Work In Progress: Linear Transformers for TinyML

Moritz Scherer†, Cristian Cioflan†, Michele Magno‡, Luca Benini†§
†Dept. of Information Technology and Electrical Engineering, ETH Zürich, Switzerland
‡Center for Project-Based Learning, ETH Zürich, Switzerland
§Dept. of Electrical, Electronic and Information Engineering, University of Bologna, Italy

***Abstract*— We present the WaveFormer, a neural network architecture based on a linear attention transformer to enable long sequence inference for TinyML devices. Waveformer achieves a new state-of-the-art accuracy of 98.8 % and 99.1 % on the Google Speech V2 keyword spotting (KWS) dataset for the 12 and 35 class problems with only 130 kB of weight storage, compatible with MCU class devices. Top-1 accuracy is improved by 0.1 and 0.9 percentage points while reducing the model size and number of operations by 2.5× and 4.7× compared to the state of the art. We also propose a hardware-friendly 8-bit integer quantization algorithm for the linear attention operator, enabling efficient deployment on low-cost, ultra-low-power microcontrollers without loss of accuracy.**

I. Introduction

While research on TinyML systems has made significant progress in the past, the main class of networks studied remains Convolutional Neural Networks (CNNs). The attention layer is the main bottleneck preventing transformers' adoption for embedded time-series processing in the TinyML context. The memory and computational requirements of implementing the conventional attention mechanism scale quadratically with the input length, severely limiting the ability to process long data sequences. A solution to this bottleneck is using alternative forms of attention [1], relying on random feature maps to approximate the softmax kernel without the costly explicit calculation of the attention matrix. This class of attention is typically referred to as *linear attention*.

This work in progress introduces the necessary building blocks to train and quantize full transformer models for deployment on Microcontrollers (MCUs) with a power budget in the order of milliwatts. As a concrete high-impact application, this paper proposes the WaveFormer, an accurate and lightweight transformer-based neural network. Experimental evaluation shows that the WaveFormer model improves on the state-of-the-art in Keyword Spotting (KWS) datasets, namely the 12- and 35-class Google Speech Commands (GSC) datasets [2] and computes on uncompressed, long-sequence audio waveforms, in contrast to related literature.

II. WaveFormer Model & Quantization

The main building block of the WaveFormer architecture is the Convolutional Linear Cross-Attention (CLCA) module. The CLCA implements Multi-Head Cross-attention [3] by using a convolutional projection for $\mathbf{Q}$ and a shared convolutional projection to compute the $\mathbf{K}$ and $\mathbf{V}$ matrices. Each convolutional projection consists of a depthwise convolution followed by a pointwise convolution. The depthwise convolution uses a configurable stride, which is used to reduce the sequence length.

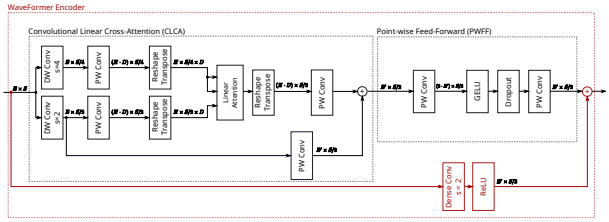

Fig. 1. Overview of the encoder used in the WaveFormer. Each depthwise convolution uses a kernel size of 5, and each dense convolution uses a kernel size of 2. Shapes of intermediate tensors are annotated above the connecting arrows. H, S, E, and D represent the number of heads, sequence length, embedding dimension, and inner dimension.

Following popular transformer literature [4]–[6], we use a pointwise feed-forward module in sequence with the attention block. Similar to other Vision Transformer inspired architectures for conventional transformers [7], the full WaveFormer consists of repeated encoder modules. Each encoder module downsamples the sequence length and optionally expands the embedding dimension. Afterward, the feature tensor is averaged over the sequence dimension. The resulting vector is passed into a dense classifier, projecting the embedding dimension to the final classification label vector. An overview of the encoder module is shown in Figure 1. The full configuration of the WaveFormer used in this work is shown in Table I.

