LSTMs for Keyword Spotting with ReRAM-based Compute-In-Memory Architectures

Clemens JS Schaefer, Mark Horeni, Pooria Taheri, and Siddharth Joshi
Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN, USA
Email: {cschaef6, mhoreni, ptaheri, sjoshi2}@nd.edu

2021 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-7281-9201-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/ISCAS51556.2021.9401295

***Abstract*—The increasingly central role of speech based human computer interaction necessitates on-device, low-latency, low-power, high-accuracy key word spotting (KWS). State-of-the-art accuracies on speech-related tasks have been achieved by long short-term memory (LSTM) neural network (NN) models. Such models are typically computationally intensive because of their heavy use of Matrix vector multiplication (MVM) operations. Compute-in-Memory (CIM) architectures, while well suited to MVM operations, have not seen widespread adoption for LSTMs. In this paper we adapt resistive random access memory based CIM architectures for KWS using LSTMs. We find that a hybrid system composed of CIM cores and digital cores achieves 90% test accuracy on the google speech data set at the cost of 25 uJ/decision. Our optimized architecture uses 5-bit inputs, and analog weights to produce 6-bit outputs. All digital computation are performed with 8-bit precision leading to a $3.7\times$ improvement in computational efficiency compared to equivalent digital systems at that accuracy.**

I. Introduction

With the spread of "smart" virtual assistants and mobile devices, speech-based human computer interaction has become increasingly prevalent. For such speech-based systems to become commonplace in mobile, energy-constrained devices, the friction of interacting with them must be minimized. Consequently, they must be extremely responsive and highly accurate. One way of ensuring responsiveness is to implement the "smarts" on the device. Accuracy, in turn, depends on the "smart" algorithms. This creates a tension between implementing complex, energy-intensive algorithms for accuracy and implementing lightweight, hardware-friendly algorithms on device. This work, analyzes these trade-offs and proposes a hardware-friendly algorithm for keyword spotting (KWS) to enable always-on voice-based interactive systems. Keyword Spotting is a central part of the speech processing pipeline for energy-constrained, always-on devices. Thus, hardware that implements energy-efficient, on-device KWS is a must for bringing voice-interactivity to the edge [1].

Long short-term memory (LSTM) neural networks (NNs), are state-of-the-art autoregressive networks used as primitives in many state of the art NN models, e.g., speech synthesis [2], speech recognition [3], and KWS [4]. While accurate, LSTMs are energy intensive because they employ a large number of parameters in comparison to fully-connected (FC) layers and consequently incur a large number of matrix-vector multiplications (MVMs) [5]. Specialized hardware has previously been employed to reduce the energy consumption of LSTMs, especially targeting KWS tasks [6]–[8].

Previous research has demonstrated that compute-in-memory (CIM) architectures can deliver energy-savings of up to $68\times$ for certain deep neural network (DNN) models such as convolutional neural networks (CNNs) [9], [10]. Thus, it is natural to ask: can a mixed-signal, CIM-based system capable of implementing energy-efficient MVMs improve the energy-efficiency of KWS systems while still operating at low latency? This is especially important since state-of-the-art KWS implementations leverage LSTMs [1], [6], [11]. CIM architectures implementing a recurrent attention model have been employed for KWS tasks [12] and CIM architectures employing resistive RAM (ReRAM) elements have implemented binarized LSTMs [5]. However, to the best of our knowledge, implementing LSTM-based KWS using CIM architectures in an energy-efficient fashion has remained elusive.

This work presents a hardware-aware approach to training LSTMs for the KWS task. We partition LSTM operations across a heterogeneous architecture that employs ReRAM-based CIM cores together with digital processing elements (PEs). The contributions of this paper are three-fold: i) We describe a novel CIM-friendly LSTM structure together with a training method for low-precision operation;[1] ii) we compare the accuracy and energy-efficiency of this structure on our proposed hybrid system against conventional implementations; and iii) we study the effect of various parameters on the energy consumption of our proposed hybrid system.

