Real-Time Low Power Audio Distortion Circuit Modeling: a TinyML Deep Learning Approach

1st Davide Plozza
Center for Project Based Learning
ETH Zürich
Zürich, Switzerland
dplozza@ethz.ch

2nd Marco Giordano
Center for Project Based Learning
ETH Zürich
Zürich, Switzerland
marco.giordano@pbl.ee.ethz.ch

3rd Michele Magno
Center for Project Based Learning
ETH Zürich
Zürich, Switzerland
michele.magno@pbl.ee.ethz.ch

***Abstract*—This work proposes a combined approach using deep learning and standard signal conditioning to model analog audio distortion circuits in real-time. The proposed model targets low latency and low power resource-constrained processors, and it is based on a quantized neural network. Pre- and post-processing consisting in filtering and dithering has been applied to improve the performance of the proposed optimized model. The model has been compared with a real-time state-of-art model WaveNet3 running on a modern desktop computer, which we used as performance reference. Our proposed model achieves a parameter size reduction of 8.4x and a neural network multiply-accumulation operations reduction of 5.1x with respect to the WaveNet3, while maintaining comparable performance. The model has been optimized and deployed on the novel MAX78000 microcontroller which features an on-board convolutional neural network hardware accelerator. Experimental results show real-time operation with a total latency of less than 10 ms and a power consumption as low as 91.8 mW in active mode, making the system suited for live music performances. Experimental results also demonstrate the capability of the hardware accelerator of the MAX78000 to reach 53.2 multiply-accumulation operations per cycle.**

***Index Terms*—Audio systems, edge computing, artificial neural networks, hardware acceleration, low-power electronics, music**

I. Introduction

Filters and effects are widely used in live music to modulate the sound outputted by musical instruments. The majority of signal processing is nowadays carried out digitally, with few exceptions that still rely on analog signal conditioning, as guitar effects and amplifiers. Guitar effects units, or ”pedals”, are devices that alter the sound of an instrument to achieve a desired modified timbre. Common effects include distortion, a distinctive ”gritty” sound, achieved through analog circuits. Since these pedals are highly non-linear it is particularly hard to model their effect with digital circuits, and the hard constraint of real-time signal processing with very short latency requirements for live music performance make their simulation particularly challenging. An emerging application for low-power battery operated digital simulations is on-board audio processing on Smart Musical Instruments [1].

There are mainly two types of digital models of analog circuits: white-box models and black-box models [2], [3]. White box models are based on analysis and simulations of analog circuits. Detailed knowledge of the circuit is required, and simulations can be very computationally demanding, making challenging to achieve the required low latency using low power processors [2], [3]. Contrarily, black-box filters do not rely on hefty simulations, but try to learn the behaviour of the analog circuit by observing input-output measurements to replicate its effect. Recently, innovative deep learning techniques have been successfully employed for black-box modeling. Temporal Convolutional Neural Networks (TCN) are a particular good fit for this type of application since its building blocks are composed of a (convolution) filter and a non-linear activation function [2]. Moreover, neural networks have been deeply improved and optimized, seeing efforts from both the research and industry communities to accelerate inference, thanks to software libraries [4] and dedicated hardware [5].

A novel branch of machine learning is TinyML, where neural network models are shrunk in dimension and new trade-offs are found between model complexity, inference time and energy [6]. Novel hardware has been developed for this purpose, specializing in speeding up neural network inference ”at the edge” [7], [8]. Contrary to CPU- or GPU-based inference, Edge AI reduces the dimension of the final product, the energy consumption and the overall latency [9].

An interesting and novel microcontroller that enables a new generation of TinyML is the MAX78000 from Analog Devices. It features a Cortex M4F, a RISC-V and a neural network accelerator all in the same package, which is able to boost inference time more than 500x with respect to a Cortex M4F alone. The accelerator in particular traded off the flexibility of a general-purpose microcontroller with a better inference speed of a limited set of NN operations implemented in hardware. Moreover, it comes with 512KB of on-chip SRAM, able to host TinyML models of medium-large size. Thus, in order to exploit all the power of the accelerator and be able to deploy a usable real-time application, a thorough revisiting of the current state-of-the-art neural network for analog distortion circuit modeling was required. This paper proposes signal processing techniques to reduce the loss of model performance caused by the simplification and quantization of the neural network. The proposed approach targets resource-constrained processors that perform better with fixed-point operation and have a limited amount of memory.

