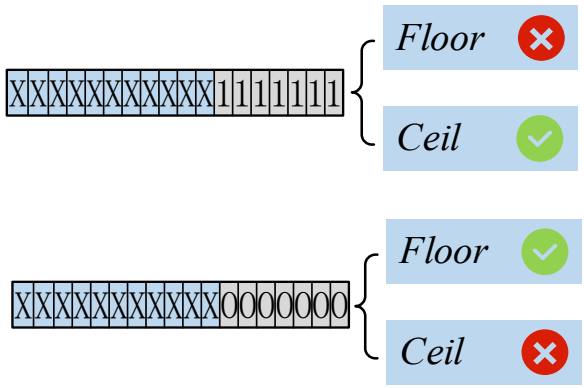
$$X\_floor = \frac{floor(X \times 2^N)}{2^N} \qquad X\_ceil = \frac{ceil(X \times 2^N)}{2^N}$$

Floor operation:



Ceil operation:

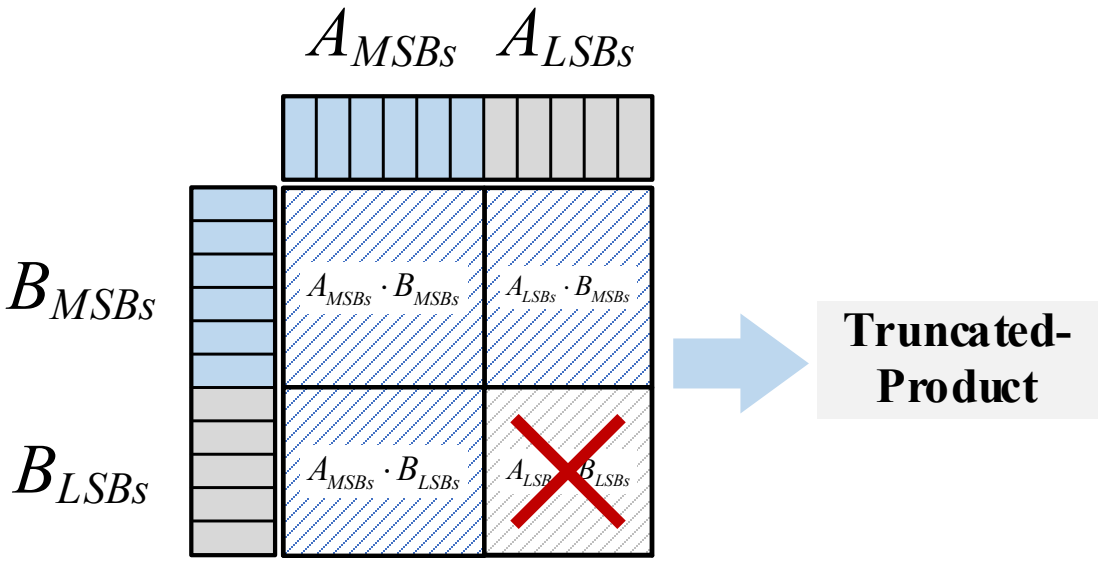




$[w_{i,2}, w_{i,1}, w_{i,0}]$,
which make error minimal

Truncated Multiplier

$$A = A_{MSBs} + A_{LSBs} \qquad B = B_{MSBs} + B_{LSBs}$$



$$Truncated\ \ product(A \times B)$$
$$\approx A_{MSBs} \times B_{MSBs} + A_{MSBs} \times B_{LSBs} + A_{LSBs} \times B_{MSBs}$$

Content

The High Precision Softmax Architecture

$$f_2(x_i)=\frac{2^{x_i}}{\sum_{j=1}^{N}2^{x_j}}=\frac{2^{x_i-x_{max}}}{\sum_{j=1}^{N}2^{x_j-x_{max}}}=2^{x_i-x_{max}-\log_2(\sum_{j=1}^{N}2^{x_j-x_{max}})}$$

