Similarity-Based LSTM Architecture for Energy-Efficient Edge-Level Speech Recognition

Junseo Jo*, Jaeha Kung†, Sunggu Lee*, and Youngjoo Lee*
*Department of Electrical Engineering, POSTECH, Pohang, Korea
†Department of Information and Communication Engineering, DGIST, Daegu, Korea
youngjoo.lee@postech.ac.kr

***Abstract*—Targeting the resource-limited edge devices, we present a novel processing architecture of long short-term memory (LSTM) networks for low-power speech recognition. The proposed scheme newly defines the similarity score between two inputs of adjacent LSTM cells, and then the processing mode of the current LSTM cell is dynamically determined to reduce the energy while providing the accurate recognition. If the similarity is high, more precisely, the current cell is disabled and the outputs are directly copied from the prior vectors, totally eliminating complex LSTM operations. To maximize the skipping ratio without degrading the accuracy, for the first time, we analyze the effects of skipping the consecutive cells and set the upper limit of the number of consecutive skips. When two adjacent inputs are weakly similar, in addition, we modify the concept of the previous delta-computing, which approximately activate the LSTM cell with low computational resolution, further reducing the energy consumption. Compared to the previous state-of-the-art solutions, as a result, the proposed LSTM architecture remarkably saves the energy consumed for the accurate speech recognition, which is suitable to the resource-limited embedded edges.**

***Index Terms*—LSTM network, Low-power processing, Speech recognition**

I. Introduction

In past years, the speech recognition has been regarded as a critical player at the consumer electronics by continuously improving its accuracy with the advanced recurrent neural networks (RNNs) [1] [2]. Among various RNN architectures, the long short-term memory (LSTM) networks have been widely applied to the practical system due to their superior recognition accuracies [3] [4]. However, the previous accuracy-aware LSTM networks in general necessitate a huge amount of computational costs as well as memory overheads, which are supported by the high-performance server-scale platforms only [5]. As the demand for intelligent edge devices is rapidly increasing [6], therefore, it is urgent to develop a cost-effective speech recognition, which can be easily adopted to the resource-constrained embedded edges [7].

Several previous studies have revealed that the RNN sequence can be simplified by looking at confidence scores [8], [9] or adding additional gates [10], [11]. However, these schemes are in general used for categorizing time-series images, and only few recent works are suitable for the speech inputs. For example, the work in [10] appends extra networks to indicate the phase of LSTM cell and the computing energy is saved by deactivating some states. The delta-network from [12] can further reduce the network complexity by increasing the sparsity of input vectors with proper clipping thresholds.

In this paper, we further extend the previous skipping and approximate solutions to construct more optimized processing steps of LSTM-based speech recognition system. More precisely, the proposed architecture first defines the similarity score by comparing two input vectors of adjacent LSTM cells. If the score exceeds a certain threshold, then we skip the current cell operations, and simply copy the prior outputs to the results of the current time step. This is a simple but effective approach as the LSTM operations generate almost the same outputs for the highly-similar inputs. To compensate the accumulated errors, we carefully analyze the effects of skipping sequences and set the maximum number of consecutive skips, leading to the energy-efficient but still accurate recognition. In addition, we introduce the concept of pseudo-skipping for weakly-similar cases, where the internal cell operations are performed approximately by reducing the computing resolution. Targeting the DeepSpeech network for the LibriSpeech dataset [13], i.e., one of the state-of-the-art networks with bidirectional LSTM architecture, experimental results show that the proposed work improves the energy efficiency by up to 49% while providing the attractive word error-rate (WER).

II. Background

A. LSTM Network

Handling the sequence of continuous data, as depicted in Fig. 1, a typical LSTM architecture utilizes a series of identical LSTM units, each of which consists of state memories associated with multiple gates, i.e., input gate, output gate, and forget gate, denoted as i, o, and f, respectively [14]. For the sake of simplicity, we define d and k to represent the number of input and output features for each time stamp. At the t-th time stamp, the t-th LSTM unit receives three vectors; a $d \times 1$ vector for the current input data and two $k \times 1$ vectors for previous cell and hidden states, which are represented as $\mathbf{x}_t$, $\mathbf{c}_{t-1}$, and $\mathbf{h}_{t-1}$, respectively.

