Journal of Semiconductor Engineering Vol. 2, Issue 2, June, 2021
https://doi.org/10.22895/jse.2021.0001
Received 7 Jan 2021, Revised 14 Mar 2021, Accepted 22 Apr 2021 ISSN(On-line) 2733-6719

Energy-Efficient DNN Processor on Embedded Systems for Spontaneous Human-Robot Interaction

Changhyeon Kim, and Hoi-Jun Yoo
School of Electrical Engineering, KAIST, Daejeon, 34141, Republic of Korea

Corresponding Author: Hoi-Jun Yoo (hjyoo@kaist.ac.kr)

Funding Information: This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2020-0-01847) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

ABSTRACT
Recently, deep neural networks (DNNs) are actively used for action control so that an autonomous system, such as the robot, can perform human-like behaviors and operations. Unlike recognition tasks, the real-time operation is essential in action control, and it is too slow to use remote learning on a server communicating through a network. New learning techniques, such as reinforcement learning (RL), are needed to determine and select the correct robot behavior locally. In this paper, we propose an energy-efficient DNN processor with a LUT-based processing engine and near-zero skipper. A CNN-based facial emotion recognition and an RNN-based emotional dialogue generation model is integrated for natural HRI system and tested with the proposed processor. It supports 1b to 16b variable weight bit precision with and 57.6% and 28.5% lower energy consumption than conventional MAC arithmetic units for 1b and 16b weight precision. Also, the near-zero skipper reduces 36% of MAC operation and consumes 28% lower energy consumption for facial emotion recognition tasks. Implemented in 65nm CMOS process, the proposed processor occupies 1784x1784 um2 areas and dissipates 0.28 mW and 34.4 mW at 1fps and 30fps facial emotion recognition tasks.

KEY WORDS
Deep learning, deep learning ASIC, deep neural network, mobile deep learning, reinforcement learning.

1. INTRODUCTION
The recent development of deep learning technology enables machines to recognize the human user's emotion accurately with facial expressions [1] and dialogues [2]. Recent studies [3-5] try to adopt emotion recognition into mobile devices, which allows machine to understand user's intend and enables more natural human-robot interaction (HRI). Unlike the recognition task, the real-time operation is essential in the field of HRI, and it is too slow to use remote learning on the server through the network. New learning techniques such as reinforcement learning (RL) are needed to determine and select the correct robot's response locally.

Deep RL (DRL) agent, an agent having both RL and DNN, adapts the DNN through with two components in general, Actor and Learner (Figure 1) [1], [2]. While processing the DRL, the actor continuously interacts with the environment, and the learner trains the DNN at regular interval to obtain the maximum reward. The actor observes the state of the environment (St) and selects the actions (At) determined by current DNN outputs and its policy. After the At, the agent receives the state (St+1) and a scalar reward (Rt) at the next time step. Every time step, the actor stores the

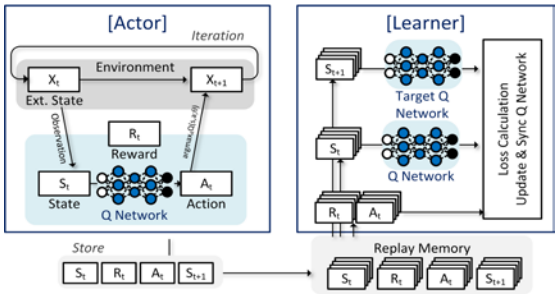

Figure 1. Deep Reinforcement Learning Agent

sampled experiences, {St, At, St+1, Rt}, in the replay memory to train DNN by the learner. The replay memory stores a large number of experiences (~10,000) to exploit the experience replay technique, which is widely used for DRL algorithms in order to stabilize the training of DRL agent [3]. The learner fetches the multiple random samples of experiences as a batch, and it uses loss function to maximize the reward with the action determined by current DNN. Then, the learner updates the DNN weights with calculated gradients from the loss function.

Previous DNN HW [4], [5] performed only inference, and since their data parallelism had only fixed datapath, their memory access is not optimized for DRL application, which requires adaptive data reuse for not only inferencing but also training. Even though [6] was able to tune the DNN, it did not update the weights of all layers of FCL but the last few layers only with a limited number of poor performance PEs.