The contribution of the paper are as follow:

978-1-6654-0996-4/22/$31.00 ©2022 IEEE

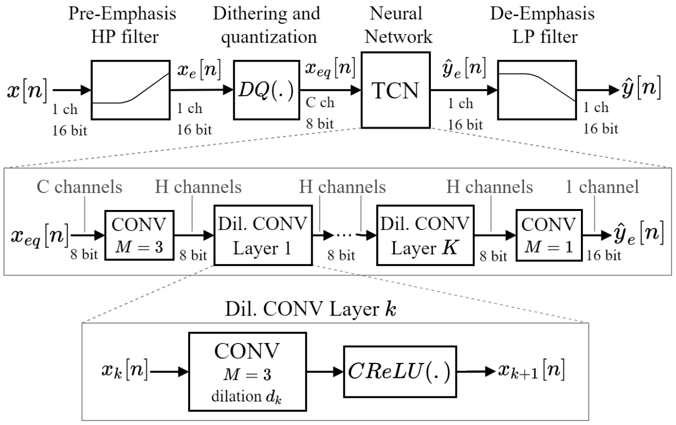

Fig. 1. Black-box model pipeline and neural network details.

- An efficient TinyML black-box model for analog distortion circuits is proposed. The network requires only $15\,\text{kB}$ of parameter memory and it is fully quantized to 8-bit.
- A signal processing and conditioning pipeline is proposed and implemented to further improve the neural network's performance.
- The proposed solution is deployed on the MAX78000 microcontroller to experimentally evaluate the performance in terms of latency, energy and operations per cycle.

II. Proposed TinyML Methods

The proposed black-box model is based on the model from [2], which is in turn based on the WaveNet model proposed in [10]. In this work, the state of the art model is adapted to work with the limitations of the MAX78000 CNN accelerator and successfully deployed on the embedded hardware with real-time performance. Pre- and post-processing have been added to achieve comparable prediction performance to the original model, and a new tradeoff between model size and prediction performance is found and proposed.

The processing pipeline of the whole model is shown in Fig. 1. The input and output consist of a 16-bit Pulse-Code Modulated (PCM) single channel signal sampled at $44.1\,\text{kHz}$. The input signal is processed with a high-pass (HP) pre-emphasis filter, then the resulting signal is split into C channels, and for each of them dithering and requantization to 8-bit is applied. The resulting C channels are fed as input to a neural network, which outputs a single 16-bit channel. Finally, the signal is passed through a low-pass (LP) de-emphasis filter.

A. Temporal Convolution Neural Network

As the core of the model we employ a causal dilated TCN, whose architecture is shown in Fig. 1. The most significant difference from the WaveNet model in [2] consists of the 8-bit quantization of the weights and activation signals, which strongly affect the quality of the output audio without the improvements proposed in this work. Additionally, we simplify the model by omitting residual and skip connections in order to increase inference speed on the hardware accelerator.

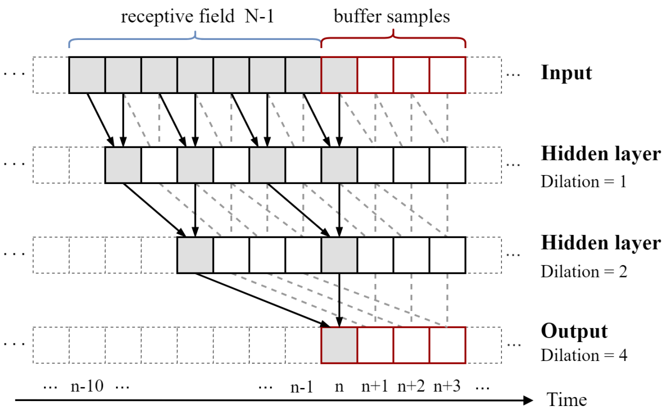

Fig. 2. Simplified example of the dilated WaveNet architecture, with one input and one hidden channel, no input and output convolutional layers, $M = 3$ and $N = 8$. The grayed out neurons contribute to the prediction of the output sample at timestamp n.

We researched different configurations with fewer layers, thus smaller receptive fields. We show that thanks to the proposed pre- and post-processing, our model is able to achieve comparable performance to the best performing real-time model WaveNet3 [2].

The network is composed of K dilated TCN layers, enclosed by 2 non-dilated convolutions without activation. The details of the dilated TCN layer are shown in Fig. 1. It consists of a dilated convolution with a 1xM kernel and dilation factor d_k followed by a clipped ReLU nonlinearity, defined as $CReLU(.) = min(max(0,.), 128)$. The clipping is done by the MAX78000 hardware to confine the output in 8-bit range.

The network can be interpreted as a causal nonlinear FIR filter, in which the impulse response length equals the network's receptive field size. We choose a filter size $M = 3$ and a dilation pattern $d_k = \{1, 2, 4, ..., 128\}$, which result in a receptive field of size $N = 513$. An example of the dilation pattern and causality property can be found in Fig. 2.