Note that performing the t-th LSTM unit generates the t-th cell and hidden states, i.e., $\mathbf{c}_t$ and $\mathbf{h}_t$. More precisely, in

This work was supported by Samsung Research Funding Incubation Center of Samsung Electronics under Project Number SRFC-TB1703-07.

978-1-7281-2954-9/19/$31.00 ©2019 IEEE

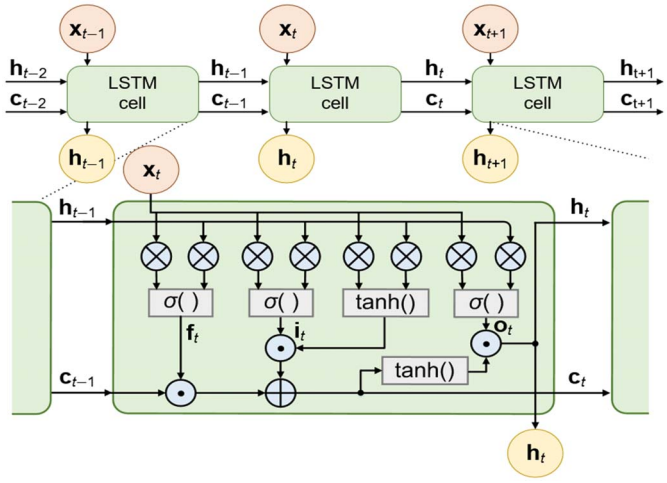

Fig. 1. A typical LSTM architecture.

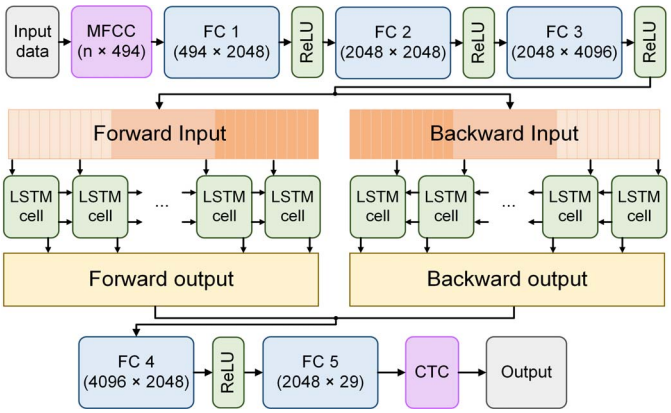

Fig. 2. The structure of DeepSpeech network [15].

each LSTM unit, three gate operations are firstly performed as follows, where $\mathbf{i}_t$, $\mathbf{o}_t$, and $\mathbf{f}_t$ are $k \times 1$ activation vectors from input, output, and forget gates, respectively.

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{V}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \quad (1)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{V}_o\mathbf{h}_{t-1} + \mathbf{b}_o) \quad (2)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{V}_f\mathbf{h}_{t-1} + \mathbf{b}_f) \quad (3)$$

For each gate operation, two matrices are used to provide the connection weights, which are determined at the training phase. Conceptually, a $d \times k$ matrix $\mathbf{W}$ represents the weighted paths from the input data to the corresponding gate, whereas a $k \times k$ matrix $\mathbf{V}$ describes the contributions from the hidden state. Note that a $k \times 1$ vector $\mathbf{b}$ is additionally trained for the bias values. Then, a sigmod function $\sigma()$ is finally applied to each gate operation, constructing three activation vectors.

After performing gate operations, the LSTM unit starts to calculate the output state vectors as follows.

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{V}_c\mathbf{h}_{t-1} + \mathbf{b}_c) \quad (4)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (5)$$

where $\odot$ denotes the element-wise product between two vectors. In the practical cases, the sequence of hidden states is regarded as the output sequence of LSTM architecture, which is used for the following fully-connected (FC) network for categorizing the recognized symbol at each time stamp.

