Keyword Spotting System using Low-complexity Feature Extraction and Quantized LSTM

Kévin Hérissé, Benoit Larras, Antoine Frappé, Andreas Kaiser
Univ. Lille, CNRS, Centrale Lille, Junia, Univ. Polytechnique Hauts-de-France, UMR 8520 – IEMN
Lille, France
{name.surname}@junia.com

***Abstract*—Long Short-Term Memory (LSTM) neural networks offer state-of-the-art results to compute sequential data and address applications like keyword spotting. Mel Frequency Cepstral Coefficients (MFCC) are the most common features used to train this neural network model. However, the complexity of MFCC coupled with highly optimized machine learning neural networks usually makes the MFCC feature extraction the most power-consuming block of the system. This paper presents a low complexity feature extraction method using a filter bank composed of 16 channels with a quality factor of 1.3 to compute a spectrogram. It shows that we can achieve an 89.45% accuracy on 12 classes of the Google Speech Command Dataset using an LSTM network of 64 hidden units with weights and activation quantized to 9 bits and inputs quantized to 8 bits.**

Keywords—Keyword Spotting, Machine Learning, Long Short-Term Memory, MFCC

I. Introduction

The latest developments in consumer electronics made voice-activated devices used every day. The need to embed Keyword Spotting (KWS) solutions at the edge led to the development of always-on low-power preprocessing units to avoid the computation of the audio signal by power-hungry elements. Figure 1 shows a typical architecture, in which a preprocessing unit triggers the main processor only if the analyzed audio signal is a relevant keyword. The unit is composed of a feature extractor that will divide the audio signal into multiple frequency bands to compute the per band energy. A classification neural network uses these features as inputs to detect if the audio signal corresponds to one of the predefined classes learned by the classifier. Long Short-Term Memory (LSTM) [1] neural networks are well-suited classifiers to manage sequential data. However, LSTMs require lots of data and computational power and therefore need to be optimized for integration at the edge. This is achieved, for example, by training the network with a small number of hidden units (56 in [2]) or by quantizing weights (5-bit in [3]). These optimizations are made possible thanks to the use of input features such as Mel Frequency Cepstral Coefficients (MFCC). However, MFCC extraction requires Fast Fourier Transforms (FFT) and Discrete Cosine Transforms (DCT). For this reason, the feature extraction (FE) block usually consumes most of the energy of the system. To tackle this challenge, this paper presents the following contributions:

- A low-complexity feature extraction technique with a 16 channels filter bank with a quality factor of 1.3 was used to compute the power spectral density.

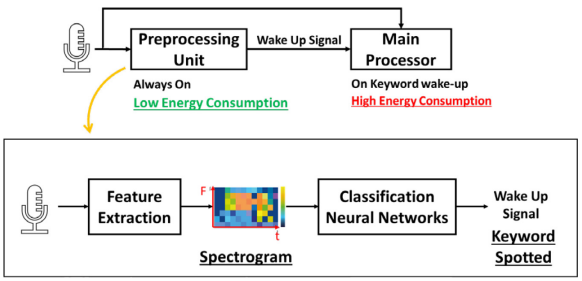

Fig. 1. Typical architecture of a preprocessing unit

- An associated optimized LSTM model with 64 hidden units post-quantized on 8 bits for inputs and 9 bits for weight/activation achieving 89.45% accuracy on recognition of 12 classes of the Google Speech Command Dataset (GSCD) [4].

The remainder of this article is structured as follows. Section II reviews different feature extraction methods and introduces the proposed filter bank together with simulations using Matlab®. Section III presents the LSTM neural network and the method to quantize it. Section IV explores the results in comparison with the state-of-the-art circuits before section V concludes this paper.

II. Feature Extraction

The GSCD is used as a reference for keyword spotting applications. It is composed of 60,000 audio files of approximately 1-second length with recordings of 31 different keywords. A common test case for comparison is to train networks using 12 classes (10 selected keywords + unknown words + silence).

A. Impact of feature extraction on the global consumption

To extract features from this dataset, state-of-the-art solutions [2], [3], [5], [6] use MFCC features. The MFCC is computed in this order: (i) FFT of an audio sample window,

TABLE I. Contribution of FE in state-of-the-art circuits

Reference	[2]	[3]	[5]	[6]
Embedded FE	FE on software	FE on software	Real FFT - MFCC	Serial FFT - MFCC
Global Consumption (µW)	5	0.5[a]	16.1	0.51
Percentage of global consumption due to FE	- (Soft.)	- (Soft.)	50%	66%

[a] Estimation from available metrics

This work was supported in part by the French National Research Agency under Grant ANR-18-CE24-0006-01 LEOPAR.