For real-time inference, instead of exploiting the WaveNet structure by caching hidden activation values between inferences as in [2], we exploit the parallelism offered by the CNN hardware accelerator and perform separate inferences on sequential buffers of samples, as shown in Fig. 2.

The quantization to 8 bit of the input and hidden layers introduces noise at the TCN output, prompting us to treat the neural network as a noisy communication channel. If dithering is applied to the input signal as described in the next section, the total noise introduced by requantization operations is iid colored additive noise. We observe that this noise is stronger at high frequencies, thus applying a pre- and de-emphasis filter before and after the network reduces the noise in the output, increasing the SNR of the network. The use of emphasis filtering to reduce quantization noise is discussed in [11]. Quantization noise can be further reduced by having multiple dithered input channels, as described in the next section.

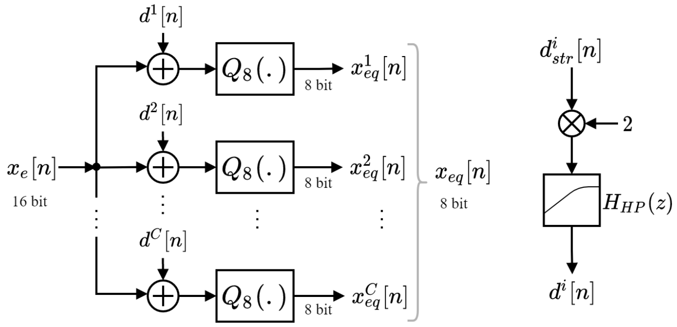

Fig. 3. Multichannel dithering and requantization operations.

B. Pre- and post-processing

The pre-emphasis filter was chosen as the first-order FIR HP filter with transfer function $H_{pre}(z)$ defined in (1). which is very commonly used in speech processing [12] to flatten the spectrum. The de-emphasis filter is the inverse of the pre-emphasis filter, thus it is a first-order IIR LP filter with transfer function $H_{de}(z)$ defined in (1).

$$H_{pre}(z) = 1 - 0.9z^{-1} \qquad H_{de}(z) = \frac{1}{1 - 0.9z^{-1}} \tag{1}$$

The second step of the preprocessing consists first in splitting the pre-emphasized input in C identical channels, then a dither signal $d^i[n]$ is added to each channel i, which is finally requantized to 8 bit. The dither signal is generated independently for each channel, so that $d^i[n]$ is statistically independent from $d^j[n]$ for $i \neq j$. We use a non-subtractive colored (filtered) dither signal. To generate each signal $d^i[n]$, a white dither signal $d^i_{str}[n]$ sampled iid from a 2-LSB peak-to-peak symmetrical triangular distribution is multiplied with 2 and filtered with a Butterworth second order HP filter $H_{HP}(z)$ with cutoff frequency of $17\,\mathrm{kHz}$, as shown in Fig. 3. The idea is to shift the dither noise energy to high frequencies, that will then be filtered out more effectively by the de-emphasis filter. The filter parameters were tuned empirically, aiming to generate the minimum noise at the output of the model while still eliminating unwanted quantization distortion.

Dithering has the effect of decorrelating the quantization noise from the signal, thus the resulting noise can be treated as iid colored noise. This effectively preserves more information in the quantized signal, and according to psychoacustics tests, the iid noise is not perceived as unpleasant distortion [13]. The empirical effect of having a larger number of input channels with statistically independent dithering is that the quantization noise is more attenuated at the output of the network.

C. Training and deployment

The network was trained using the error-to-signal ratio (ESR) with respect to the training data as loss function. The ESR can be considered as an energy-normalized sum-of-squares error and it is suitable for audio applications [2], [3]. Our definition of the ESR for the ith training example is given by

$$\mathcal{E}^{\{i\}} = \frac{\sum_{n=-\infty}^{\infty} |y_e^{\{i\}}[n] - \hat{y}_e^{\{i\}}[n]|^2}{\sum_{n=-\infty}^{\infty} |y_e^{\{i\}}[n]|^2 + \lambda} \tag{2}$$

where $y_e^{\{i\}}$ is the target signal pre-emphasized with the filter (1), and $\hat{y}_e^{\{i\}}$ is the neural network output. Training with pre-emphasized data helps the network to model high frequency components [2]. We introduced an additional factor λ to improve the stability of ESR, necessitated by the addition of dithering, which was fixed to 0.1 after hyperparameter search.