B. DeepSpeech Network

To develop an energy-efficient speech recognition system, in this work, we select the DeepSpeech architecture as a baseline system, which is the one of the state-of-the-art solutions based on the LSTM topology [15]. Fig. 2 illustrates the overall structure of DeepSpeech network. In this network, the received voice signal is firstly divided into n sections, each of which consists of 494 mel-frequency cepstrum coefficient (MFCC) values. Note that the number of sections n varies depending on the length of input voice signal. The sequence of MFCC sets are then issued to three FC layers (FC1∼FC3) in order to generate n series of feature sets, where each feature set contains 4096 elements. Two identical n-length LSTM architectures are then applied to the sequence of feature sets in opposite direction, realizing the bidirectional LSTM structure as depicted in Fig. 2. As each direction produces a hidden state vector of 2048 elements at a time, total 4096 elements are existed in each time stamp. The results of the bidirectional LSTM are directly fed to two FC layers (FC4∼FC5) in time order, selecting the character from 29 candidates at each time stamp, i.e., 26 alphabets and 3 special symbols. Note that the connectionist temporal classification (CTC) unit is appended at the end of DeepSpeech processing to remove the redundant characters, finally constructing the recognized words.

TABLE I
COMPLEXITY ANALYSIS OF DEEPSPEECH NETWORK

Operation	FC1∼FC3	LSTM	FC4∼FC5	Total
Complexity ratio	0.11	0.82	0.07	1

Targeting the LibriSpeech data in [13], the baseline DeepSpeech network achieves a WER of 9.3% with the 16-bit fixed-point operations. Even for the attractive accuracy, however, this network requires a tremendous amount of computing costs as well as the processing energy, which cannot be the practical solution for the resource-limited edge devices. In terms of the number of multiply-accumulate (MAC) operations, for example, Table I analyzes the complexity of DeepSpeech network. Due to the intensive vector/matrix operations, it is clear that the bidirectional LSTM processing dominates the overall computing costs of DeepSpeech network, accordingly consuming a huge amount of energy. For the energy-efficient speech recognition, therefore, we propose in this paper the aggressive skipping and approximation methods on LSTM operations while providing the attractive WER performance.

III. PROPOSED METHODS

A. Similarity-based Cell Skipping

For the input voice signal, in general, the MFCC-based speech recognition manages the excessive number of sections

for taking account of various speakers having different speeds. Therefore, even a single character input is divided into several consecutive sections associated with similar MFCC coefficients, leading to the similar features handled by consecutive LSTM units. As a result, in the baseline DeepSpeech system in Fig. 2, the output sequence at the FC5 layer contains several identical characters for a single input character, and the final CTC step is inevitable to remove these redundant results. In this work, we basically propose the cell-level deactivation and approximation schemes by detecting these redundant positions at the run time. The energy-efficiency of speech recognition can be potentially enhanced by offering the optimized processing mode for each LSTM cell where the previous works cannot categorize the important LSTM cells, equally applying their approaches to all the cells [12], [16].

By checking the input features of consecutive LSTM units, i.e., $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$, we first define the similarity score (S_t) at the t-th time position as follows.

$$S_t = \frac{C(\mathbf{x}_t/2^w - M(\mathbf{x}_t) \odot \mathbf{x}_{t-1}/2^w) - C(\mathbf{x}_t)}{C(\mathbf{x}_t)} \quad (6)$$

The functions $C(\cdot)$ and $M(\cdot)$ are introduced for computing the similarity score. For a given input vector with k elements, $C(\cdot)$ counts the number of nonzero elements and $M(\cdot)$ returns an $1 \times k$ masking vector at which its elements become one for nonzero values and otherwise zero. The term '$M(\mathbf{x}_t) \odot \mathbf{x}_{t-1}$' identifies elements in $\mathbf{x}_{t-1}$ at positions of nonzero values in $\mathbf{x}_t$. Note that we mask out nonzero values in $\mathbf{x}_{t-1}$ which appear at positions of zeros in $\mathbf{x}_t$, as there is no computation involved at time 't'. We adopt the parameter w to set the similarity range for each element, in other words, it determines the number of bits to compare for the similarity analysis. Then, S_t estimates the relative distance of $\mathbf{x}_t$ from the previous input vector $\mathbf{x}_{t-1}$ for meaningful computations (i.e., nonzero elements) in $\mathbf{x}_t$.

