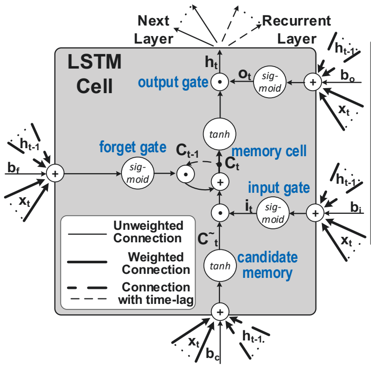
A 8.93 TOPS/W LSTM Recurrent Neural Network Accelerator Featuring Hierarchical Coarse-Grain Sparsity with All Parameters Stored On-Chip

Deepak Kadetotad, Visar Berisha, Chaitali Chakrabarti, Jae-sun Seo
School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA
{deepak.kadetotad, visar, chaitali, jaesun.seo}@asu.edu

***Abstract*—Long short-term memory (LSTM) networks are widely used for speech applications but pose difficulties for efficient implementation on hardware due to large weight storage requirements. We present an energy-efficient LSTM recurrent neural network (RNN) accelerator, featuring an algorithm-hardware co-optimized memory compression technique called hierarchical coarse-grain sparsity (HCGS). Aided by HCGS-based block-wise recursive weight compression, we demonstrate LSTM networks with up to 16× fewer weights while achieving minimal accuracy loss. The prototype chip fabricated in 65nm LP CMOS achieves 8.93/7.22 TOPS/W for 2-/3-layer LSTM RNNs trained with HCGS for TIMIT/TED-LIUM corpora.**

***Index Terms*—long short-term memory, speech recognition, hardware accelerator, weight compression, structured sparsity**



$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$
$$c_t^{\sim} = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot c_t^{\sim}$$
$$h_t = o_t \cdot \tanh(c_t)$$

h_{t-1} : hidden vector
x_t : input vector
W_{x*} / W_{h*} : weight matrices

Fig. 1. Illustration of LSTM cell with computation equations.

I. Introduction

The emergence of internet of things (IoT) devices that require edge computing with limited area and energy has garnered intense interest in energy-efficient ASIC accelerators for deep learning applications. The particular challenge of performing on-device automatic speech recognition (ASR) is that LSTMs that show high accuracy suffer from high complexity and require a large number of parameters to be trained and stored [1].

Recent works presented methods to reduce the complexity and storage of ASR hardware. Magnitude-based pruning was applied to LSTM hardware in [2], resulting in 20× model size reduction, but element-wise sparsity incurs considerable index memory and irregular memory access, hurting both performance and power. To overcome this, structured sparsity techniques have been proposed with row-/column-wise sparsity for RNNs [3], with block-wise sparsity for multi-layer perceptrons (MLPs) [4], and with block-circulant weight matrix for RNNs [5] in speech processing applications. However, these works exhibit limited weight compression of ~4× [3], [4] or high error rate [5], and have not been implemented in ASIC [2]–[5]. While recent ASIC designs targeting RNNs focus on improved energy-efficiency [6], [7], they do not incorporate compression techniques and do not report RNN accuracy for representative benchmarks, which are both necessary to accomplish practical ASR on small-form-factor edge devices.

In this work, we present a new hierarchical coarse-grain sparsity (HCGS) scheme that structurely compresses LSTM weights by 16× with minimal accuracy loss. HCGS-based LSTM accelerator which executes 2-/3-layer LSTMs for real-time speech recognition was prototyped in 65nm LP CMOS. It consumes 1.85/3.42 mW power and achieves 8.93/7.22 TOPS/W for TIMIT/TED-LIUM corpora.

II. LSTM and Hierarchical Coarse-Grain Sparsity

A. LSTM-based Speech Recognition

LSTM is a type of recurrent neural network (RNN) that shows state-of-the-art accuracy for speech recognition [1]. Each layer of a LSTM consists of neurons, which computes the final output h_t through four intermediate results called gates (Fig. 1). From the LSTM equations in Fig. 1, we see that the weight memory requirement of LSTMs is 8× when compared to MLPs with the same number of neurons per layer.

