Analog LSTM for Keyword Spotting

Kofi Odame, Maria Nyamukuru
Dartmouth College, 15 Thayer Drive, Hanover NH, 03755

***Abstract*— In this paper, we present a novel analog application-specific integrated circuit (ASIC) architecture for edge artificial intelligence. Our novel architecture is a power efficient long short-term memory, which is a deep learning neural network that is especially suited for processing time-series sensor data. Evaluated on a 10-class keyword spotting machine learning task, our neural network achieves a classification accuracy of 91 %, with a model size of 2264 parameters and estimated power consumption of 0.76 μW.**

I. Introduction

Self-powered edge artificial intelligence (AI) is a key enabling technology for the billions of sensors that will potentially be deployed worldwide as part of the internet-of-things (IoT). One class of edge AI that is well-suited for IoT sensors is analog long short-term memory (LSTM) [1]. But even for moderately complex applications like keyword spotting, analog LSTM consumes more power than is feasible for a self-powered solution (see Fig. 1).

To address this problem, we introduce a new, energy-efficient analog LSTM for keyword spotting. We achieve energy efficiency by using (1) fewer inputs and (2) fewer operations than the state-of-the-art (Fig. 2). Our approach can be used in addition to conventional power reduction strategies like compute-in-memory, weight quantization or knowledge distillation.

II. Pyknogram: Fewer Inputs

The first processing stage in keyword spotting is generally spectral analysis. This analysis produces dozens of input *spectrogram features*, a comprehensive, but sparse, time-frequency representation of the audio signal. Because they are a sparse representation, many spectrogram features contain little useful information, resulting in unnecessary processing and power consumption.

To improve power efficiency, we have developed the pyknogram filter, which produces a dense, low-dimensional version of a spectrogram. Figure 3(a) (top panel) shows one channel of the pyknogram filter. First, a bandlimited signal, V_u, is input to an adaptive bandpass filter. Then, the adaptive filter attempts to minimize the error between its input and output V_w by adjusting its center frequency, f_c; the error is minimized when f_c is tracking the input signal's most energetic frequency region. Thus, the center frequency, f_c, is the output "feature" of the pyknogram analysis. For voice inputs,

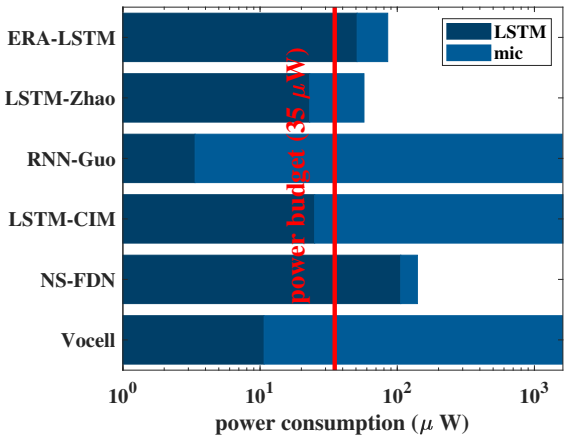

Fig. 1: Power consumption exceeds budget for a self-powered wearable sensor comprising a 69 dB(A) SNR microphone ('mic') and LSTM chip ('LSTM'). Digital LSTM consumes less power, but requires a 1.6 mW digital microphone [2]. Analog LSTM consumes more power, but requires a 34 μW analog microphone [3]. Both strategies exceed the 35 μW harvested power budget [4, 5]. Digital LSTM chips: LSTM-CIM [6], Vocell [7], RNN-Guo [8]. Analog LSTM chips: ERA-LSTM [9], LSTM-Zhao [10], NS-FDN [11]. LSTM power is normalized to 10 kHz.

this corresponds to the speech formant (see Fig. 3(b)). Figure 3(a) (bottom panel) shows a schematic of the pyknogram circuit. Here, nmos M_x is in the subthreshold regime with a drain current, I_x, given by [13]

$$C_x \frac{dI_x}{dt} = \frac{\kappa G_m}{U_\mathrm{T}} \cdot (V_u - V_w) \cdot \mathrm{sgn}\,(V_y) \cdot I_x, \quad (1)$$

where κ is the body-effect coefficient and U_T is the thermal voltage. I_x is also the bias current that controls the gain of the transconductance amplifiers, and hence the filter's center frequency [14]. So, Eqn. (1) continually adjusts I_x—and hence the center frequency—to minimize the $(V_u - V_w)$ error. That is, I_x tracks the most energetic frequency region in the input. The simulation results of Fig. 3(b) illustrate this behavior, with I_x tracking the formants of a speech signal.