II. Background

This work integrates low-precision LSTM model design and CIM-based system design. In comparison to a digital implementation of KWS, our system offers improved energy-efficiency for competitive accuracy in KWS. These improvements are achieved by trading-off CIM energy-efficiency against the resilience achieved through increased redundancy.

Applying LSTMs to the KWS task requires extracting informative features from raw audio signals, followed by decision-making on these features. In this work, we build upon microcontroller-friendly LSTMs [13]. First, Mel-frequency

This work was supported in part by ASCENT, one of six centers in JUMP, a SRC program sponsored by DARPA. The work was also supported in part by NSF/Intel Partnership on Machine Learning for Wireless Networking Systems under grant CNS-2002921

[1]PyTorch code and models available under https://github.com/Intelligent-Microsystems-Lab/QuantizedLSTM

978-1-7281-9201-7/21/$31.00 ©2021 IEEE

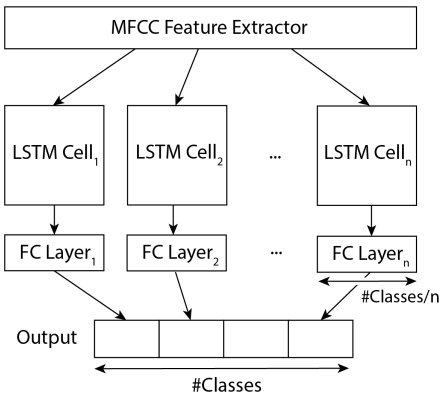


Fig. 1. System-level block diagram of our proposed compute-in-memory (CIM) based keyword spotting system. Mel-frequency cepstral coefficient (MFCC) feature extraction provides inputs to the multiple parallel long short-term memory (LSTM) cells. The outputs of these cells are concatenated into one vector which is then evaluated against a one-hot encoded target vector.

cepstral coefficients (MFCC) are extracted from the raw audio signals over a window of 40.0625 ms with a stride of 20 ms. These MFCC features are then fed to a NN model composed of LSTM cells and FC layers. The KWS system processes the features to decide whether an input audio snippet included a specific set of key words. This work focuses on the LSTM cells and subsequent FC layers, assuming that the MFCC features are extracted and provided as inputs.

A. Long short-term memory (LSTM)

The core of our system is the LSTM cell described by [14]:

$$\begin{aligned} i_t &= \text{sigmoid}(W^i x_t + U^i h_{t-1}), \\ f_t &= \text{sigmoid}(W^f x_t + U^f h_{t-1}), \\ o_t &= \text{sigmoid}(W^o x_t + U^o h_{t-1}), \\ \tilde{c}_t &= \tanh(W^c x_t + U^c h_{t-1}), \\ c_t &= i_t \odot \tilde{c}_t + f_t + \tilde{c}_{t-1}, \\ h_t &= o_t \odot \tanh(c_t). \end{aligned} \tag{1}$$

Here, x is the input, h is the hidden state, c is the memory cell state and t the current time step. Correspondingly, $[W^i, W^f, W^o, W^c, U^i, U^f, U^o, U^c]$ are *learnable* weights, including biases. The current state depends on the previous state through $\tilde{c}_{t-1}$ and h_{t-1} as well as the current inputs x_t. Each LSTM cell is followed by a FC, trainable, layer to map the cell output to a vector the size of the output classes.