TABLE I
Model Architecture

Layer Type	Emb. Dim.	Seq. Len.	Head Dim.	MLP Dim.
Encoder	1	8192	4	8
Encoder	4	4096	4	16
Encoder	8	2048	8	16
Encoder	8	1024	8	32
Encoder	16	512	16	32
Encoder	16	256	16	64
Encoder	32	128	32	64
Encoder	32	64	32	128
Encoder	64	32	32	128
Avg. Pooling	64	16	-	-
Dense Layer	64	1	-	-

A core issue of quantizing linear attention is performing divisions in integer arithmetic, as rounding errors in the denominator can lead to division by zero errors. A common way to address this issue is adding a numerically small constant μ to the denominator, guaranteeing that the denominator is greater than zero. To this end, we introduce an integer parameter η, which acts as a scaling factor. As a quantization constraint, we enforce that $\eta \cdot \mu$ is larger than the denominator's quantum, resulting in Equation 1, which

979-8-3503-4859-0/DATE24/© 2024 EDAA

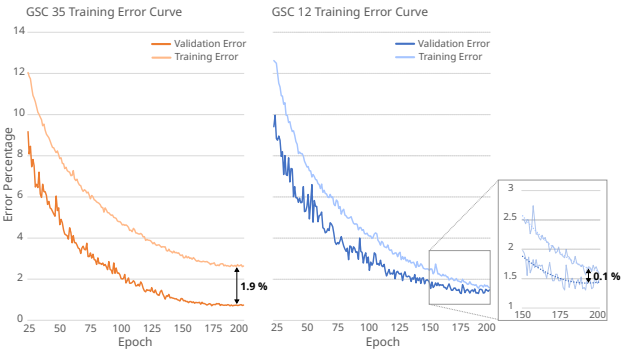

Fig. 2. Plot of error rate curves for training and validation of the 35-class and 12-class model. Notably, the training-validation margin is much larger for the 35-class model, indicating better generalization.

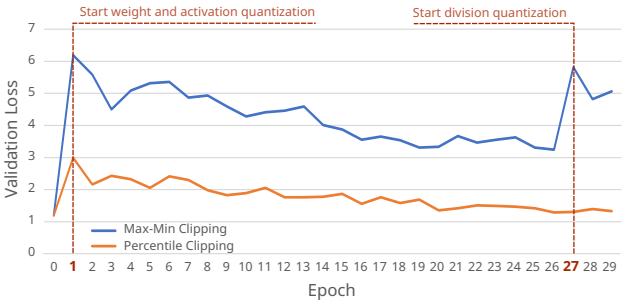

Fig. 3. Validation loss curve of the 30 epoch quantization-aware training of a 12 class model. After an initial increase in loss, the percentile clipping quantization strategy recovers, while the max-min clipping strategy diverges once divisions are quantized.

describes the proposed quantization-friendly linear attention.

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V})\,[i] \approx \frac{\eta \cdot \sum_{j=1}^{N} \Phi\left(\mathbf{Q}_i\right)^T \times \left(\Phi\left(\mathbf{K}_j\right) \times \mathbf{V}_j\right)}{\eta \cdot \sum_{j=1}^{N} \Phi\left(\mathbf{Q}_i\right)^T \times \Phi\left(\mathbf{K}_j\right) + \eta \cdot \mu} \tag{1}$$

III. Results & Conclusion

We used the Quantlib library [9] for all quantization experiments. Quantlib allows the import of a pre-trained network and the replacement of standard PyTorch layers with layers that implement fake-quantization algorithms and supports the export of fully integerized networks in a customized ONNX format. We extended the library to support percentile-based clipping initialization. Similarly, we extended the library with support for I-BERT quantization [10] of GELU layers. The retrained, quantized 12- and 35-class networks using the percentile strategy achieve an accuracy of 98.7 %, and 99.2 % on the test dataset, closely matching their unquantized versions, while the max-min clipping-based training diverges, as shown in Figure 3.