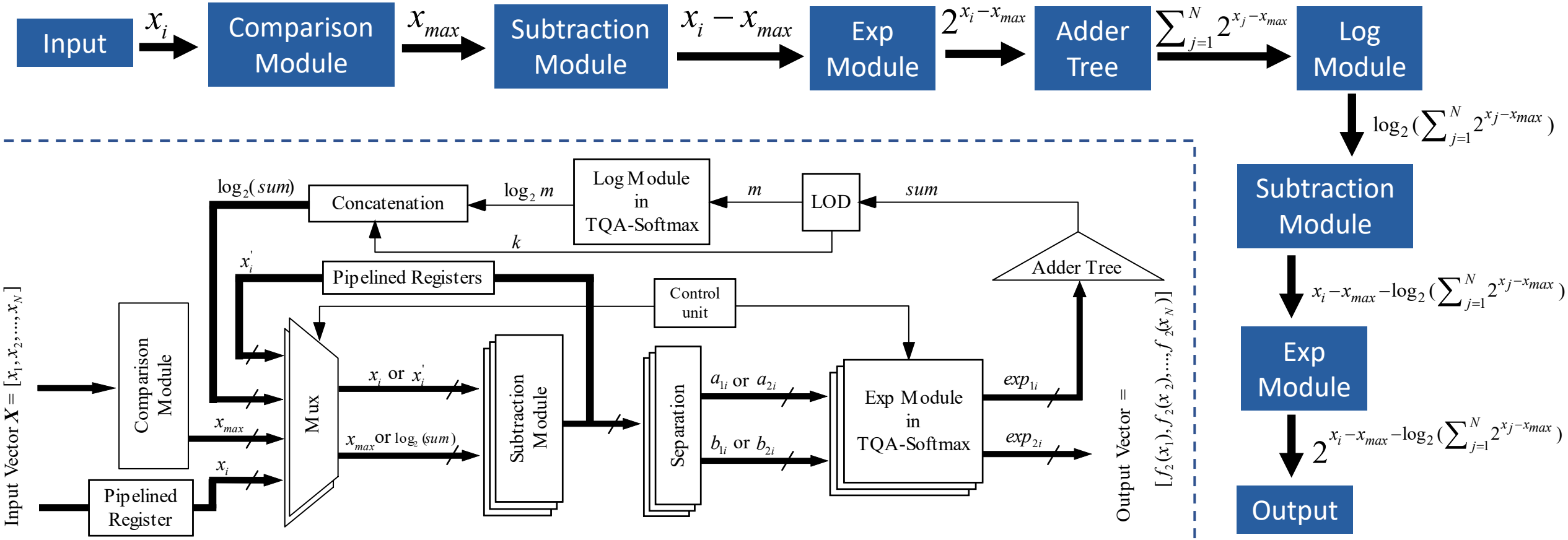
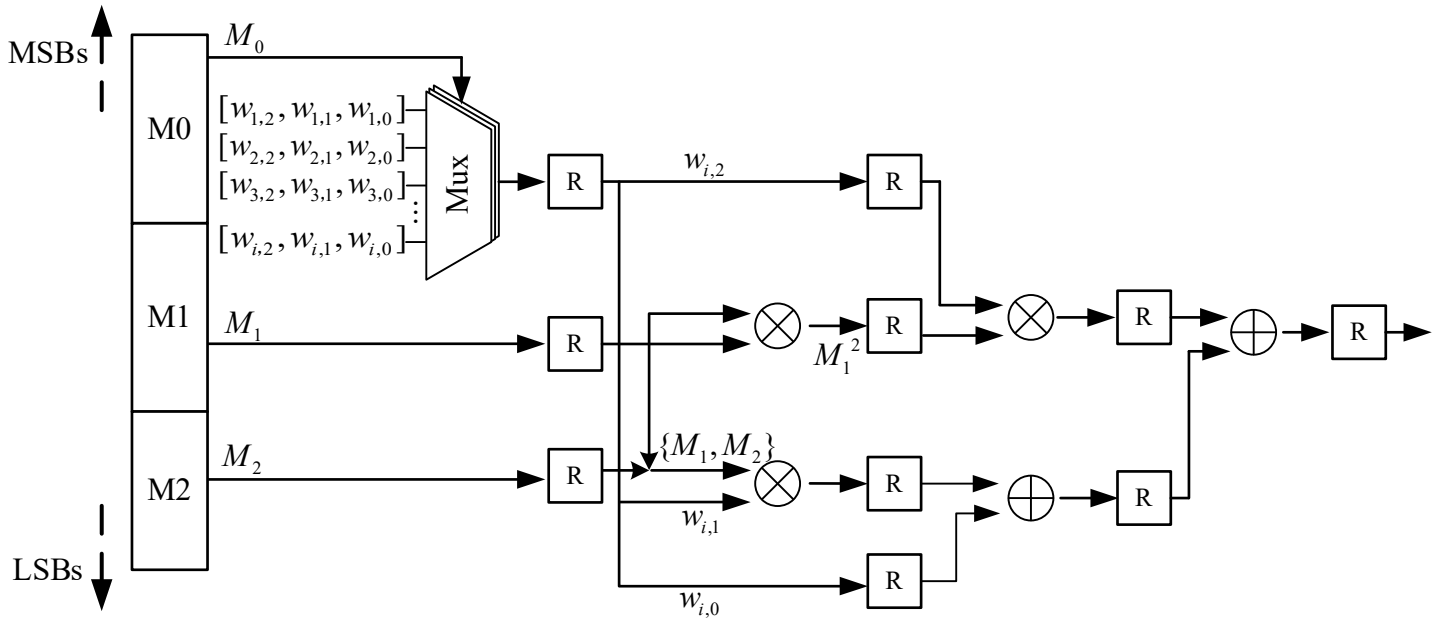


Fig.2 :TQA-Softmax Architecture

TQA Core Compute Unit



- M_0 for $w_{i,2}, w_{i,1}, w_{i,0}$.
- M_1 for quadratic term.
- M_1 & M_2 for linear term.