LSTM-based speech recognition typically consists of a pipeline of a feature extraction module, followed by a LSTM RNN and then by a Viterbi decoder. A commonly used feature for speech recognition is feature-space Maximum Likelihood Linear Regression (fMLLR). fMLLR features are extracted from Mel Frequency Cepstral Coefficients (MFCC) features, derived typically from 25 ms windows of audio samples with a 10 ms overlap between subsequent windows. The features for the current window are combined with those of past and future windows to provide context and the merged set of features are inputs to the neural network. In our implementation, we merge five past and five future windows to the current window to create an input frame with 11 windows, leading to 440 fMLLR features per frame. The output layer of the LSTM consists of

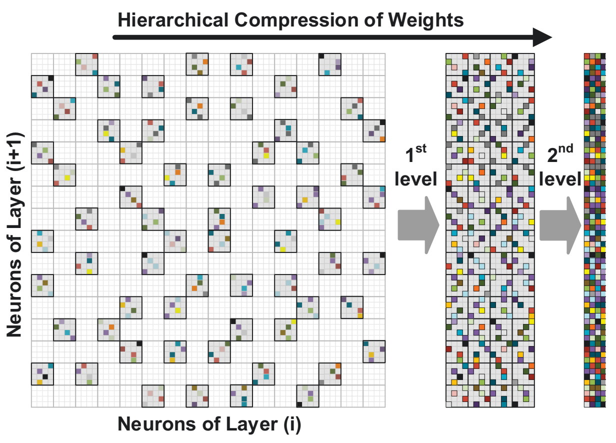


Fig. 2. Illustration of LSTM RNN weight compression featuring the proposed hierarchical coarse-grain sparsity (HCGS).

probability estimates that are sent to the Viterbi decoder unit to determine the best sequence of phonemes/words.

B. Hierarchical Coarse-Grain Sparsity

The proposed HCGS scheme maintains coarse-grain sparsity while further allowing fine-grain weight connectivity, leading to significant area and energy savings. Two-level HCGS is illustrated in Fig. 2, where the first level compresses weights (e.g. 4× compression) using a larger block size (e.g. 32×32) and the remaining weights in the large blocks go through the second level of compression (e.g. 4×) with a smaller block size (e.g. 8×8). The hierarchical structure of block-wise weights is randomly selected before the RNN training process starts, and is maintained throughout training and classification phases. The dropped blocks remain at zero and do not contribute to the physical memory footprint during both training and classification. TIMIT and TED-LIUM corpora are used to train the RNNs for phoneme and speech recognition, respectively. The baseline 3-layer, 512-cell LSTM RNN that performs speech recognition for TED-LIUM corpus requires 24 MB of weight memory in floating-point precision. Aided by (1) the proposed HCGS that reduces the number of weights by 16× and (2) low-precision (6-bit) representation of weights, the compressed parameters of a 3-layer, 512-cell LSTM RNN are reduced to only 288 kB (83× reduction in model size compared to 24 MB). The resultant LSTM network can be fully stored on-chip, which results in energy-efficient acceleration.

III. Architecture and Design Optimizations

A. Hardware Architecture

Fig. 3 shows the overall architecture of the proposed LSTM accelerator. It consists of the HCGS selector, input and output buffers, MAC unit, H-buffer, C-buffer, two memory banks (144 kB each) for weight storage, bias/index memory bank (8.5 kB), and the global controller. The proposed architecture facilitates the computation of one LSTM cell output per cycle after an initial latency period and reuses the MAC unit as outputs are computed in a layer-by-layer manner.