III. AFUA: Fewer Operations

Fundamentally, an LSTM is a neuron that selectively retains, updates or erases its memory of input data [1]. The gated recurrent unit (GRU) is a simplified version

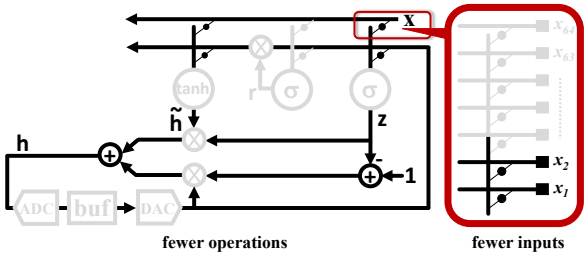

Fig. 2: Our approach cuts power consumption by by processing fewer inputs and by avoiding several operations (greyed out blocks) of a standard LSTM [12]. We replace the *tanh* and *sigmoid* functions with zero-cost *softplus* operations.

of the classical LSTM, and it is described with the following set of equations [12]:

$$r_j = \sigma([\mathbf{W}_r\mathbf{x}]_j + [\mathbf{U}_r\mathbf{h}_{\langle t-1\rangle}]_j) \quad (2)$$

$$z_j = \sigma([\mathbf{W}_z\mathbf{x}]_j + [\mathbf{U}_z\mathbf{h}_{\langle t-1\rangle}]_j) \quad (3)$$

$$\tilde{h}_j^{\langle t\rangle} = \tanh([\mathbf{W}\mathbf{x}]_j + [\mathbf{U}(\mathbf{r}\odot\mathbf{h}_{\langle t-1\rangle})]_j) \quad (4)$$

$$h_j^{\langle t\rangle} = z_j h_j^{\langle t-1\rangle} + (1-z_j)\tilde{h}_j^{\langle t\rangle}, \quad (5)$$

where $\mathbf{x}$ is the input, h_j is the hidden state, $\tilde{h}_j$ is the candidate state, r_j is the reset gate and z_j is the update gate. Also, $\mathbf{W}_*$ and $\mathbf{U}_*$ are learnable weight matrices.

We can replace the GRU with a simpler set of equations via the following manipulations. The *sigmoid* function of Eqn. (3) gives z_j a range of $(0,1)$, and the extrema of this range reveals the basic mechanism of the update equation, Eqn. (5). For $z_j = 0$, the update equation is $h_j^{\langle t\rangle} = \tilde{h}_j^{\langle t\rangle}$. For $z_j = 1$, the update equation becomes $h_j^{\langle t\rangle} = h_j^{\langle t-1\rangle}$. Without loss of generality, we can replace $(1-z_j)$ with z_j (this merely inverts the logic of the update gate). So, replacing $(1-z_j)$ and rearranging the update equation gives us

$$\left(h_j^{\langle t\rangle} - h_j^{\langle t-1\rangle}\right)/z_j + h_j^{\langle t-1\rangle} = \tilde{h}_j^{\langle t\rangle}, \quad (6)$$

which is simply a first-order low pass filter with a continuous-time form of

$$\frac{\tau}{z_j(t)}\frac{dh_j}{dt} + h_j(t) = \tilde{h}_j(t), \quad (7)$$

where $\tau = \Delta T$, the time step of the discrete-time system. The gating mechanics of the continuous- versus discrete-time update equations are equivalent, modulo the inverted logic: For $z_j(t) = 0$, Eqn. (7) is a low-pass filter with an infinitely large time constant, and $h_j(t)$ does not change (this is equivalent to $h_j^{\langle t\rangle} = h_j^{\langle t-1\rangle}$ in discrete time). For $z_j(t) = 1$, Eqn. (7) is a low-pass filter with a time constant of $\tau = \Delta T$. Since the ΔT time step is small relative to the GRU's dynamics, a time constant of $\tau = \Delta T$ produces $h_j(t) \approx \tilde{h}_j(t)$ (equivalent to $h_j^{\langle t\rangle} = \tilde{h}_j^{\langle t\rangle}$ in discrete time).