B. ReRAM Based Compute-In-Memory (CIM)

Examining (1), each LSTM cell employs eight independent MVMs. LSTMs implemented on a digital accelerator incur significant energy-costs due to data-movement resulting from repeatedly fetching parameters from memory. Consequently, the energy-costs of digital MVMs are dominated by data movement [15]. Thus, architectures that minimize weight movement can be used for energy-efficient implementations of LSTMs. Dense storage of LSTM parameters in emerging nonvolatile resistive RAM arrays, coupled with energy-efficient compute-in-memory architectures address precisely this challenge. In such architectures, the NN model weights are stored as conductances of the memory elements. CIM architectures implement *in-situ* MVM by using digital-to-analog converters (DACs) to drive the bitlines by voltages proportional to the input activation. This induces currents accumulating on the bitlines, where the current is the sum of activation-weight products through Kirchoff's current law. Integrating the current onto a capacitance and reading out the voltage value through analog-to-digital converters (ADCs) results in an effective MVM [16], [17]. CIM architectures can use a range of nonvolatile storage elements, including Ferroelectric devices, metal-oxide based devices, or spin-based devices [18]. This work uses metal oxide-based ReRAM to model the storage device due to the availability of calibrated models from fabricated CIM arrays [5], [17], [19].

C. Quantization

In energy-optimized digital implementations of LSTMs, the LSTM weights, activations, and inputs must all be heavily quantized [13]. Primarily driven by the energy and latency impact of computing at high-precisions. However, the trade-offs are different for CIM based system, for example CIM architectures can employ write-verify schemes to set conductances to arbitrary values with some degree of precision [17], [19]. Thus avoids quantizing weight values in the LSTM, at the cost of weight-noise induced by imprecision in setting the conductance values. Similarly, the use of ADCs to digitize the output of the MVMs results in exponentially more energy expended in digitizing this 'pre-activation'. Thus, reducing ADC precision through coarser quantization of the pre-activation is critical to ensuring CIM energy-efficiency. In this work, we develop a symmetric variation of "PArameterized Clipping acTivation" (PACT) [20]:

$$\begin{aligned} \text{PACT}(x) &= 0.5\,\text{sgn}(x)(|x| - ||x| - \alpha| + \alpha) \\ &= \begin{cases} -\alpha, & x \in (-\infty, -\alpha) \\ x, & x \in [-\alpha, \alpha] \\ \alpha, & x \in (\alpha, +\infty). \end{cases} \end{aligned}$$

PACT learns the clipping parameter α, and taken together with an L_2 regularization of its magnitude maintains LSTM accuracy while aggressively quantizing the 'pre-activation' value. In contrast to the original PACT [20] which replaced a ReLU activation our variation is symmetric and clips negative as well as positive values at the learned threshold α. To quantize clipped values we first scale the values into a range between -1 and 1 and then apply:

$$Q(x, k) = \sigma(k) \cdot \text{round}\left[\frac{x}{\sigma(k)}\right],$$

where $\sigma(x) = 2^{1-x}$ [21]. Subsequently we apply a clipping operation to ensure that the values are within the viable range of $-1 + \sigma(k)$ and $+1 - \sigma(k)$.

D. Regularized Learning

We employ multiple regularization techniques for improved model accuracy. We use an LSTM-specific DropConnect [22]