For natural human-robot interaction (HRI), we newly introduce an emotional HRI system for mobile devices, as shown in Figure 2. It consists of 3 parts, (1) face detection/alignments for face RoI generation, (2) CNN-based facial emotion recognition (FER). And RNN-based emotional dialogue generation (EDG) where the RNN is simultaneously updated through continuous interaction with users by DRL. The output of FER, user's emotion, is used as input of EDG with user's speech. EDG performs natural language processing and outputs different dialogues due to the user's emotional state. We optimized both FER's and EDG's DNN network's weight bit-precision in order to realize real-time operation. The optimized CNN network has 16b weight and 1b weight bit precision for the input layer and other layers, respectively. It achieves <1% accuracy degradation in comparison with the model using 32b floating point weights. Besides, the EDG model is trained by reinforcement learning method. It uses 4b weight bit precision.

In this paper, an energy-efficient DNN processor has been integrated, which supports variable weight bit precisions in order to combine two different DNN models into a battery-powered mobile device. We propose variable weight-bit precision DNN processor with an energy-efficient look-up-table based processing engine (LPE). It fully supports 1b-to-16b weight precision with binary multiplication optimized LPE and bit-serial shifter. It consumes 57.6% and 28.5 % less energy in comparison with a conventional multiplier. Further, it supports near-zero skipping, which reduces the average 36% MAC operation and 28% energy consumption for facial emotion recognition tasks.

The rest of this paper is organized as follows. In Section 2, the overall architecture and the key building blocks are described. Section 3 shows the implementation results and measurement results. Section 4 provides a conclusion.

2. BUILDING BLOCKS

A. Overall Architecture

Figure 3 shows the overall architecture of the proposed DNN processor. It consists of the preprocessing core, and the LUT-based PE core (LP Core), which are connected with a network-on-chip interface for communications. The pre-processing core performs face detection and alignment for face RoI generation, and the face RoI is transferred to LPE core. Two 6KB OMEMs and 36KB WMEM are integrated into LPE core. When the activation of DNN comes into activation buffer, the Near-zero detector bit-shift activation data and skip the MAC operation of activations smaller than the predefined threshold value. 4 LPE clusters perform DNN processing of different input channels. Each LP cluster consists of 4 LPEs, and each LPE includes 8-entry look-up table of 4 input activations and outputs 12 different output data using 12 8-to-1 multiplexers. The detailed operation of LPE is described later. 12 outputs of 4 LPs and LP clusters are accumulated by 12 4-way add/sub trees. The bit-serial shifter performs bit-serial multiplication for weight-bit scalable DNN processing. The outputs are stored

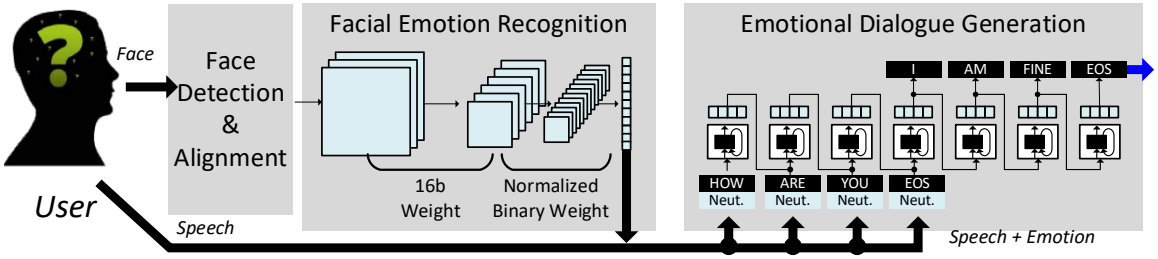

Figure 2. Emotional HRI System with Facial Emotion Recognition and Emotional Dialogue Generation Model