In this work, we model three types of distortion effects: an overdrive pedal (Ibanez Tube Screamer TS808), which introduces soft-clipping of the input waveform, a distortion pedal (ElectroHarmonix OD Glove), in which the signal is clipped more harshly, thus more nonlinear than the previous, and a ”fuzz” pedal (ElectroHarmonix Big Muff Pi), an effect that has an even higher distortion. The training data was generated with the same method as in [2]. Raw guitar sounds are processed through the distortion pedals and recorded to create the input/target data pairs. The raw guitar input sound were obtained from the IDMT-SMT-Guitar dataset [14], from which subsets of approximately 3.5, 0.5 and 1 minutes were selected as train, validation and test sets respectively.

For the experiments, both the proposed network and the reference model from [2] were trained with the Adam optimizer. Training data was split into $100\,\mathrm{ms}$ training examples with a mini-batch size of 64. The proposed network was trained with quantization-aware-training, then quantized to 8-bit using the toolchain provided by Analog devices, based on PyTorch.

The same toolchain was used to deploy our model on the MAX78000 MCU. The pre- and post-processing runs on the ARM Cortex M4F core running at 100MHz and was implemented using the optimized CMSIS-DSP library.

III. Results

To evaluate the performance, the ESR of the model output $\hat{y}$ with respect to the target y was computed over the test set. An audio buffer size (number of predicted samples per inference) of 192 samples was chosen, which at $44.1\,\mathrm{kHz}$ has duration of $4.35\,\mathrm{ms}$. To achieve real-time operation, the total model inference time has to be smaller than the buffer duration. The latency of the model amounts to the buffer length plus the inference time, while round-trip-latency (RTP) considers also the latency introduced by the ADC and DAC, which is typically negligible on modern hardware. Thus, a buffer of 192 samples is suitable to achieve an RTP latency under $10\,\mathrm{ms}$.

Fig. 4 shows the ESR losses for the OD Glove pedal against the inference times (CNN inference times, pre- and post-processing times) of models with different configurations. As expected, the ESR decreases consistently with more hidden channels, while the number of input channels mostly affects the noise reduction, a quality that is very noticeable by listening tests but it is not captured perfectly by the ESR metric. The lowest ESR is achieved by the 16 input channels and 24 output channels configuration; thus we select it for detailed evaluations.

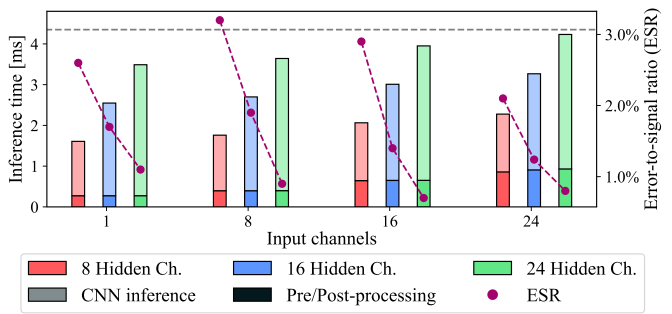

Fig. 4. ESR losses (OD Glove pedal) and inference times of models with different input channels C and hidden channels H. Configurations with inference times values under the gray line (4.35 ms) can be run in real-time.

The MAX78000 implementation of this model configuration can be run in real-time as it shows a total inference time of 3.95 ms, which is shorter than the buffer duration 4.35 ms, and is decomposed in 3.3 ms of CNN inference and 0.65 ms of pre- and post-processing. Each full inference uses 399 µJ of energy, which divided by the audio buffer duration of 4.35 ms results in an average power usage of 91.8 mW, enabling long battery operation. The neural network requires 15 kB of weight memory and needs 8.8 M multiply-accumulation (MAC) operations to predict the 192 output buffer samples. We achieve 53.2 MAC operations per cycle during inference on the CNN hardware accelerator.

Table I compares the performance of the reference models WaveNet3 and WaveNet1 from [2] with our best real-time model, with 16 input channels and 24 hidden channels. For comparison we also show a non-quantized, floating point (FP) version of our model. We compare the ESR performances for the three modeled pedals as well as the network complexity.

TABLE I
COMPARISON BETWEEN OUR MODEL AND THE REFERENCE MODEL

Model	Size	MACs	ESR		
			TS808	OD Glove	Big Muff Pi
Our	15 kB	8.8 M	0.44%	0.74%	12.58%
Our - FP	60 kB	8.8 M	0.15%	0.53%	11.96%
WavenNet1	67 kB	29.1 M	0.19%	0.98%	9.63%
WavenNet3	125 kB	44.7 M	**0.11%**	**0.21%**	**8.26%**

We see that our proposed quantized model performs comparably to both the WaveNet3 and WaveNet1 models. As observed in [2], the ESR is larger for more nonlinear circuits, this behaviour is also exhibited by our model. The WaveNet models target modern desktop computers and can run in real-time on an Apple iMac with an 2.8 GHz Intel processor [2], while our model runs in real-time on a Cortex M4F clocked at 100 MHz and a CNN accelerator clocked at 50 MHz.