By choosing 6 upper bits, i.e., $w = 10$, simulations on the baseline system reveal that similarity scores for redundant time positions are well separated from the non-redundant position. If we select the top 20% of time positions in terms of the similarity score, more than 70% of them were the time positions with redundant characters. Therefore, the proposed similarity score in Eq. (6) can be used to identify less important LSTM units. When the current similarity score becomes larger than the pre-determined threshold θ_s, in this work, we deactivate the entire LSTM operation at time 't' to reduce the computing energy as depicted in Fig. 3. For the state vectors of the skipped LSTM unit, we just transfer the previous results to the current outputs, i.e., $\mathbf{h}_k = \mathbf{h}_{k-1}$ and $\mathbf{c}_k = \mathbf{c}_{k-1}$. The skipping scheme is simple but effective to directly reduce the computing cost as two similar inputs tend to generate almost same outputs [12].

Fig. 4 illustrates the ratio of skipped LSTM operations to the total LSTM cells when we apply the straightforward skipping based on the proposed similarity score. Allowing the accuracy drop of 1%, which is widely accepted in practice [17], we can eliminate around 10% of LSTM operations, which is regarded as a marginal improvement. To enlarge the number of skipped

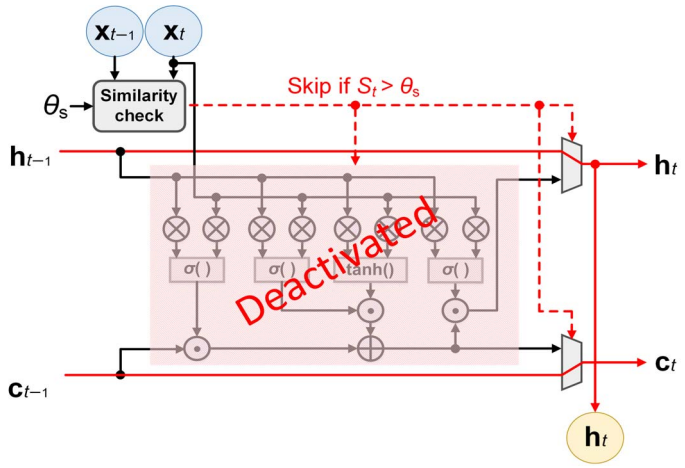

Fig. 3. The LSTM cell supporting the proposed similarity-based skipping.

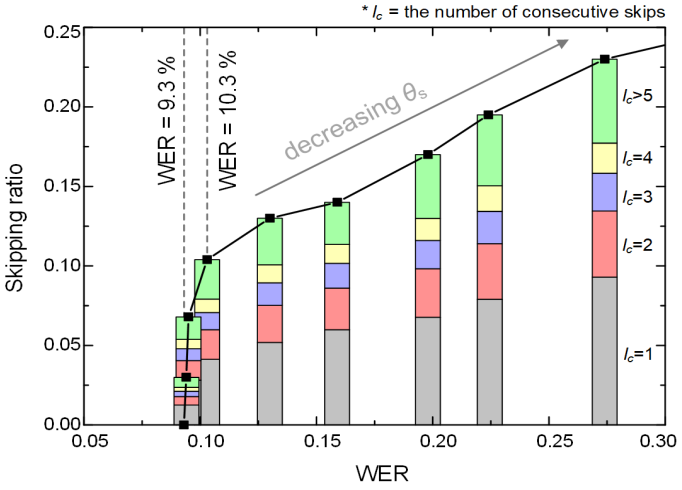

Fig. 4. The ratio of skipped cells without limiting the consecutive skips.

cells without degrading the accuracy, we further analyze the relations between skipping ratio and WER performance.

B. Limiting the Consecutive Skipping

Although the straightforward skipping of highly-similar LSTM cells causes only small amount of errors on the hidden and cell states, the accumulated errors from the consecutive skips severely degrade the quality of speech recognition. To show the amount of accumulated errors quantitatively, we calculate the mean squared error (MSE) between two $\mathbf{h}_t$ vectors, i.e., the vector from the exact LSTM cell computations, and the vector after some consecutive skips. As provided in Table II, the consecutive cell skipping propagates erroneous results, which directly affects the final recognition accuracy. Note that the portion of LSTM cells associated with more than 3 consecutive skips rapidly increases as θ_s decreases, significantly reducing the overall recognition accuracy as depicted in Fig. 4. Therefore, there is a clear limitation for improving the energy efficiency of LSTM processing with the straightforward skipping based on the proposed similarity score.