Content

Experiment Results (1)

南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

More Segments, Smaller Model Error （error from curve fitting）

$$Error_{target} \leq 1 \times 10^{-6}$$

$$N_{ulp} \geq -log_2(Error_{target}) \approx 19.93 \Rightarrow N_{ulp} = 21$$

Table I : Model Error of Exponent and Logarithm

N0	Exponent		Logarithm	
	MACE	MAE	MACE	MAE
3	1.04e-5	2.54e-6	3.96e-5	5.74e-6
4	**1.33e-6**	**3.17e-7**	5.42e-6	7.15e-7
5	1.68e-7	3.97e-8	**7.05e-7**	**8.94e-8**

(MAE: Mean Absolute Error)
(MACE: Maximum Absolute Calculation Error)

Table II: Numerical Error Analysis

Method	Input range	MACE	MAE
TQA-Acc	[-1,1]	8.34e-7	2.15e-7
TQA-App		1.04e-6	2.26e-7
TQA-Acc	[-5,5]	2.08e-6	2.30e-7
TQA-App		2.50e-6	2.60e-7
TQA-Acc	[-10,10]	2.92e-6	2.52e-7
TQA-App		2.98e-6	2.75e-7
[12]	—	3.81e-6	—

(TQA-Acc: using accurate multipliers)
(TQA-App: using truncated multipliers)

[12] Y. Zhang, L. Peng, L. Quan, Y. Zhang, S. Zheng, and H. Chen, “High precision method and architecture for base-2 softmax function in DNN training,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 70, no. 8, pp. 3268–3279, 2023.

Experiment Results (2)

南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Table III: Comparison of Hardware Synthesis Results

[1]Design	Area (μm^2)	Power (mW)	[2]Latency	[3]ADP ($mm^2 \cdot ns$)	[4]Efficiency (G/s/mm^2 ·mW)
TQA-Acc	26028.20	28.60	18	0.468	10.75
TQA-App	24891.80	27.44	18	0.448	11.71
CORDIC-based[12]	98787.43	24.27	26	2.568	3.276
PWL-based[30]	166804.48	40.70	17	2.836	1.178
LUT-based[14]	482304.12	45.95	17	8.199	0.361

[1]Design: All designs are synthesized at TSMC 28nm with a 1GHz clock frequency and a throughput of 8G/s.

[2]Latency: Computation Latency (clock cycles).

[3]ADP: Area-Delay Product ($mm^2 \cdot ns$).

[4]Efficiency = Throughput / (Area · Power), Throughput = Inputs × Frequency.

- Compared with TQA-Acc:

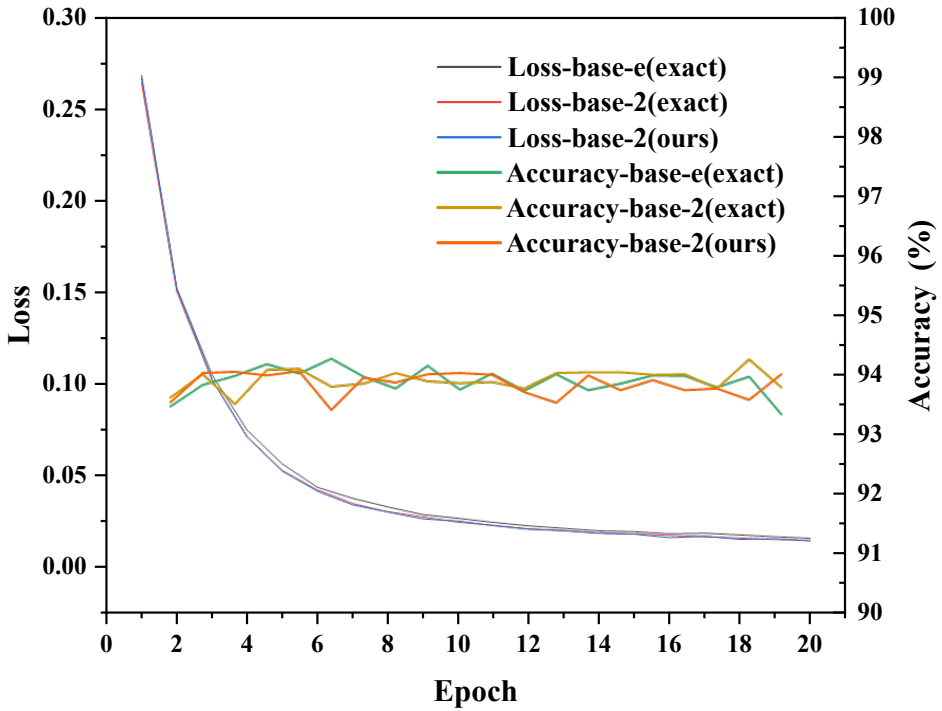
Area	↘	4.37%
Power	↘	4.06%
ADP	↘	4.27%
Efficiency	↗	8.93%