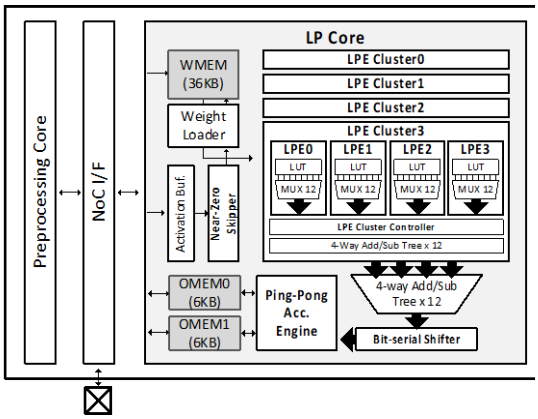

Figure 3. Overall Architecture

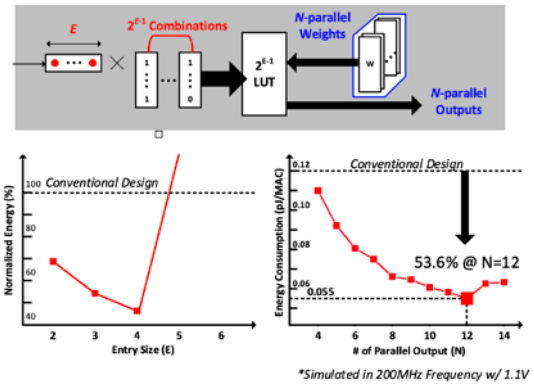

Figure 5. Energy Consumption of LPE with various number of entry size and parallel outputs.

in OMEMs, and ping-pong accumulation engine accumulates the partial-summations of DNN.

B. LUT-based Processing Engine (LPE)

Figure 4 explains the binary weight multiplication of LPE. The outputs of LPEs are accumulated and shifted by the bit-serial shifter for bit-serial multiplication. In the case of 1b weight, 1-weights, and 0-weights represent addition, and substitution of activation, respectively. When LPE fetches new input activations, the LPE generates and keeps all the combinations that can be made from 4 input activations with binary weights (A-step: LUT update). Because each input activations is multiplied output channel size (CO) kernel size (k) times with DNN weights, if we pre-calcu late and store 16 different combinations, whole Co k binary weight multiplication can be replaced by simple indexing of the LUT based on the weights (B-step: Calculation). To further reduce the number of entries of the LUT, we exploit the characteristic of 2's complement. The total of 16 combinations can be divided into two parts, as shown in the figure. The 8-entry table in the left, physical LUT, stores multiplication results of weight's MSB equals one. The multiplication results of weight's MSB equals zero, logical LUT, is the full inversion of the physical LUT that we replace the half entries, eight

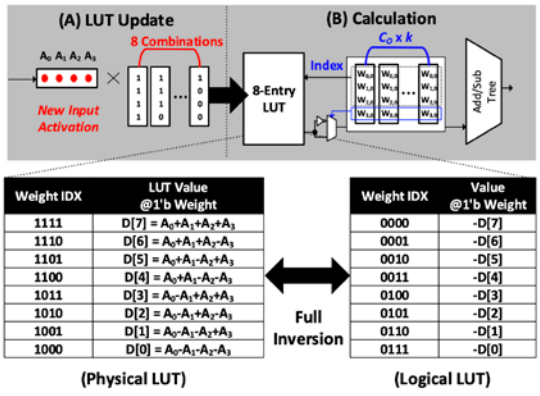

Figure 4. Detailed Binary Weight Multiplication of LPE

16b registers, with an inverter and a multiplexer. In addition, LPE cluster controller reconfigures the data path of 4-way add/sub trees that it calculates and store the pre-calculate the LUT value during the A-step, and it accumulates and outputs the indexed outputs of 4 LPEs at B-step.

In Figure 5, we simulated and analyzed the energy consumption of LPE by the number of input activation per LUT (E) at 200MHz, 1.1V operating condition. The normalized energy consumption is measured with the same external memory bandwidth and the same throughput condition. When the E becomes larger, the accumulation power of LUT outputs becomes smaller, but the entry size of LUT becomes exponentially larger. In our simulation, the optimal size of E is 4 that we designed LUT with 8-entries. In addition, because it is more efficient to index single LUT with multiple weights, we measured energy consumption by the number of parallel outputs of LPE (N), and we find the optimal size of E and N as 4 and 12. Therefore, 12 multiplexers are integrated into an LPE, and an LPE cluster accumulates 4 12 parallel outputs with 12 4-way add/sub tree.