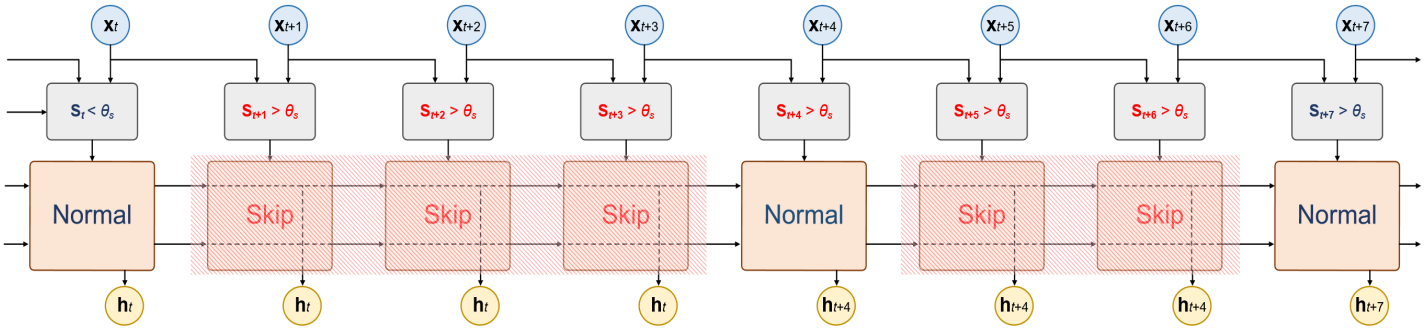

Fig. 5. Example of limiting the consecutive skipping for the case of $l_{\max} = 3$

TABLE II
MEAN SQUARE ERRORS OF $\mathbf{h}_k$ FROM CONSECUTIVE SKIPS

Number of consecutive skips	1	2	3	4
MSE	0.001698	0.011102	0.016924	0.031660

To minimize the amount of accumulated errors, we propose the limited cell-skipping approach by setting the maximum number of consecutive skips denoted as $l_{\max}$. In Fig. 5, it illustrates how the proposed scheme works with six successive LSTM cells having similarity scores larger than θ_s. By setting $l_{\max} = 3$, we first allow three consecutive skips, and then force the next LSTM cell to operate in the normal mode regardless of its similarity score. By limiting the maximum number of skips, the error accumulation is naturally bounded, also limiting the accuracy degradation due to the consecutive skips of LSTM cells. In achieving the target recognition accuracy, as a result, we can expect more skipped LSTM cells compared to the straightforward skipping method allowing the unlimited number of consecutive skips.

Fig. 6 shows how different $l_{\max}$ values affect the overall ratio of cell skipping. Note that reducing the number of consecutive skips gradually increases the skipping ratio for the similar WER level, remarkably reducing the computational complexity. When $l_{\max}$ is set to 2, the accumulated errors are minimized, allowing 2.7 times more removal of LSTM cells than the straightforward method with $l_{\max} = \infty$. Compared to the baseline system, the proposed similarity-based processing algorithm saves the computational complexity by 26.8% while degrading the WER performance by only 1%. If we allow more accuracy drops, the proposed method with $l_{\max} = 2$ can eliminate much more LSTM cells. Targeting the WER of 11.3%, for example, we can disable more than 45% of LSTM operations as depicted in Fig. 6.

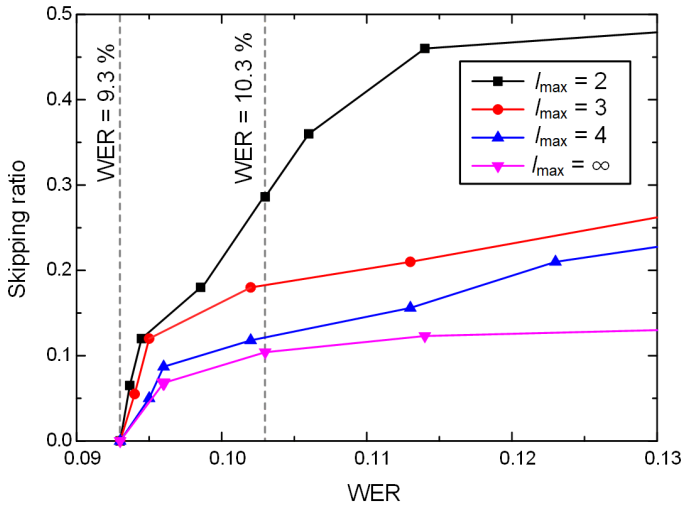

Fig. 6. The ratio of skipped cells for different $l_{\max}$ values.