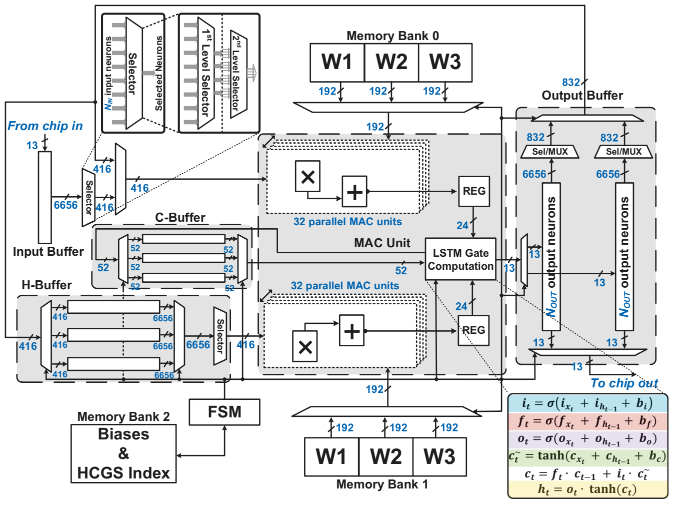


Fig. 3. Overall architecture of the proposed LSTM RNN accelerator.

1) HCGS Selector: The HCGS selector (Fig. 3, top-left) has two levels, where the first level of selector only enables the propagation of activations associated with larger non-zero weights blocks and the second level further filters through the activations associated with smaller non-zero blocks. For 16× HCGS compression, only 32 activation outputs are required from a total of 512 activations, ensuring only activations corresponding to non-zero weights propagate to the MAC unit, greatly boosting energy-efficiency.

2) Input and Output buffers: An input frame consists of fMLLR features as described in Sec. II-A. The input buffer is used to store the fMLLR features of an input frame, which is streamed in 13-bits at a time over 512 cycles. The output buffer consists of two identical buffers for double buffering, which enables continuous computation of the LSTM accelerator while streaming out the final layer outputs. Each output buffer employs a HCGS selector and a 6656:416 multiplexer to feedback the output of the current layer output to the next layer. The feedback from the output buffer to the input of the MAC facilitates the reuse of the MAC unit.

3) H-buffer and C-buffer: The H-buffer and C-buffer store the outputs of the previous frame (h_{t-1}) and cell state (c_{t-1}) for each layer, respectively.

4) MAC Unit: The MAC unit consists of 64 parallel MACs (computing vector-matrix multiplications) and the LSTM gate computation module (computing intermediate LSTM gate and final output values), which can effectively perform 2,064 operations in each cycle aided by the proposed HCGS compression. The non-linear activation functions (sigmoid and hyperbolic tangent) are implemented through piece-wise linear modules using 20 linear segments.

5) Weight/bias storage and global controller: Weights are stored in the interleaved fashion as described in Sec. III-B, where each memory sub-bank (W1-W3) stores weights corresponding to a single layer. This allows sub-banks storing weights of layers not currently being computed to be in sleep mode, leading to improved energy-efficiency.

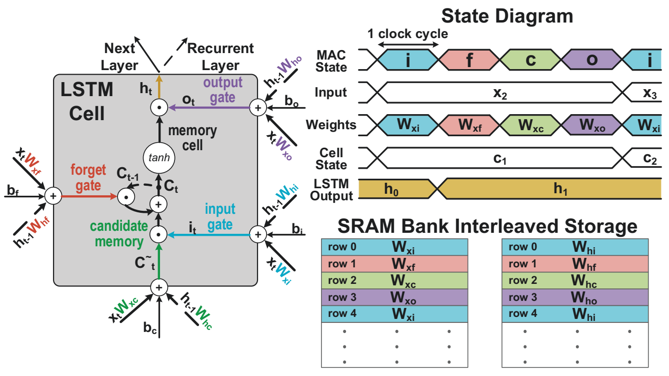


Fig. 4. LSTM data flow and core computations.

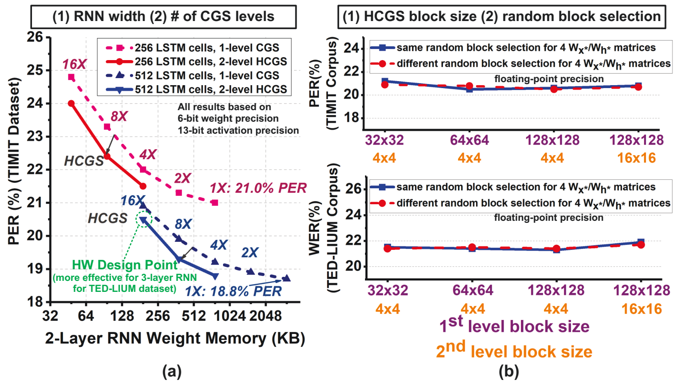


Fig. 5. HCGS design space exploration. (a) RNN width and the number of CGS levels. (b) HCGS block size and random block selection.