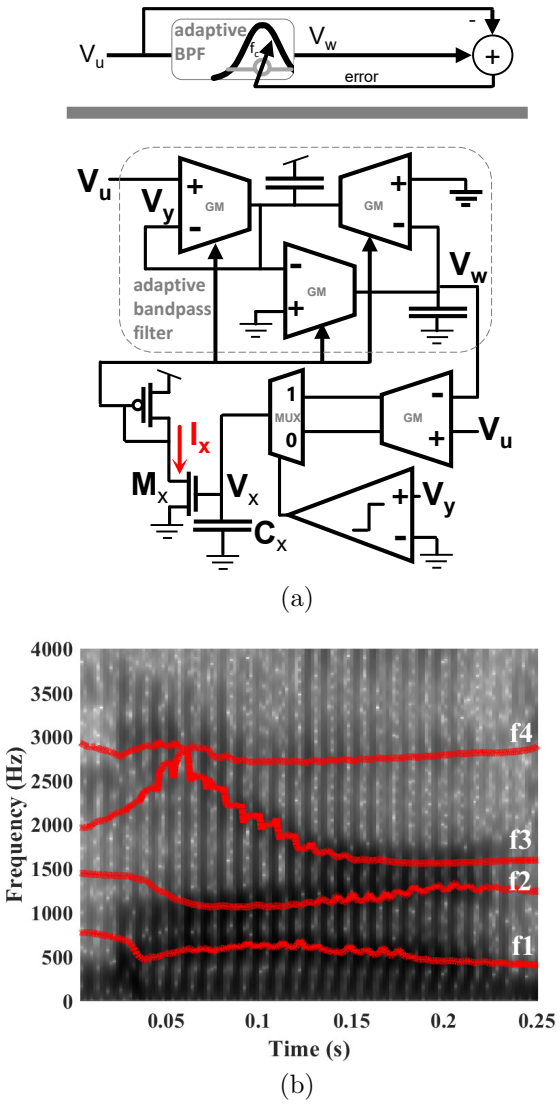

Fig. 3: (a) Pyknogram filter single channel. (b) Output of 4-channel pyknogram filter (red lines) overlaid on spectrogram.

Various studies have found the reset gate unnecessary with slow-changing signals, and also for event detection [15]. Both these scenarios describe our keyword spotting application, so we can discard the reset gate.

Finally, if we translate the origins of both $h_j(t)$ and $\tilde{h}_j(t)$ to 1, then we can replace the $tanh$ with a saturating function that has a range of $(0,2)$. Replacing $tanh$ with $min(softplus(\cdot),2)$, translating the origin and discarding the reset gate, we arrive at the *Adaptive Filter Unit for Analog LSTM* (AFUA):

$$z_j(t) = \min(\text{soft}([\mathbf{W}_z\mathbf{x}]_j + [\mathbf{U}_z(\mathbf{h}(t)-\mathbf{1})]_j), 1)$$

$$\tilde{h}_j(t) = \min(\text{soft}([\mathbf{W}\mathbf{x}]_j + [\mathbf{U}(\mathbf{h}(t)-\mathbf{1})]_j), 2)$$

$$\frac{\tau}{z_j(t)}\frac{dh_j}{dt} = \tilde{h}_j(t) - h_j(t), \quad (8)$$

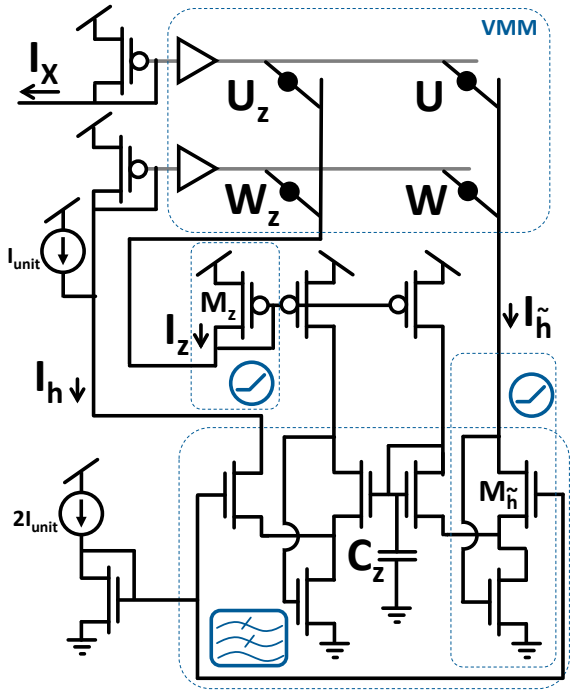

Fig. 4: Circuit implementation of the AFUA unit. Equation (11) is implemented as a current-mode adaptive low pass filter. The *softplus* is implemented with diode-connected transistors, whose outputs clip due to the circuit's finite power supply voltages. The vector matrix multiplier ('VMM') is implemented using conventional circuitry [17].

where we have also replaced the $sigmoid$ with a clipped $softplus$. The benefit of the $softplus$ is that we can implement it as a diode-connected transistor, which costs minimal area and zero power (see e.g. M_z in Fig. 4).