(ii) Mel filtering using a digital high-order filter bank, (iii) computation of the log of the power for each filter output, and (iv) DCT of each computed value. The MFCCs are extracted as the amplitudes of the output spectrum. We can analyze from the literature (Table I) that the contribution of the feature extraction is more than half of the global consumption of the classification system. In [5], the authors report that the computational power is dominated by the FFT, which accounts for 72% of the total number of sums and multiplications of the FE block. To reduce the computationally expensive MFCC extraction, [7] presents a 32-channel analog filter bank employing a passive N-path filter topology consuming 800nW, while [8] introduces an event-driven approach, in which the system simulations show up to a 4000x lower consumption compared to a conventional discrete-time system.

TABLE II. FILTER BANK CONFIGURATION

Number of bands	16
Filter order	3
Frequency range	50 Hz to 5 kHz
Q	1.3

B. *Proposed feature extraction architecture*

The proposed FE architecture is composed of a 16-channel filter bank, with center frequencies spread from 50 Hz to 5 kHz. The quality factor Q is only 1.3, making this solution easily realizable in analog or mixed-signal domains. Table II presents the configuration of our filter bank and Figure 2 presents the frequency response of the filter bank according to Mel, Bark, and Logarithmic scales. Mel and Bark scales are perceptual scales and are created from how humans hear. Classification results using these scales are compared in section V.

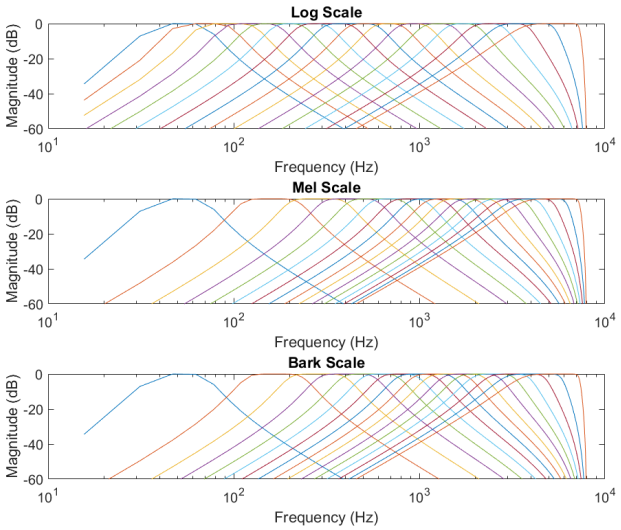

Fig. 2. 16-channel filter bank frequency response according to different scales. Mel and Bark scales are perceptual scales, used to mimic human hearing. There is less frequency bands under 100 Hz with these scales.

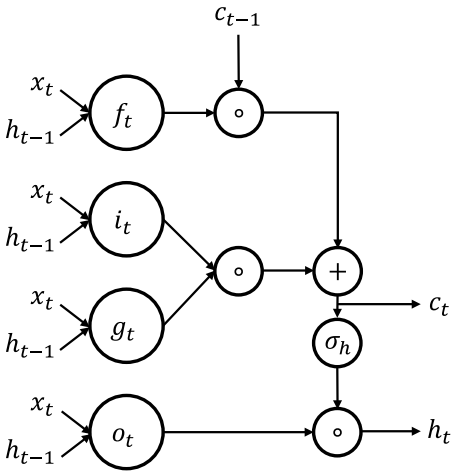

Fig. 3. Schematics of a Long Short-Term Memory neural network

When the spectral signal is divided into 16 bands, the energy in each band is calculated as:

$$E = \sum_{t=t_0}^{t_0+dt} |y(t)^2| \tag{1}$$

with $y(t)$ the filtered audio signal and *dt* the frame duration (set to 25 ms with an overlap of 12.5 ms). The filter bank is simulated in Matlab using Butterworth filters. There is no logarithmic scaling on the output data, meaning that when the energy is extracted from each band, it can directly be converted using an ADC and sent to the classifier.