C. Pseudo Skipping

In order to enhance the energy efficiency further, we introduce additional processing mode of LSTM operation with low complexity. When S_t is not large enough to skip the corresponding LSTM operation, then it is compared to another threshold θ_p, which is slightly smaller than θ_s. If the current input is marginally similar to the previous one, i.e., $\theta_s > S_t > \theta_p$, we perform the LSTM operation based on the previous delta-RNN algorithm [12]. In this algorithm, the input of an LSTM cell is reconstructed by subtracting the current input vector by the previous one. Then, the input sparsity is increased by applying the clipping threshold, which masks the small differences reducing the computational complexity. As the delta computing conceptually removes element-level operations in an LSTM cell, it can be regarded as the pseudo-skipping of that cell. Compared to the prior work that equally applies the delta-based operation for all LSTM cells [12], the proposed method dynamically relaxes the complexity depending on the level of similarity and set the proper skipping levels.

In addition, as the proposed pseudo skipping is used only for the LSTM cells whose similarity are larger than a certain level θ_p, element-wise differences between two adjacent input vectors tend to be much smaller than the elements in the original input vectors. Hence, we can lower the computing resolution of pseudo-skipped LSTM cells, further reducing the computational complexity. Note that the previous delta-computing cannot reduce the computing resolution to account for the larger differences. In this work, we adopt 8-bit operations for pseudo-skipped cells where the normal LSTM

cells are based on the 16-bit computing resolution. Hence, the proposed similarity-based algorithm with multiple skipping levels enables the energy-optimized platform for edge-level speech recognition by allowing finer-grained control between the accuracy and model complexity.

IV. Experimental Results

Targeting the LibriSpeech dataset [13], we carefully analyzed the computational overhead and the required energy consumption of various bidirectional LSTM networks: the original DeepSpeech network [15], DeepSpeech with delta-based processing [12], and DeepSpeech with the proposed similarity-based cell skipping. For the analysis, we first simulated two LSTM architectures; the baseline model activating all LSTM cells (BASE), and the delta-computing model preserving the baseline WER (DEL1). Allowing the 1% drop from the baseline WER, we also simulated an energy-aware LSTM architecture using the previous delta-computing by scaling the clipping threshold (DEL2), which is regarded as the state-of-the-art solution. The proposed methods are then applied to the BASE and DEL1 models, which are denoted as PRO_{BASE} and PRO_{DEL1}, gradually reducing the network complexity.

Table III compares all simulated LSTM models, with various energy-aware computing techniques, in terms of the recognition accuracy, the number of MAC operations, and the total number of parameters. For the pseudo-skipping operation, we normalize the number of MAC operations by considering the hardware complexity in 65nm CMOS technology, i.e., three 8-bit MAC operations are matched to one 16-bit operation. Allowing the unlimited skipping denoted as $l_{\max} = \infty$, the PRO_{BASE} model can relax the complexity of BASE by only 10%, which is a disappointing result compared to the DEL2 solution showing the same WER. Allowing at most two consecutive skips ($l_{\max} = 2$), the complexity of PRO_{BASE} becomes comparable to that of DEL1.

After introducing the pseudo-skipping approach, the PRO_{BASE} model finally overcomes the state-of-the-art solution (DEL2), reducing the number of MAC operations and the required parameters by 27% and 18%, respectively. When the proposed techniques are applied to DEL1, i.e., PRO_{DEL1}, we can further optimize the computational cost of DEL1 as summarized in Table III. However, the improvements by PRO_{DEL1} are less attractive than those from PRO_{BASE} as the DEL1 architecture removes the element-level operations even for the important LSTM cells in the BASE model, which degrades the skipping ratio achieving the same WER level. Normalized to the BASE model, Fig. 7 illustrates how the proposed solutions gradually improve the computational complexity and the size of involved parameters. Note that the fully-optimized PRO_{BASE} model provides the most attractive solution among different approaches. Compared to the BASE model, for example, we can save the number of MAC operations and the required bits of parameters by 55% and 49%, respectively, while providing the acceptable recognition accuracy.