The AFUA is a type of continuous-time GRU. But while the GRU contains 3 Hadamard multiplications, the AFUA contains none. The AFUA has no reset gate. Finally, as Fig. 4 shows, the AFUA avoids the overhead of operational amplifiers, current/voltage converters and internal digital/analog converters found in other analog LSTM implementations [11, 9, 10]. Fewer operations and smaller overhead means less power consumption.

IV. AFUA Circuit Implementation

Figure 4 shows a schematic of the AFUA circuit implementation. The drain currents of M_z and $M_{\tilde{h}}$ are

$$I_z = \min(\text{soft}([\mathbf{W}_z\mathbf{I_x}] + [\mathbf{U}_z(\mathbf{I_h} - \mathbf{I}_{\text{unit}})]), I_{\text{unit}}), \quad (9)$$

$$I_{\tilde{h}} = \min(\text{soft}([\mathbf{W}\mathbf{I_x}] + [\mathbf{U}(\mathbf{I_h} - \mathbf{I}_{\text{unit}})]), 2I_{\text{unit}}). \quad (10)$$

Also, from the translinear loop principle, current I_h is defined by [18, 19]:

$$\underbrace{\frac{C_z U_{\text{T}}}{\kappa I_{\text{unit}}}}_{\tau} \frac{I_{\text{unit}}}{I_z} \frac{dI_h}{dt} = I_{\tilde{h}} - I_h. \quad (11)$$

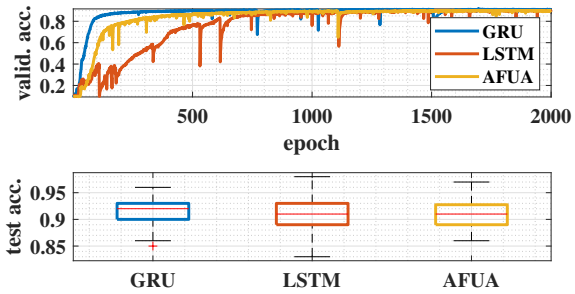

Fig. 5: Training curves and test set performance for keyword spotting based on Google Speech Commands dataset [16].

Eqns. (9), (10) and (11) are the current-mode representations of the AFUA. When I_z is close to zero, the hidden state (represented by I_h in Eqn. (11)) changes slowly, and the AFUA is able to retain long-term memory. When I_z is large, the state's rate of change increases, and the AFUA replaces its memory with newly-arrived information via $I_{\tilde{h}}$ in Eqn. (11).

V. Results and Conclusion

We evaluated the AFUA against the GRU and the classical LSTM in a 10-word (spoken digits) keyword spotting task based on the Google Speech Commands dataset [16]. We implemented the following network architecture in Tensorflow: two 16-unit recurrent layers (AFUA, GRU, or LSTM cells); 10-unit linear layer with ReLU; 10-unit softmax layer. The train/validation/test data split was 53/17/30. We used the ADAM optimizer, mini-batch sizes of 1024 and trained for 2000 epochs.

The Fig. 5 results show that AFUA performs similarly to GRU and LSTM. The AFUA's advantage is evident from Table I: with fewer inputs and fewer operations per neuron, the AFUA's computational complexity is up to 100× less than that of other approaches. Also, the AFUA is a smaller model, as illustrated by its parameter count (Fig. 6). Finally, the AFUA implemented in a 0.18 μm 1.8 V process with $I_{\text{unit}} = 0.2$ nA would consume 0.76 μW. This power consumption meets the constraints of a self-powered sensor.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[2] Infineon. *IM69D130V01XTSA1 Datasheet*. Dec. 2019.

[3] Knowles. *MM20-33639-B116 Datasheet*. Mar. 2018.

[4] Shad Roundy, Paul K Wright, and Jan Rabaey. "A study of low level vibrations as a power source for wireless sensor nodes". In: *Computer communications* 26.11 (2003), pp. 1131–1144.