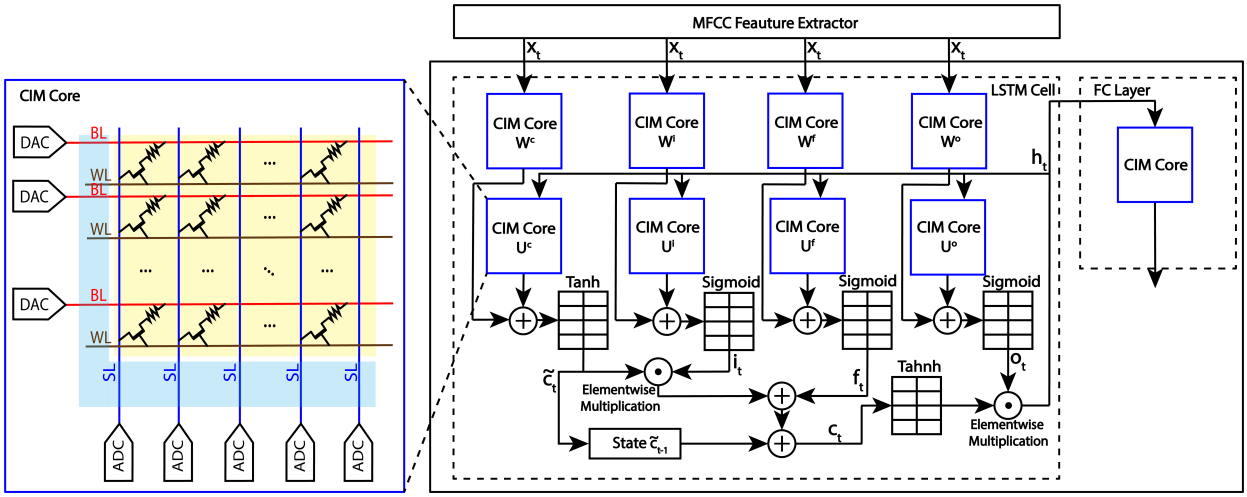


Fig. 2. An illustration of our proposed hybrid architecture using compute-in-memory (CIM) cores and digital processing elements (PEs). The CIM core is expanded on the left, with DACs driving the bitlines and a 1T1R scheme employed per resistive RAM device. The voltages applied on the bit line (BL) are weighed by the individual conductances of each ReRAM element followed by parallel accumulation on to the source line (SL). The resulting charge is digitized by the analog-to-digital converters (ADCs) at the periphery. The output of the ADC is digitally added followed by the application of a nonlinearity. These nonlinearities are applied through a lookup table (stored in a local register file) mapping the bits onto the nonlinear function. This application of the nonlinearity is followed by elementwise multiplication when required.

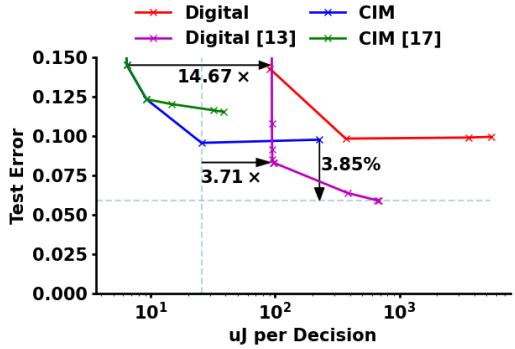


Fig. 3. Energy-accuracy trade-offs for CIM and digital implementations. Each line represents a efficient frontier for energy and performance. Blue and green represent CIM based solution whereby the green frontier is a restricted version of the blue based upon [17]. The red and magenta frontier stand for digital solution, whereby red is a digital implementation of our proposed CIM architecture and magenta an architecture proposed by [13].

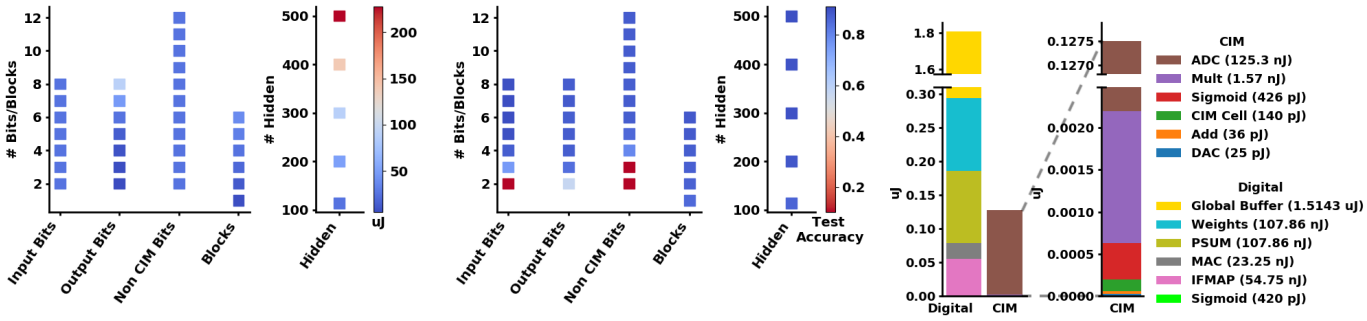
for an initial training phase of $20,000$ epochs (one epoch is $100 - 474$ training examples). This is followed by $10,000$ epochs of training with Gaussian random noise added to the model weights, emulating device noise and mismatch [23]. Characterization of ReRAM weight noise from literature [17], indicates that the standard deviation of the weight noise, σ_{noise}, should be set to 10% of the maximum weight stored in the CIM core. However we found through a grid search that during training weight noise of 16% results in better accuracies on the test set with 10% weight noise during inference.