A comparison of our work with other networks on the GSC dataset is shown in Table II. Notably, our proposed model achieves the highest absolute Top-1 accuracy reported in the literature for models of any type, both on the 12-class and the 35-class problem. We attribute these improvements to two key factors; first, unlike all other models reported in literature, we do not use compressed spectrogram data but instead process raw waveforms, enabling trained, controlled compression of the model's feature space. Second, we use a novel form of linear attention, allowing us to process long sequences efficiently and capture attention effects over a long sequence of input samples within a microcontroller's stringent memory and computational constraints.

In conclusion, we show that our model outperforms the state-of-the-art in terms of accuracy by 0.1 and 0.9 percentage points while simultaneously reducing the number of parameters by 2.5× and the number of operations by 4.7×. We further propose a novel quantization scheme for linear attention layers and demonstrate quantization without loss of accuracy, enabling the use of the proposed network on resource-constrained embedded devices with limited support for floating-point computation.

Acknowledgements

This work was supported in part by EU project 101120726 — dAIEDGE — HORIZON-CL4-2022-HUMAN-02. The authors would also like to thank *armasuisse Science & Technology* for supporting this research.

References

[1] A. Katharopoulos *et al.*, "Transformers are RNNs: fast autoregressive transformers with linear attention," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, Jul. 2020, pp. 5156–5165.

[2] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," arXiv, Tech. Rep. arXiv:1804.03209, Apr. 2018, arXiv:1804.03209 [cs] type: article. [Online]. Available: http://arxiv.org/abs/1804.03209

[3] H. Lin *et al.*, "CAT: Cross Attention in Vision Transformer," arXiv, Tech. Rep. arXiv:2106.05786, Jun. 2021, arXiv:2106.05786 [cs] type: article. [Online]. Available: http://arxiv.org/abs/2106.05786

[4] A. Vaswani *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[5] A. Berg *et al.*, "Keyword Transformer: A Self-Attention Model for Keyword Spotting," in *Interspeech 2021*, Aug. 2021, pp. 4249–4253, arXiv:2104.00769 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2104.00769

[6] K. Ding *et al.*, "LETR: A Lightweight and Efficient Transformer for Keyword Spotting," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 7987–7991, iSSN: 2379-190X.

[7] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," arXiv, Tech. Rep. arXiv:2010.11929, Jun. 2021. [Online]. Available: http://arxiv.org/abs/2010.11929

[8] D. C. de Andrade *et al.*, "A neural attention model for speech command recognition," Aug. 2018, arXiv:1808.08929 [cs, eess]. [Online]. Available: http://arxiv.org/abs/1808.08929

[9] M. Spallanzani *et al.*, "QuantLab: a Modular Framework for Training and Deploying Mixed-Precision NNs," *TinyML Summit*, Mar. 2022. [Online]. Available: https://cms.tinyml.org/wp-content/uploads/talks2022/Spallanzani-Matteo-Hardware.pdf

[10] S. Kim *et al.*, "I-BERT: Integer-only BERT Quantization," arXiv, Tech. Rep. arXiv:2101.01321, Jun. 2021, arXiv:2101.01321 [cs] type: article. [Online]. Available: http://arxiv.org/abs/2101.01321

TABLE II
Comparison of the WaveFormer model with state-of-the-art networks. Best results highlighted.

Model Name	Model Type	Model Size [Params]	Ops per Inference	Top-1 Acc. GSC-12-V2	Top-1 Acc. GSC-35-V2
KWT-3 [5]	Transformer	5360 k	526.3 M	98.6 %	97.7 %
LeTR-256 [6]	Transformer	1110 k	80.7 M	-	98.2 %
Attention RNN [8]	Attention & LSTM	202 k	-	96.9 %	93.9 %
WaveFormer (This work)	Linear Transformer	130 k	19.0 M	**98.8 %**	**99.1 %**