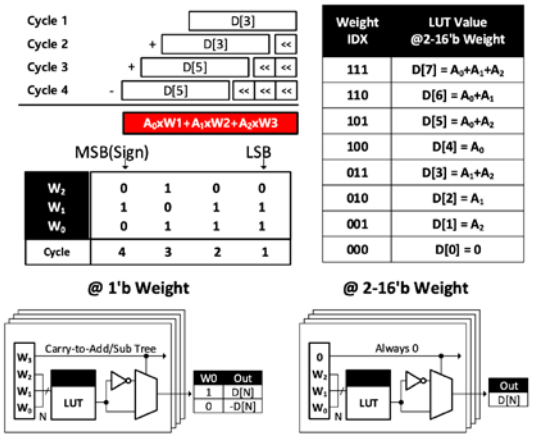

Figure 6. 2b-to-16b bit-serial multiplication using LPE results

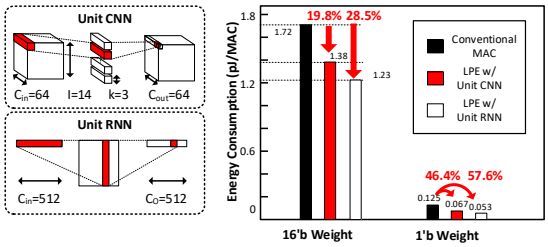

Figure 7. Energy Consumption Caparison with Conventional

With binary-multiplication optimized LPE, the bit-serial multiplication method is adopted in order to realize the variable weight bit precision of DNN. Figure 6 explains the PE configuration at N-bit weights. It sequentially calculates the addition and substitution of input activations with multiplication results of binary weights from the LSB to the MSB for N-cycles. During the 2b-to-16b multiplication, LPE controller reconfigures the configurations of LPE that LUT stores all combinations of 3 input activations as shown in the figure. In Figure 7, the LPE energy consumption for CNN and RNN is described. The power consumption for 16'b CNN and RNN is reduced by19.8% and 28.5%, respectively. And for the binary weight, the power consumption for the 1'b CNN and RNN is reduced by 46.4% and 57.6%, respectively.

Figure 8 shows the spatial and temporal data mapping of LPEs for CNN processing. In the beginning, an LPE cluster fetches 16 input activations, which are the pixels of the same coordinate but 16 different input-