III. Methods

The externally-digital, internally-analog computational interface provided by CIM-based architectures offer a different set of trade-offs compared to conventional digital systems. Thus, algorithms tailored to digital systems under-perform on CIM architectures — in terms of energy-efficiency and accuracy [24]. In this work we address two limitations of energy-efficient CIM systems. First, the energy-efficiency gained from employing a CIM-based architecture is completely negated when MVMs larger than core-size are required. This is driven by the higher linearity requirements from the ADCs at the CIM core periphery. Additionally, any technique that generates partial sums further incurs the same energy-cost. Second, device-to-device variation as well as noise due to analog computation drastically limits the algorithmic performance of a model on CIM architectures [23], [25].

A. System Architecture

Our proposed solution, shown in Fig. 1, separates the computation of different output classes across smaller, independent LSTM cells. A group of 9 CIM cores of size 128×128 compose one LSTM cell and one FC layer, as shown in Fig. 2. This allows for a set of CIM-tiles to be specialized to certain output classes. Figure 2 shows the internal working of a single CIM core and the placement of CIM cores in the computational flow for LSTM computation. The size of the CIM core, determines the size of the hidden dimension. The nonlinearities like the $\mathrm{sigmoid}$ and $\tanh$ operations as well as the element-wise operations are implemented digitally. For our simulations, we define a default configuration to be one where CIM cores take 4-bit inputs, generate 6-bit outputs, and incur 10% weight-noise. All digital operations are performed with 8-bit accuracy. All energy numbers were obtained using Accelergy [26] and Timeloop [27], evaluated on a generic 32 nm technology node. Accelergy is a compound energy estimation tool, which provides a breakdown of the total system energy into its components.



(a) (b) (c)

Fig. 4. (a) Effect of varying bit-widths of CIM inputs, outputs and non-CIM/digital operations as well as number of blocks and hidden state vector size on energy consumption. Hidden dimensionality has a major effect on the energy consumption as well as the output activation bit-width due to the necessity of increased ADC accuracy. (b) Effect of varying bit-widths of CIM inputs, outputs and non-CIM/digital operations as well as number of blocks and hidden state vector size on test accuracy. Accuracy is most sensitive towards low bit-widths of input activations and digital operations. (c) A comparison of the breakdown of energy expended by different subsystems in the default CIM configuration and the default digital configuration. The energy expenditure of the digital implementation is dominated by the global buffer. Note the order of magnitude difference between CIM and digital solutions. The right hand side is a enlarged version of the CIM bar in left plot highlighting the energy consumption of the ADC compared other components.

B. LSTM Model Evaluation

We evaluated KWS performance on the Google Speech Commands dataset [28]. We trained LSTM networks to detect 10 different keywords as well as silence and unknown utterances from 1 second sound snippets (total of 12 output classes). The data was enriched with 10% background noise during training and random time-shifts of 100 ms. The input data to the LSTM cells consisted of 40 MFCCs as described in II. We selected the final model based on the performance on the validation data and all numbers reported here are obtained from the test set. The data was split 80% training, 10% validation and 10% test.