Model	Neuron Type		Inputs, n	Neurons, m	Ops/Neuron	Total Ops
This work	AFUA	layer 1	10	16	53 $[2(n+m)+1]$	**1,888**
		layer 2	16	16	65 $[2(n+m)+1]$	
RNN-Guo [8]	RNN		256	256	770 $[n+2m+2]$	197,120
Vocell [7]	LSTM		39	64	610 $[4(n+m)+3m+6]$	39,040
LSTM-CIM [6]	LSTM		40	128	1062 $[4(n+m)+3m+6]$	135,936

TABLE I: Total number of operations in the recurrent layer of this work versus state-of-the-art neural network models for 10-word keyword spotting [16]. Softplus, addition and subtraction are zero-cost analog operations.

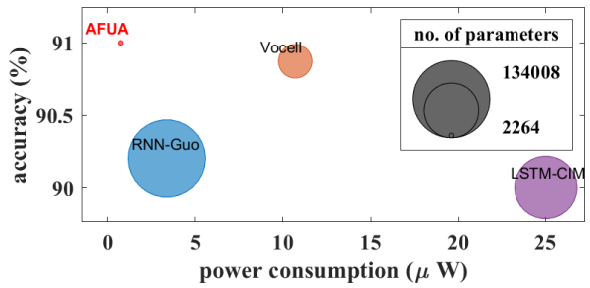

Fig. 6: Power consumption, accuracy and number of parameters for the neural network models listed in Table I. AFUA power consumption is calculated for a 0.18 μm 1.8 V process with $I_{\text{unit}} = 0.2$ nA. If combined with a 34 μW analog microphone [3], AFUA meets the Fig. 1 power budget.

[5] Paul D Mitcheson et al. "Energy harvesting from human and machine motion for wireless electronic devices". In: *Proceedings of the IEEE* 96.9 (2008), pp. 1457–1486.

[6] Clemens JS Schaefer et al. "LSTMs for Keyword Spotting with ReRAM-based Compute-In-Memory Architectures". In: *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2021, pp. 1–5.

[7] Juan Sebastian P Giraldo et al. "Vocell: A 65-nm Speech-Triggered Wake-Up SoC for 10-uW Keyword Spotting and Speaker Verification". In: *IEEE Journal of Solid-State Circuits* 55.4 (2020), pp. 868–878.

[8] Ruiqi Guo et al. "A 5.1 pJ/neuron 127.3 us/inference RNN-based speech recognition processor using 16 computing-in-memory SRAM macros in 65nm CMOS". In: *2019 Symposium on VLSI Circuits*. IEEE. 2019, pp. C120–C121.

[9] Jianhui Han et al. "ERA-LSTM: An efficient ReRAM-based architecture for long short-term memory". In: *IEEE Transactions on Parallel and Distributed Systems* 31.6 (2019), pp. 1328–1342.

[10] Zhou Zhao et al. "Long short-term memory network design for analog computing". In: *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 15.1 (2019), pp. 1–27.

[11] Qin Li et al. "NS-FDN: Near-Sensor Processing Architecture of Feature-Configurable Distributed Network for Beyond-Real-Time Always-on Keyword Spotting". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 68.5 (2021), pp. 1892–1905.

[12] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[13] Christian C. Enz, François Krummenacher, and Eric A. Vittoz. "An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications". In: *Analog Integr. Circuits Signal Process.* 8.1 (1995), pp. 83–114.

[14] Arun Rao and Kofi Odame. "Estimating the short-time bandwidth of wheeze sounds". In: *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE. 2015, pp. 1–4.

[15] J Amoh and K Odame. "An optimized recurrent unit for ultra-low-power keyword spotting". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.2 (2019), pp. 1–17.

[16] Pete Warden. "Speech commands: A dataset for limited-vocabulary speech recognition". In: *arXiv preprint arXiv:1804.03209* (2018).

[17] Jonathan Binas et al. "Precise deep neural network computation on imprecise low-power analog hardware". In: *arXiv preprint arXiv:1606.07786* (2016).

[18] J Mulder et al. "Dynamic translinear RMS-DC converter". In: *Electronics letters* 32.22 (1996), pp. 2067–2068.

[19] K Odame and B Minch. "The translinear principle: a general framework for implementing chaotic oscillators". In: *Int'l Journal of Bifurcation and Chaos* 15.08 (2005), pp. 2559–2568.