We note that the ESR error is a good metric but does not fully represent how good a simulation is to the human ear. Furthermore, it is not possible to determine a threshold ESR value that characterizes an acceptable audio quality. Future work include using perceptually motivated pre-emphasis filtered ESR, as proposed in [15], and performing psychoacoustic hearing tests, especially to assess the quality of the noise reduction techniques proposed in this work.

IV. CONCLUSION

This work proposed a tinyML optimized low latency neural network based model to simulate audio distortion circuits on a low power resource-constrained microcontroller. Standard signal processing, as filtering and dithering, has been employed to improve the performance of the deep learning model in terms of latency, memory requirements and energy. A neural network parameter size reduction of 8.4x and a MAC operation reduction of 5.1x with respect to the WaveNet3 reference model was achieved with comparable performance. A real-world implementation exploiting the CNN accelerator of the novel MAX78000 microcontroller has been evaluated, which achieves real-time operation with RTP latency below 10 ms and requires 91.8 mW of power. Thus, we conclude that this implementation is suited for low power, battery operated digital simulations intended for live music performances.

REFERENCES

[1] L. Turchet, “Smart musical instruments: Vision, design principles, and future directions,” *IEEE Access*, vol. 7, pp. 8944–8963, 2019.

[2] E.-P. Damskagg, L. Juvela, and V. Valimaki, “Real-Time Modeling of Audio Distortion Circuits with Deep Learning,” p. 8.

[3] E.-P. Damskägg, L. Juvela, E. Thuillier, and V. Välimäki, “Deep Learning for Tube Amplifier Emulation,” *arXiv:1811.00334 [cs, eess]*, Feb. 2019.

[4] X. Wang, M. Magno, L. Cavigelli, and L. Benini, “FANN-on-MCU: An Open-Source Toolkit for Energy-Efficient Neural Network Inference at the Edge of the Internet of Things,” *arXiv:1911.03314 [cs, eess, stat]*, Feb. 2020.

[5] D. Rossi, F. Conti, A. Marongiu, A. Pullini, I. Loi, M. Gautschi, G. Tagliavini, A. Capotondi, P. Flatresse, and L. Benini, “Pulp: A parallel ultra low power platform for next generation iot applications,” in *2015 IEEE Hot Chips 27 Symposium (HCS)*, pp. 1–39, 2015.

[6] C. R. Banbury *et al.*, “Benchmarking TinyML Systems: Challenges and Direction,” *arXiv:2003.04821 [cs]*, Jan. 2021.

[7] M. G. Ulkar and O. E. Okman, “Ultra-Low Power Keyword Spotting at the Edge,” *arXiv:2111.04988 [cs, eess]*, Nov. 2021.

[8] M. Giordano and M. Magno, “A Battery-Free Long-Range Wireless Smart Camera for Face Recognition,” in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, SenSys ’21, (New York, NY, USA), pp. 594–595, Association for Computing Machinery, Nov. 2021.

[9] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet of Things Journal*, vol. 3, pp. 637–646, Oct. 2016. Conference Name: IEEE Internet of Things Journal.

[10] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A Generative Model for Raw Audio,” *arXiv:1609.03499 [cs]*, Sept. 2016.

[11] J. R. Stuart and R. J. Wilson, “Dynamic Range Enhancement Using Noise-Shaped Dither Applied to Signals with and without Preemphasis,” Audio Engineering Society, Feb. 1994.

[12] K. S. Rao and S. G. Koolagudi, *Emotion Recognition using Speech Features*. SpringerBriefs in Electrical and Computer Engineering, New York, NY: Springer New York, 2013.

[13] S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy, “Quantization and Dither: A Theoretical Survey,” *Journal of the Audio Engineering Society*, vol. 40, pp. 355–375, May 1992. Publisher: Audio Engineering Society.

[14] C. Kehling, J. Abeer, C. Dittmar, and G. Schuller, “Automatic Tablature Transcription of Electric Guitar Recordings by Estimation of Score- and Instrument-related Parameters,” p. 8, 2014.

[15] A. Wright and V. Välimäki, “Perceptual Loss Function for Neural Modelling of Audio Systems,” *arXiv:1911.08922 [cs, eess]*, Nov. 2019.