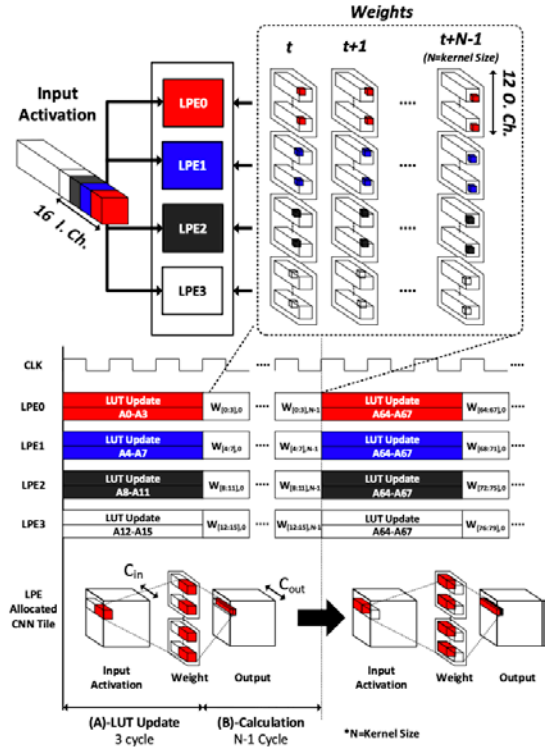

Figure 8. Spartial and Temporal Data Mapping of LPEs

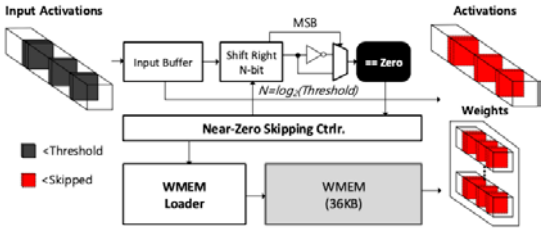

Figure 9. Near-zero Skipping Method

channels. Then, each LPE updates LUT with 4 input activations (A-step). It takes 3-cycles. After the LUT update step, LPEs fetch weights and outputs the values of LUT by weights. At every cycle, LPE fetches 4×12 weight, which are the kernels of the 4 different input channels, 12 different output channels, but the same coordinate. So, each LPE outputs 12 LUT values at every cycle, and the 4-way add/sub-tree in LPE cluster accumulates the 4×12 outputs to 12 output. It takes kernel-size cycles. After kernel-size cycles, there are no kernels to multiply with input activations in LPE, that LPE fetches new input activation in the same coordinate but 16 different input-channels and repeat the above procedure. What if the input activations within the same coordinate are fully processed, the LPE fetches the new input activations in the next coordinate.

The near-zero skipper is explained in Figure 9. While the input buffer fetches new input activations, near-zero-skipper monitors input buffer and blocks the activations, smaller than the threshold value. This threshold value is programmable that can be varied through different layers of DNN. It simply bit-shift the data in input-buffer to the right, and what if the shifted data is 0, it blocks the activations. In the case of negative numbers, it inverses full-bits of the data before bit-shift. Moreover, it skips the corresponding weights of the blocked input activations. While processing the FER normalized binary weight CNN with FER2013 dataset, the average skipping ratio is 36% with the threshold of 4 that the overall power consumption is reduced by 28%.

3. IMPLEMENTATION RESULTS

Figure 10 shows the chip micrograph. The proposed processor is fabricated with 65nm 1P8M Logic CMOS process with 1784×1784 um^2. It operates from 0.67-1.1V supply voltage with 5-200MHz clock frequency range. In the case of facial expression recognition with weight-bit precision optimized CNN, it consumes 0.28 mW and 34.4mW at 1fps and 30fps, respectively. For the emotional dialogue generation tasks, it generates different response due to the user's emotional states. It consumes 56 mW and takes 1.36-second latency with 4 layered 8b weight GRU model. The peak CNN power efficiency is measured as 13.6 TOPS/W at 0.66V, 5MHz operating condition, and peak RNN/FC power efficiency is measured as 15.7 TOPS/W at the same condition with binary weight bit

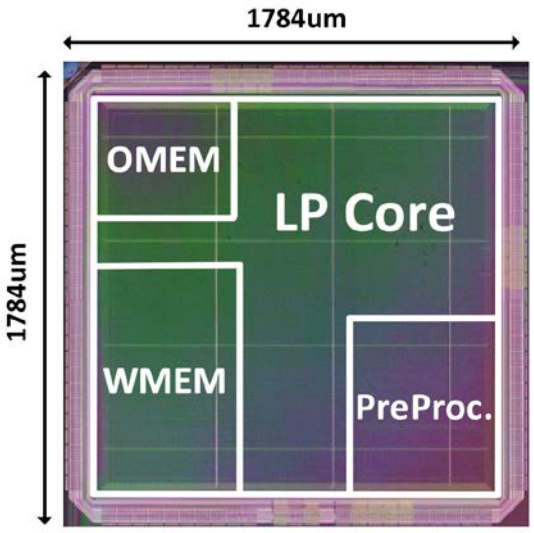

Figure 10. Chip Micrograph

precision.

Figure 11 shows the facial emotion recognition test with FER2013 dataset. The network is trained to have 8 CNN layers followed by 2 fully connected layers using DRL. Only the first input layer of the network has 16-bit weight kernel, and the other layers are having 1-bit binary weights. Test result shows that near-zero skipper skips the 36% of input and intermediate activations in average when the skipping threshold is set as 4. In our face recognition test, our proposed chip consumes 0.28 mW average for the 1fps frame rate at 5MHz, 0.66V. For the 30fps frame rate, it consumes 34.4 mW at 100MHz, 1.1V operating condition.