B. Interleaved Memory Storage

Fig. 4 shows the LSTM module computation operations and detailed state diagram of the MAC unit in our LSTM accelerator. The LSTM cell stores the intermediate products to compute the cell state (c_t) and output (h_t). Conventionally the cell states and outputs of an entire layer are computed only after every intermediate gate output for the corresponding layer is completed, this leads to additional memory requirements to store the intermediate gate outputs for all the LSTM cells in the layer. Instead, by taking advantage of the structure of the LSTM cell, the proposed architecture cycles between the four states computing internal gates of the LSTM cell, namely input gate (i_t), forget gate (f_t), output gate (o_t), and candidate memory ($\tilde{c}_t$). Additionally, the vector-matrix multiplications of $x_t W_{x*}$ and $h_{t-1} W_{h*}$ can be computed in independent streams, which increases throughput via parallelism.

To support this, we store each row of four matrices W_{xi}, W_{xf}, W_{xo}, and W_{xc} in a staggered manner (same for W_{h*}) in on-chip SRAM (Fig. 4, right-bottom), so that the computation of new c_t and h_t values can be completed after every four cycles, hence eliminating the requirement to store all intermediate gate outputs of the layer. In addition, the same random hierarchical block selection for HCGS is applied to all four matrices of W_{xi}, W_{xf}, W_{xo}, and W_{xc} (same for W_{h*}) to further reduce the index memory of the HCGS selector by 4×, resulting in only 1.17% index memory overhead.

C. Design Space Exploration

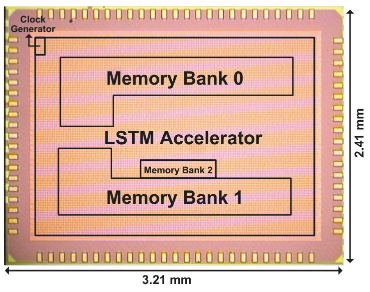
There are several important design parameters for HCGS based LSTM hardware design, including HCGS block size, compression ratio, and random block assignments. For this design space exploration, we constructed a number of LSTM RNNs; the simulation results are summarized in Fig. 5. For our LSTM accelerator, we reduced the weight precision to 6-bit and activation precision to 13-bit with negligible accuracy loss. Compared to single-level CGS [4], the 2-level HCGS scheme shows a favorable trade-off between phoneme error rate (PER) for TIMIT corpus and weight compression (Fig. 5(a)). Different random block assignments and sharing the HCGS masks for four LSTM gates do not affect the RNN accuracy (Fig. 5(b)). Overall, the 512-cell LSTMs shows better PER than the 256-cell LSTMs for various HCGS experiments. Based on these results, we selected the 512-cell LSTM and two-level HCGS to achieve up to 16× compression, for phoneme/speech recognition using TIMIT/TED-LIUM corpora.

IV. Measurement & Comparison

The proposed LSTM RNN accelerator is fabricated in 65nm LP CMOS. The chip micrograph and performance summary are shown in Fig. 6. For chip testing, we initially load the weights, biases and configuration bits to on-chip memory. To verify real-time operation, 13-bit input fMLLR features are streamed into the input buffer, while RNN outputs from the chip are streamed out and stored.

Fig. 7 shows the chip measurement results, where the accelerator operates up to 80MHz at 1.1V while consuming 67.3/72.5 mW for 2-/3-layer LSTMs, respectively. With voltage scaling, the power consumption at 0.68V for the 2-layer RNN for TIMIT is 1.85 mW at 8 MHz (Fig. 7(a)), and at 0.75V for the 3-layer RNN for TED-LIUM is 3.42 mW at 12 MHz (Fig. 7(b)). In all cases, the accelerator satisfies the real-time speech/phoneme recognition requirement of 100 frames/second. The memory and logic power breakdown for the 3-layer RNN at 0.75V is shown in Fig. 7(d). It can be seen that logic power is dominant due to the highly compressed weight memory despite the large number of RNN weight matrices. Pipelined with the LSTM gate computation unit, the MAC engines exhibit a high utilization ratio of 99.66%.