III. CLASSIFICATION NEURAL NETWORK

A. *Long Short-Term Memory*

LSTM networks are a type of recurrent neural network composed of 4 intermediate sets of neurons called gates (input i_t , forget f_t, output o_t, candidate gate g_t) (2)-(5) that will compute a state vector c_t (6) that in turn is used to compute the hidden vector h_t (7). This vector will be used in the next inference together with the next input vector x_t.

$$f_t = \sigma_s(W_f x_t + R_f h_{t-1} + b_f) \tag{2}$$

$$i_t = \sigma_s(W_i x_t + R_i h_{t-1} + b_i) \tag{3}$$

$$o_t = \sigma_s(W_o x_t + R_o h_{t-1} + b_f) \tag{4}$$

$$g_t = \sigma_h(W_g x_t + R_g h_{t-1} + b_f) \tag{5}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ g_t \tag{6}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{7}$$

where W_* and R_* are weight matrices for each gate and b_* biasing values that are obtained by training the neural networks. Figure 3 shows a schematic of the LSTM architecture. The length of the states and hidden unit vectors allows storing information in time at each loop, meaning that the LSTM has the faculty to remember what just happened and therefore modify its outputs knowing this information, making this type of network a good choice for sequential tasks.

LSTM models can be stacked and are followed by one or more fully connected layers:

$$z_t = W_{fc} h_t \quad (8)$$

where W_{fc} is a weight matrix for the fully connected layer. A softmax layer is added at the end to perform the prediction as can be seen in Figure 4. This network is trained using Stochastic Gradient Descent algorithms.

B. Post-Quantization

Post-quantization techniques are introduced to perform an n-bit quantization of the LSTM model. The custom LSTM layers are described in Matlab® and are initialized with the weights obtained from the full-precision training, to accelerate the convergence. At each forward propagation, the results of equations (2)-(7) are quantized using equation (9).

$$quant(x) = round\left(\frac{x}{\max(x)} \times 2^{n-1} - 1\right) \times \frac{\max(x)}{2^{n-1} - 1} \quad (9)$$

where n is the number of quantization bits. Figure 5 shows that the weights and activation vectors follow a normal distribution. Therefore, to improve the internal representation of the system, clipping can be introduced with little impact. The introduced method consists of performing iterations with decreasing clipping values until the accuracy drops by a given amount. The clipping value associated with the maximum accuracy is eventually selected. This simple and effective method is suited to a low-complexity network that allows exploring several parameters over multiple iterations. However, for larger and computationally-intensive networks, more efficient in-training quantization methods such as PACT [9] have been developed.

IV. Simulation Results

To compose the dataset, 10 keywords are chosen from the GSCD: {"zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"} (around 1800 samples of each word) plus 20% of other keywords from the dataset labeled "unknown" and 4,000 samples of background noise. The selected dataset is shared between training, validation, and testing datasets following this repartition: 70%, 15%, and 15%. The simulations are made with an LSTM composed of 64 hidden units. Using the computed spectrograms from the FE block as described in section II, multiple simulations are run to observe the impact of the number of bands and the input bit width on the accuracy. The per-band energy values are quantized on n bits and are then trained with the basic *LstmLayer* model from Matlab® with weights and activation coded on 32 bits. Figure 6 recapitulates those simulations and shows that there is no particular scale that stands out and would give better results than others. However, filter banks with more than 12 frequency bands offer better results than the 8 band case. An input bit width below 8 bits significantly decreases the accuracy. The best accuracy obtained for the test dataset is 90.02% for a 16-channel filter bank using a logarithmic scale with input data quantized on 8 bits. This setup is taken as a reference point for the development of the quantized LSTM model.

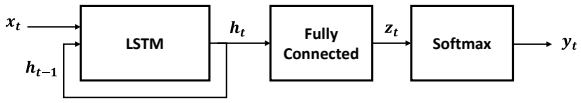

Fig. 4. Schematics of the neural network used in this paper.