- Compared with [12]:

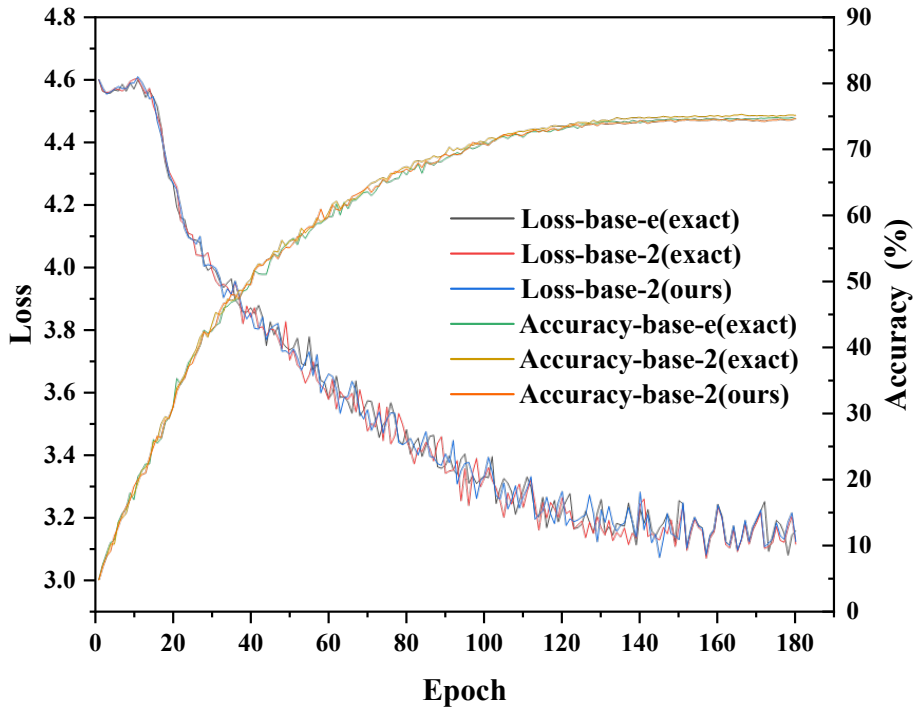
Area	↘	74.80%
Power	↗	+ 3.17
ADP	↘	- 0.448
Efficiency	↗	3.57×

[12] Y. Zhang, L. Peng, L. Quan, Y. Zhang, S. Zheng, and H. Chen, "High precision method and architecture for base-2 softmax function in DNN training," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 70, no. 8, pp. 3268–3279, 2023.
[14] Y. Zhang, Y. Zhang, L. Peng, L. Quan, S. Zheng, Z. Lu, and H. Chen, "Base-2 softmax function: Suitability for training and efficient hardware implementation," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, no. 9, pp. 3605–3618, 2022.
F. Lyu, X. Xu, Y. Wang, Y. Luo, Y. Wang, and H. Pan, "Ultralow latency VLSI architecture based on a linear approximation method for computing Nth roots of floating-point numbers," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 2, pp. 715–727, 2020.

Experiment Results (3)



(a) ViT (training from scratch)



(b) BERT-based Model (fine-tuning)

- Base-2 softmax can be used in Transformer models.
- The proposed approximation closely matches the exact baselines.

Content

Conclusions

- This paper proposes a high-precision, low-latency base-2 softmax hardware architecture for Transformer training. It achieves a mean absolute error on the order of **10^{-7}** .
- Through systematic error analysis and hardware-oriented algorithmic optimizations, the design achieves a highly robust architecture. Compared to state-of-the-art solutions, it reduces hardware area by up to **74.8%** and improves efficiency by **3.57×** .
- Network-level training experiments on Transformer models confirm that the hardware design achieves stable convergence with negligible performance degradation, proving its suitability for high-performance neural network training tasks.

Thanks!