IV. Energy Impact

Given the constraint from CIM architectures, limited input/output resolution and analog noise, the energy savings come at the cost of reduced accuracy. Figure 3 shows the accuracy-energy trade-offs for CIM implementations, including configurations previously realized through an ASIC implementation [17]. We contrast this performance against that of a digital solution implementing an optimized model [13]. LSTMs designed for digital systems are much less resilient to aggressive quantization [13], demonstrating a dramatic decline in performance once the digital bit-width is below 6 bits. Our proposed model is optimized for more aggressive quantization, demonstrating a more graceful degradation in performance. Thus, using the computational efficiency of a single CIM-core, we can achieve KWS task performance comparable to microcontroller-targeted digital implementations while operating with an order-of-magnitude greater energy-efficiency. However the maximum achievable accuracy of CIM architecture is also lower than their digital counterpart, limited in part by ADC resolution. The digital model delivers 3.85% improvement in accuracy on the test-set.

Figure 4a breaks down how different model sizes and hardware configurations impact the energy-efficiency of the system. Only a single parameter changes in each column, all other parameters are assigned their default value. As expected, ADC resolution has a significant effect on the energy-consumption. Otherwise the hidden dimensionality has the biggest effect on the energy consumption, e.g. increased vector size for hidden LSTM output causes the biggest increase in energy consumption. Similarly, Fig. 4b depicts how different parameters affect the model's accuracy on the test-set. Saliently, quantization of the input and the number of bits allocated to the digital PE significantly impact the accuracy. Contrasting Figs 4a and 4b, it becomes clear that parameters most sensitive to the accuracy are not necessarily the parameters that impact energy-efficiency. This opens up opportunities for energy-accuracy optimizations, with the pareto curves shown in Figure 3.

Figure 4c offers insight into the main energy drivers of a LSTM cell. Energy consumption in CIM architectures is dominated by the energy expended by the ADCs, in the digital PEs this is dominated by the memory-access energy. In both of these cases, these dominant components expend 83% to 98% of the total energy, opening the way to future optimizations.

V. Conclusion

We developed a LSTM model targeting CIM architectures, demonstrating a different set of accuracy-energy trade-offs compared to models optimized for digital architectures. Our tailored model, achieves 90% test accuracy on the google speech dataset at 25 uJ/decision, a 3.7× improvement over digital systems. Our findings suggest that memory management and storage have the largest impact on resource constrained architectures. Meanwhile the energy-efficiency of CIM architectures is dominated by ADCs and the accuracy is limited by input bit-widths.

References

[1] J. Giraldo, C. O'Connor, and M. Verhelst, "Efficient keyword spotting through hardware-aware conditional execution of deep neural networks," in *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2019, pp. 1–8.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[3] J. Li, R. Zhao, E. Sun, J. H. Wong, A. Das, Z. Meng, and Y. Gong, "High-accuracy and low-latency speech recognition with two-head contextual layer trajectory lstm model," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7699–7703.

[4] M. Zeng and N. Xiao, "Effective combination of densenet and bilstm for keyword spotting," *IEEE Access*, vol. 7, pp. 10 767–10 775, 2019.

[5] S. Yin, X. Sun, S. Yu, J.-s. Seo, and C. Chakrabarti, "A parallel rram synaptic array architecture for energy-efficient recurrent neural networks," in *2018 IEEE International Workshop on Signal Processing Systems (SiPS)*. IEEE, 2018, pp. 13–18.

[6] J. S. P. Giraldo, S. Lauwereins, K. Badami, and M. Verhelst, "Vocell: A 65-nm speech-triggered wake-up soc for 10-μw keyword spotting and speaker verification," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 4, pp. 868–878, 2020.

[7] F. Conti, L. Cavigelli, G. Paulin, I. Susmelj, and L. Benini, "Chipmunk: A systolically scalable 0.9 mm 2, 3.08 gop/s/mw@ 1.2 mw accelerator for near-sensor recurrent neural network inference," in *2018 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2018, pp. 1–4.

[8] D. Kadetotad, S. Yin, V. Berisha, C. Chakrabarti, and J. Seo, "An 8.93 tops/w lstm recurrent neural network accelerator featuring hierarchical coarse-grain sparsity for on-device speech recognition," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 7, pp. 1877–1887, 2020.