By leveraging HCGS, the LSTM accelerator achieves average energy-efficiency of 8.93/7.22 TOPS/W for running end-to-end 2-/3-layer LSTM RNNs for TIMIT/TED-LIUM corpora (Fig. 7(c)). We report the measured accuracy results of 20.6%



Technology	65nm LP CMOS	
On-chip SRAM	288 KB (weights) 8.5 KB (index & bias)	
Supply Voltage	0.68V – 1.1V	
RNN Precision	6-bit (weights) 13-bit (activations)	
Speech Dataset	TIMIT	TED-LIUM
# of RNN Layers	2	3
Error Rate	PER=20.6%	WER=21.3%
Power (@ real-time perf.) Energy-Efficiency (Vdd, Freq.)	1.85 mW 8.93 TOPS/W (0.68V, 8MHz)	3.42 mW 7.24 TOPS/W (0.75V, 12MHz)

Fig. 6. Prototype chip micrograph and performance summary.

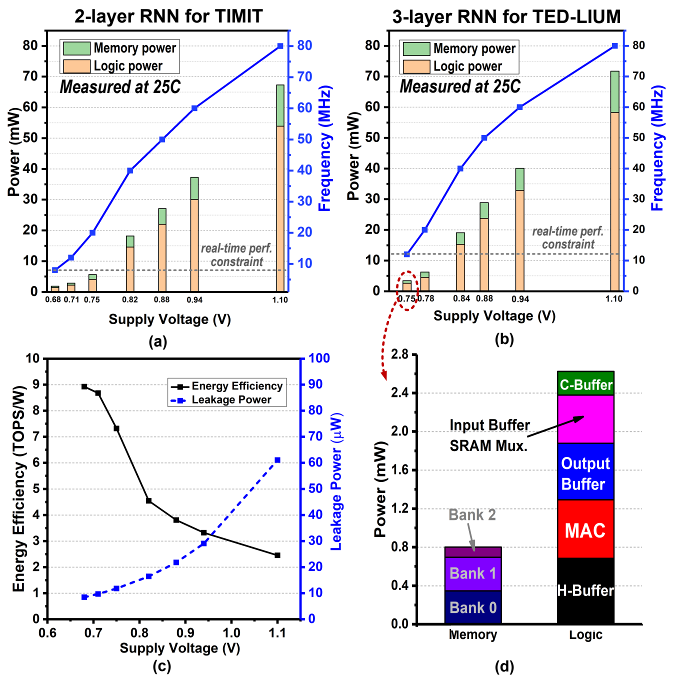
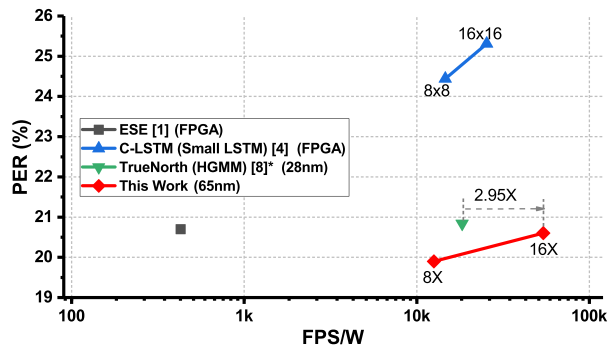


Fig. 7. Power and frequency measurement results with voltage scaling for (a) 2-layer LSTM for TIMIT and (b) 3-layer LSTM for TED-LIUM. (c) Measurement results for energy-efficiency (TOPS/W) and leakage power. (d) Power breakdown of 3-layer LSTM at 0.75V supply.



* [8] classification error from HGMM.

Fig. 8. Comparison of TIMIT PER and energy efficiency (frames per second/power, FPS/W) with prior LSTM implementations.

TABLE I
COMPARISON OF RNN PERFORMANCE WITH PRIOR WORKS.