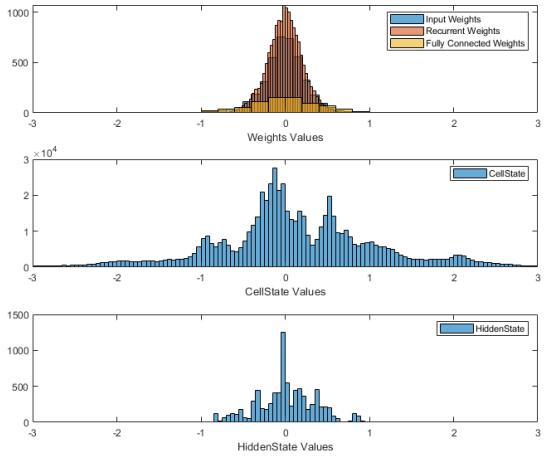

Fig. 5. Histograms showing the weights and activation functions distributions.

TABLE III. Comparison with SoA

Reference	[2]	[3]	[5]	Our Work
Feature Extraction	MFCC	MFCC	MFCC	Power Spectrum
FFT	Yes (soft.)	Yes (soft.)	DFT	No FFT
DCT	Yes (soft.)	Yes (soft.)	Yes	No DCT
Number of channels	39	40	13	16
Quantization Method	Post Quantization	In-training	Post Quantization	Post Quantization
Hidden Units	56	128	64	64
Inputs Bit width	8	5	8	8
Weights Bit width	8	5	8	9
Activation Bit width	32	8	8	9
Number of classes	4	12	12	12
Dataset	TIMIT [b]: 4 KW	GSCD: 10KW + unknown + silence	GSCD: 10KW + unknown + silence	GSCD: 10KW + unknown + silence
Accuracy	91.7%	90%	90.87%	89,45%

[b] Texas Instrument Massachusetts Institute of Technology dataset [10]

A custom LSTM model has been developed to explore the impact of quantization. Simulating our custom LSTM model with the previous setup gives a similar reference accuracy value. The code of our model and the simulation methods are available on GitHub [1]. The model allows quantizing the network to n bits using the proposed post-quantization method. The simulated accuracy is shown in Figure 7. The

[1] https://github.com/kevinherisse/leo_lstm

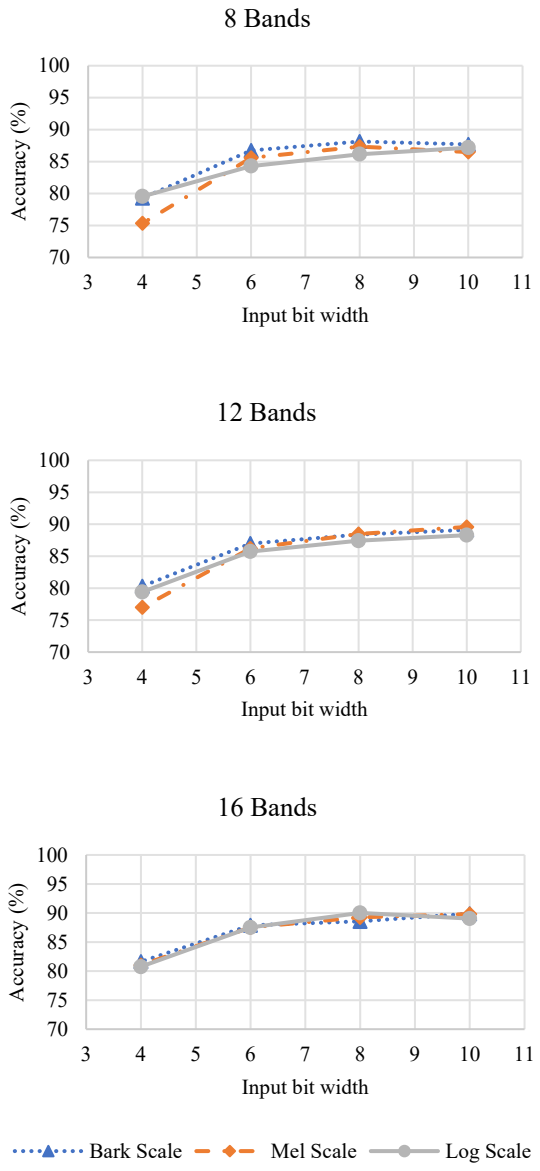

Fig. 6. Accuracy of the full-resolution network as a function of the input bit width according to different number of bands. The values are extracted as the best accuracy found over multiple trainings. The best value obtained during training is 90.02% with 16-channel and 8-bit input quantization.