4. CONCLUSIONS

We propose an energy-efficient DNN processor with a LUT-based processing engine and near-zero skipper. A CNN-based facial emotion recognition and an RNN-based emotional dialogue generation model is integrated for natural HRI system and tested with the proposed processor. LPE supports 1b to 16b variable weight bit precision with and 57.6% and 28.5% lower energy consumption than conventional MAC arithmetic units for 1b and 16b weight precision. Also, the near-zero skipper reduces 36% of MAC operation and consumes 28% lower energy consumption for facial emotion recognition tasks. Implemented in 65nm CMOS process, the proposed processor occupies 1784×1784 um^2 areas and dissipates 0.28 mW and 34.4 mW at 1fps and 30fps in CNN-based facial emotion recognition task with face detection and face alignment. Also, for the RNN-based emotional dialogue generation task, it consumes 56mW with 1.36-second latency. In conclusion, the 1b-to16b fully variable weight bit precision low-power DNN processor for <100mW natural HRI system is successfully realized for mobile devices.

Happy Neutral

Facial Expression Recognition: 71.8% FER2013										
Layers	C1	C2	C3	C4	C5	C6	C7	C8	FC1	FC2
Kernel	7x7	3x3								
CH. In	3	64	64	128	256	256	256	256	1024	1024
CH. Out	64	64	128	128	256	256	256	256	1024	7
W. Bit.	16	1 (Binary)								

Performance

Precision	Feature – 16b integer
	Weight – 16b (input layer), 1b (others)
Avg. Skip Ratio	36% @ Threshold=4
Power	0.28mW @ 5MHz, 0.66V (1 fps)
	34.4mW @ 100MHz, 1.1V (30 fps)

Figure 11. Facial emotion recognition test with FER2013 dataset and performance summary

REFERENCES

[1] V. Mnih et al., "Human-level control through deep reinforcement learning," in Nature, vol. 518, no. 7540, p. 529, 2015.
[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. J. a. p. a. Klimov, "Proximal policy optimization algorithms," 2017.
[3] V. Mnih et al., "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013.
[4] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H.-J. Yoo, "UNPU: A 50.6 TOPS/W unified deep neural network accelerator with 1b-to-16b fully-variable weight bit-precision," in 2018 IEEE International Solid-State Circuits Conference-(ISSCC), 2018: IEEE, pp. 218-220.
[5] K. Ueyoshi et al., "QUEST: A 7.49 TOPS multi-purpose log-quantized DNN inference engine stacked on 96MB 3D SRAM using inductive-coupling technology in 40nm CMOS," in 2018 IEEE International Solid-State Circuits Conference-(ISSCC), 2018: IEEE, pp. 216-218.
[6] Z. Yuan et al., "STICKER: A 0.41-62.1 TOPS/W 8bit Neural Network Processor with Multi-Sparsity Compatible Convolution Arrays and Online Tuning Acceleration for Fully Connected Layers," in 2018 IEEE Symposium on VLSI Circuits, 2018: IEEE, pp. 33-34.

AUTHOR BIOGRAPHIES

Changhyeon Kim received the B.S (2014), M.S (2016), and Ph.D (2020) degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. His current research interests include low power SoC design, especially focused on parallel processor for artificial intelligence and machine learning algorithms.

Hoi-Jun Yoo (Fellow, IEEE) graduated from the Electronic Department, Seoul National University, Seoul, South Korea, in 1983, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1985 and 1988, respectively.
Prof. Yoo served as a member for the Executive Committee of ISSCC, Symposium on VLSI, and A-SSCC, the TPC Chair for the A-SSCC 2008 and ISWC 2010, the IEEE Distinguished Lecturer from 2010 to 2011, the Far East Chair for the ISSCC from 2011 to 2012, the Technology Direction Sub-Committee Chair for the ISSCC in 2013, the TPC Vice Chair for the ISSCC in 2014, and the TPC Chair for the ISSCC in 2015. (Detail biography is on http://ssl.kaist.ac.kr)