To analyze the reduction in energy consumption by using the proposed similarity-based skipping algorithm, which is critical

TABLE III
Analysis of Speech Recognition Algorithms

LSTM model		WER	Number of MAC operations	Required memory size
BASE		9.3%	8,483K	136Mbit
DEL1		9.5%	6,149K	98Mbit
DEL2		10.3%	5,279K	84Mbit
PRO_{BASE}	$l_{\max} = \infty$	10.3%	7,511K	120Mbit
	$l_{\max} = 2$	10.3%	6,363K	102Mbit
	$l_{\max} = 2$ + pseudo	10.3%	3,830K	69Mbit
PRO_{DEL1}	$l_{\max} = \infty$	10.3%	5,869K	94Mbit
	$l_{\max} = 2$	10.3%	5,324K	85Mbit
	$l_{\max} = 2$ + pseudo	10.3%	4,772K	74Mbit

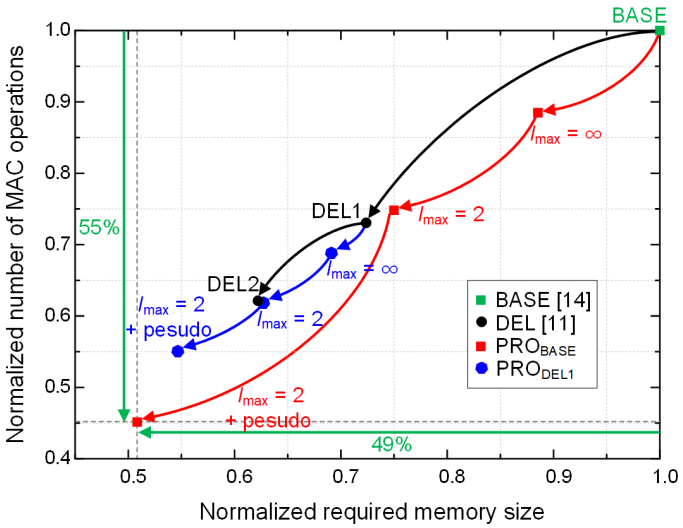

Fig. 7. Performance improvements by the proposed techniques.

in realizing the speech recognition on mobile edge devices, we designed and synthesized an LSTM accelerator architecture in 65nm CMOS technology. Similar to the prior works in [6], [16], as illustrated in Fig. 8, the LSTM accelerator consists of MAC cores, nonlinear operators, registerfiles and on-chip SRAM buffers. To reduce the number of external memory accesses, which normally necessitates a huge amount of energy, the accelerator architecture fully re-utilizes the parameters by introducing two-level internal memories, i.e., registerfiles and on-chip SRAMs. Then, different LSTM processing algorithms are mapped to the LSTM accelerator to calculate the required energy for recognizing the input speech signal. For fair comparisons, we adopt the pre-reported energy consumption for accessing the external DRAMs targeting the same technology [18], which are inevitable for storing the large-sized weight matrices. Note that the upper bits of MAC hardware are masked for performing the pseudo skipping, effectively reducing the dynamic energy consumption.

Based on the accelerator architecture in Fig. 8, different LSTM solutions are compared in terms of the energy efficiency, which is the required energy per LSTM cell operation.

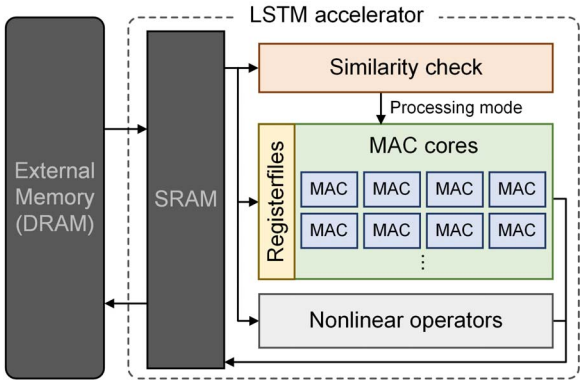

Fig. 8. The LSTM accelerator for estimating the energy consumption.