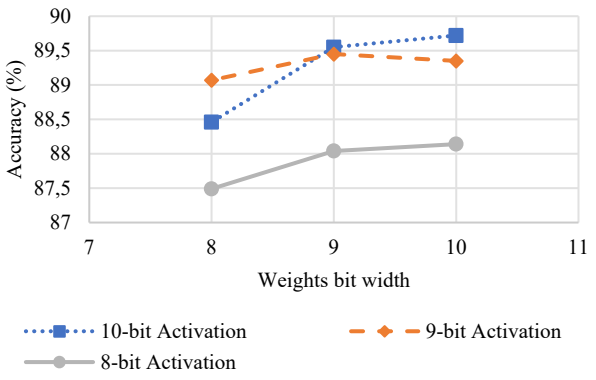

Fig. 7. Accuracy versus quantization weight bit width for multiple activation quantization

quantization only implies an accuracy drop of 0.55% for 8-bit input and 9-bit activation/weight.

Table III shows a comparison with state-of-the-art approaches. Similar accuracy is obtained while the feature extraction method is much less complex than the MFCC computation. The presented method uses a 16-channel filter bank, with third-order filters and a quality factor of 1.3 that could be implemented with low-consumption techniques such as [7] or [8].

V. CONCLUSION

This paper shows that it is possible to extract relevant audio features with a simple FE block composed of a 16-channel third order filter bank. Using a quantized 64-hidden unit LSTM model, an accuracy of 89.45% on 12 classes of the GSCD is demonstrated. These results open significant perspectives on reducing the hardware complexity of the FE function. Future work will concern the implementation of the complete processing chain and measurement of the impact on energy consumption.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[2] J. S. P. Giraldo and M. Verhelst, “Laika: A 5uW Programmable LSTM Accelerator for Always-on Keyword Spotting in 65nm CMOS,” in *ESSCIRC 2018 - IEEE 44th European Solid State Circuits Conference (ESSCIRC)*, Dresden, Sep. 2018, pp. 166–169. doi: 10.1109/ESSCIRC.2018.8494342.

[3] C. J. Schaefer, M. Horeni, P. Taheri, and S. Joshi, “LSTMs for Keyword Spotting with ReRAM-based Compute-In-Memory Architectures,” in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, Daegu, Korea (South), May 2021, pp. 1–5. doi: 10.1109/ISCAS51556.2021.9401295.

[4] P. Warden, “Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition,” *ArXiv180403209 Cs*, Apr. 2018, Accessed: May 05, 2021. [Online]. Available: http://arxiv.org/abs/1804.03209

[5] J. S. P. Giraldo, S. Lauwereins, K. Badami, and M. Verhelst, “Vocell: A 65-nm Speech-Triggered Wake-Up SoC for 10-μ W Keyword Spotting and Speaker Verification,” *IEEE J. Solid-State Circuits*, vol. 55, no. 4, pp. 868–878, Apr. 2020, doi: 10.1109/JSSC.2020.2968800.

[6] W. Shan *et al.*, “A 510-nW Wake-Up Keyword-Spotting Chip Using Serial-FFT-Based MFCC and Binarized Depthwise Separable CNN in 28-nm CMOS,” *IEEE J. Solid-State Circuits*, vol. 56, no. 1, pp. 151–164, Jan. 2021, doi: 10.1109/JSSC.2020.3029097.

[7] D. A. Villamizar, D. G. Muratore, J. B. Wieser, and B. Murmann, “An 800 nW Switched-Capacitor Feature Extraction Filterbank for Sound Classification,” *IEEE Trans. Circuits Syst. Regul. Pap.*, vol. 68, no. 4, pp. 1578–1588, Apr. 2021, doi: 10.1109/TCSI.2020.3047035.

[8] S. Mourrane, B. Larras, A. Cathelin, and A. Frappe, “Event-Driven Continuous-Time Feature Extraction for Ultra Low-Power Audio Keyword Spotting,” in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, Washington DC, DC, USA, Jun. 2021, pp. 1–4. doi: 10.1109/AICAS51828.2021.9458425.

[9] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, “PACT: Parameterized Clipping Activation for Quantized Neural Networks,” *ArXiv180506085 Cs*, Jul. 2018, Accessed: Jul. 29, 2021. [Online]. Available: http://arxiv.org/abs/1805.06085

[10] *TIMIT: acoustic-phonetic continuous speech corpus.* Philadelphia, Pa.: Linguistic Data Consortium, 1993.