	[2]	[5]	[6]	[7]	This Work
Technology	FPGA	FPGA	65nm CMOS	65nm CMOS	65nm CMOS
Area (mm^2)	-	-	1.57	19.36	7.74
On-Chip Memory (KB)	4.2 MB	280	82	348	297
Number of MACs	-	-	96	-	65
Bit-Precision Weights / Activations	12/16	16/16	8/16	16/16	6/13
Core Voltage (V)	-	-	1.24/0.75	1.2/0.67	1.1/0.68
Frequency (MHz)	200	200	168/20	200/10	80/8
Power (mW)	41W	22W	29/1.2	447/4	67.3/1.85
Peak Performance (GOPS)	2500	-	-	-	164.95/24.60
Energy-Efficiency (TOPS/W)	0.061	2.08	1.11/3.08	1.06/5.09	2.45/8.93
PER (TIMIT)	20.7%	25.3%	-	-	20.6% (measured)
WER (TED-LIUM)	-	-	-	-	21.3% (measured)

PER for TIMIT and 21.3% word error rate (WER) for TED-LIUM in Fig. 8. Compared to the RNN ASIC works of [6] and [7], this work shows 2.90× and 1.75× higher energy-efficiency (TOPS/W), respectively. Table I shows the detailed comparison with prior ASIC/FPGA works for RNNs.

Fig. 8 shows a comparison of frames/second/power (FPS/W) and PER for TIMIT corpus with prior works [2], [5], [8] that perform speech/phoneme recognition. The RNN accelerator [9] reports low power consumption but can only support limited keyword spotting tasks and is not considered. Compared to 28nm ASIC design supporting speech recognition [8], this work shows 2.95× higher energy-efficiency (FPS/W) with slightly better PER. Although FPS/W in [5] is comparable to our work, we achieve considerably lower PER. Conversely, [2] has comparable PER to our work but poor FPS/W. Overall, this demonstrates the effectiveness of our proposed design due to the algorithm-hardware co-optimization.

V. CONCLUSION

This paper presents a hierarchically compressed, energy-efficient LSTM accelerator for speech recognition. Exploiting the hierarchical block-wise sparsity and low-precision quantization, the accelerator stores the entire compressed weights of 3-layer, 512-cell LSTMs in 288 kB of on-chip SRAM and reduces the required computation by up to 16×. The 65nm prototype chip achieves average energy-efficiency of 8.93/7.22 TOPS/W for 2-/3-layer LSTMs for TIMIT/TED-LIUM corpora.

ACKNOWLEDGMENT

This work was in part supported by NSF grant 1652866, Samsung, ONR, and C-BRIC, one of six centers in JUMP, a SRC program sponsored by DARPA.

REFERENCES

[1] W. Xiong *et al.*, "The Microsoft 2017 conversational speech recognition system," in *IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, 2018.
[2] S. Han *et al.*, "ESE: Efficient speech recognition engine with sparse LSTM on FPGA," in *ACM/SIGDA Int. Symp. on Field-Programmable Gate Arrays*, 2017.
[3] W. Wen *et al.*, "Learning intrinsic sparse structures within long short-term memory," in *Int. Conf. on Learning Representations (ICLR)*, 2018.
[4] D. Kadetotad *et al.*, "Efficient memory compression in deep neural networks using coarse-grain sparsification for speech applications," in *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, 2016.
[5] S. Wang *et al.*, "C-LSTM: Enabling efficient LSTM using structured compression techniques on FPGAs," in *ACM/SIGDA Int. Symp. on Field-Programmable Gate Arrays*, 2018.
[6] F. Conti *et al.*, "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 *IEEE Custom Integrated Circuits Conf. (CICC)*, 2018.
[7] S. Yin *et al.*, "A 1.06-to-5.09 TOPS/W reconfigurable hybrid-neural-network processor for deep learning applications," in *IEEE Symp. on VLSI Circuits*, 2017.
[8] S. K. Esser *et al.*, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proc. of the National Academy of Sciences of the United States of America*, vol. 113, no. 41, 2016.
[9] J. Giraldo and M. Verhelst, "Laika: A 5μW programmable LSTM accelerator for always-on keyword spotting in 65nm CMOS," in *IEEE European Solid State Circuits Conf. (ESSCIRC)*, 2018.