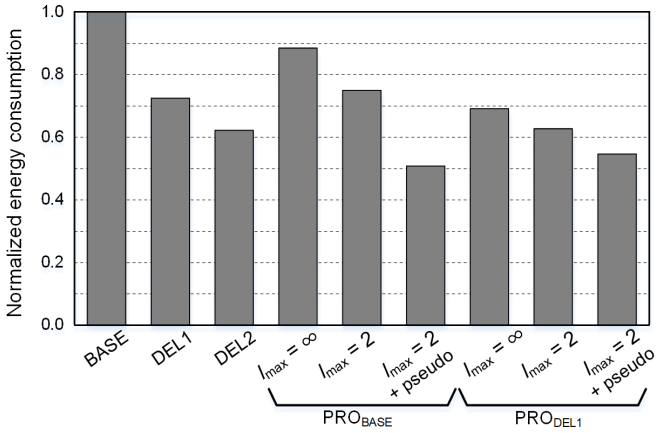

Fig. 9. The normalized energy consumption for different LSTM models.

As the proposed similarity-based skipping drastically reduces the number of computations and parameter size, it saves the energy consumption from the computing logics as well as the memory accesses. Allowing only 1% WER drop, as a result, the proposed work enhances the energy efficiency by more than 49.2% and 18.3% compared to the original DeepSpeech network [15] and the state-of-the-art delta-computing architecture [12], respectively, successfully supporting the energy-efficient speech recognition on resource-limited edge devices.

V. Conclusion

We have presented several optimization schemes for reducing the computing costs of LSTM processing of speech recognition system. Based on the newly-defined similarity score, the proposed method measures the importance of each LSTM cell at the run time, and then deactivates the redundant LSTM operations. We also determine the marginally-similar cells to perform the pseudo skipping operation, which is the delta-based computation with the reduced resolution. As a result, we can remarkably reduce the energy consumption without degrading the accuracy by relaxing the computing complexity only for the less important cells. Experimental results show that the proposed algorithm saves the required energy by more than 49% compared to the baseline network, allowing the energy-optimized edge-level speech recognition.

References

[1] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[2] T. Robinson, M. Hochberg, and S. Renals, "The use of recurrent neural networks in continuous speech recognition," in *Automatic speech and speaker recognition*. Springer, 1996, pp. 233–258.

[3] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[4] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 273–278.

[5] E. Nurvitadhi, , D. Sheffield, A. Mishra, S. Krishnan, and D. Marr, "Accelerating recurrent neural networks in analytics servers: Comparison of fpga, cpu, gpu, and asic," in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, Aug 2016, pp. 1–4.

[6] S. Han, J. Kang, H. Mao, Y. Hu, X. Li, Y. Li, D. Xie, H. Luo, S. Yao, Y. Wang *et al.*, "Ese: Efficient speech recognition engine with sparse lstm on fpga," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2017, pp. 75–84.

[7] R. Prabhavalkar, O. Alsharif, A. Bruguier, and L. McGraw, "On the compression of recurrent neural networks with an application to lvcsr acoustic modeling for embedded speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5970–5974.

[8] J. Kung, D. Kim, and S. Mukhopadhyay, "Dynamic approximation with feedback control for energy-efficient recurrent neural network hardware," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*. ACM, 2016, pp. 168–173.

[9] S. Sen and A. Raghunathan, "Approximate computing for long short term memory (lstm) neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2266–2276, 2018.

[10] V. Campos, B. Jou, X. Giró-i Nieto, J. Torres, and S.-F. Chang, "Skip rnn: Learning to skip state updates in recurrent neural networks," *arXiv preprint arXiv:1708.06834*, 2017.

[11] D. Neil, M. Pfeiffer, and S.-C. Liu, "Phased lstm: Accelerating recurrent network training for long or event-based sequences," in *Advances in Neural Information Processing Systems*, 2016, pp. 3882–3890.

[12] D. Neil, J. H. Lee, T. Delbruck, and S.-C. Liu, "Delta networks for optimized recurrent network computation," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2584–2593.

[13] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[16] C. Gao, D. Neil, E. Ceolini, S.-C. Liu, and T. Delbruck, "Deltarnn: A power-efficient recurrent neural network accelerator," in *Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2018, pp. 21–30.

[17] S. Wang, Z. Li, C. Ding, B. Yuan, Q. Qiu, Y. Wang, and Y. Liang, "C-lstm: Enabling efficient lstm using structured compression techniques on fpgas," in *Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2018, pp. 11–20.

[18] F. Tu, S. Yin, P. Ouyang, S. Tang, L. Liu, and S. Wei, "Deep convolutional neural network architecture with reconfigurable computation patterns," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 8, pp. 2220–2233, 2017.