[9] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-tile 2.4-mb in-memory-computing cnn accelerator employing charge-domain compute," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, 2019.

[10] M. Imani, M. Samragh, Y. Kim, S. Gupta, F. Koushanfar, and T. Rosing, "Rapidnn: In-memory deep neural network acceleration framework," *arXiv preprint arXiv:1806.05794*, 2018.

[11] 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)*. IEEE, 2018, pp. 166–169.

[12] H. Dbouk, S. K. Gonugondla, C. Sakr, and N. R. Shanbhag, "Keyram: A 0.34 uj/decision 18 k decisions/s recurrent attention in-memory processor for keyword spotting," in *2020 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2020, pp. 1–4.

[13] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," *arXiv preprint arXiv:1711.07128*, 2017.

[14] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[15] Chen, Yu-Hsin and Krishna, Tushar and Emer, Joel and Sze, Vivienne, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," in *IEEE International Solid-State Circuits Conference, ISSCC 2016, Digest of Technical Papers*, 2016, pp. 262–263.

[16] N. Verma, H. Jia, H. Valavi, Y. Tang, M. Ozatay, L.-Y. Chen, B. Zhang, and P. Deaville, "In-memory computing: Advances and prospects," *IEEE Solid-State Circuits Magazine*, vol. 11, no. 3, pp. 43–55, 2019.

[17] W. Wan, R. Kubendran, S. B. Eryilmaz, W. Zhang, Y. Liao, D. Wu, S. Deiss, B. Gao, P. Raina, S. Joshi *et al.*, "33.1 a 74 tmacs/w cmos-rram neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models," in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2020, pp. 498–500.

[18] J. S. Meena, S. M. Sze, U. Chand, and T.-Y. Tseng, "Overview of emerging nonvolatile memory technologies," *Nanoscale research letters*, vol. 9, no. 1, p. 526, 2014.

[19] X. Zheng, R. Zarcone, D. Paiton, J. Sohn, W. Wan, B. Olshausen, and H.-S. P. Wong, "Error-resilient analog image storage and compression with analog-valued rram arrays: an adaptive joint source-channel coding approach," in *2018 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2018, pp. 3–5.

[20] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "Pact: Parameterized clipping activation for quantized neural networks," *arXiv preprint arXiv:1805.06085*, 2018.

[21] S. Wu, G. Li, F. Chen, and L. Shi, "Training and inference with integers in deep neural networks," *arXiv preprint arXiv:1802.04680*, 2018.

[22] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing lstm language models," *arXiv preprint arXiv:1708.02182*, 2017.

[23] A. Kazemi, C. Alessandri, A. C. Seabaugh, X. S. Hu, M. Niemier, and S. Joshi, "A device non-ideality resilient approach for mapping neural networks to crossbar arrays," *arXiv preprint arXiv:2004.06094*, 2020.

[24] T.-J. Yang and V. Sze, "Design considerations for efficient deep neural networks on processing-in-memory accelerators," in *2019 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2019, pp. 22–1.

[25] Q. Lou, T. Gao, P. Faley, M. Niemier, X. S. Hu, and S. Joshi, "Embedding error correction into crossbars for reliable matrix vector multiplication using emerging devices," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 139–144. [Online]. Available: https://doi.org/10.1145/3370748.3406583

[26] Y. N. Wu, J. S. Emer, and V. Sze, "Accelergy: An Architecture-Level Energy Estimation Methodology for Accelerator Designs," in *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2019.

[27] A. Parashar, P. Raina, Y. S. Shao, Y.-H. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S. W. Keckler, and J. Emer, "Timeloop: A systematic approach to dnn accelerator evaluation," in *2019 IEEE international symposium on performance analysis of systems and software (ISPASS)*. IEEE, 2019, pp. 304–315.

[28] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.