An Energy-Efficient Low Power LSTM Processor for Human Activity Monitoring

Conference Paper · September 2020

DOI: 10.1109/SOCC49529.2020.9524796

3 authors, including:

Hasib-Al Rashid
University of Maryland, Baltimore County
24 PUBLICATIONS **237** CITATIONS
SEE PROFILE

Tinoosh Mohsenin
University of Maryland, Baltimore County
169 PUBLICATIONS **3,747** CITATIONS
SEE PROFILE

An Energy-Efficient Low Power LSTM Processor for Human Activity Monitoring

Arnab Neelim Mazumder, Hasib-Al Rashid, and Tinoosh Mohsenin
Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County
Email: (arnabm1, hrashid1, tinoosh)@umbc.edu

2020 IEEE 33rd International System-on-Chip Conference (SOCC) | 978-1-7281-8746-4/20/$31.00 ©2020 IEEE | DOI: 10.1109/SOCC49529.2020.9524796

***Abstract*—A low complexity Long Short-Term Memory (LSTM) based neural network architecture is proposed in this paper for the classification task of recognizing different human activities in relation to various sensor modalities. The proposed model consists of one LSTM layer of 8 units, two dense layers having 80 and 32 neurons respectively and one output layer with 13 neurons for multi-class classification. We achieved 87.17% classification accuracy with our proposed model to classify 12 activities from each other. The proposed work involves extensive hyperparameter optimization in order to develop a hardware implementable model architecture while also maintaining high classification accuracy. In this case, quantization allowed the model to have a small size of 365 kB which resulted in 2x improvement over the 16-bit precision. The hardware architecture is designed in a parameterized way with respect to the number of input channels, filters, and data width to give more flexibility in terms of reconfigurability. The proposed LSTM based model is fully synthesized and placed-and-routed on Xilinx Artix–7 FPGA. Our reconfigurable hardware architecture consumes 82 mW power at an operating frequency of 160 MHz. Our LSTM based FPGA hardware achieves 7.7 GOP/s/W energy efficiency which outperforms previous hardware architecture implementations on Human Activity Recognition (HAR) by atleast $5.2\times$. The proposed low power LSTM processor also has an improvement of atleast $4.1\times$ for energy efficiency over previous LSTM works based on language modeling and artifact detection.**

Keywords: Human Activity Recognition, Physiological Monitoring, Long Short Term Memory (LSTM), FPGA

I. Introduction

Time series classification has been at the forefront of the modern-day research paradigm due to the vast amount of application-specific opportunities that are entwined in our day to day lifestyle. Assessment of time series data has seen a wide range of applications over the years. This has accelerated areas of further research particularly for multimodal time-series data. Nevertheless, the size, nature and dimensionality of these data make the analysis of these signals ever more challenging.

A stream of data collected and sampled at specific time intervals denoting some distinct action is generally called a time series data. Depending on the process of data collection time series data can be univariate or multivariate. Multivariate time series data relates to the multimodal signals which are used widely in different applications. These multimodal signals are captured using accelerometers, magnetometers, gyroscopes, and heart rate monitors sampled at different frequencies. Human Activity Recognition (HAR) is one such application where different algorithmic and application-based procedures are being introduced daily to understand human behavior at its most granular level. Usually, time series problems involving physiological monitoring have been solved using various approaches such as Dynamic Time Warping (DTW) [1], K-nearest neighbors (KNNs) [2], End to End Convolutional Neural Networks (End to End CNN) [3], [4] and Deep Neural Networks (DNNs). However, classification tasks employing KNNs and DTW are associated with long execution time which is not warranted. Recently, DNNs have become very popular for multimodal signal processing [5], [6]. This work introduces an energy-efficient, scalable hardware implementation of the proposed LSTM model illustrated in Fig. 1 that can classify different human activity from multi-channel time-series data that meets the 1-second deadline of data processing time. The major contributions of this paper are as follows:

- Propose an LSTM based physical activity recognition algorithm for multimodal time-series signals.
- Perform substantial hyperparameter tuning with the goal of reducing computation complexity and memory requirements to meet the required accuracy.
- Present an 8-bit quantized LSTM hardware to improve power consumption and memory requirements.
- Implement a parameterized hardware architecture that replicates the algorithm for low-power deployment at embedded application level.
- A comprehensive comparison of the proposed work with state-of-the-art FPGA implementation results.

II. Related Work

There is a plethora of works and algorithmic models that deal with real time-series signals. With the introduction of deep neural networks complex time series data with multiple modalities have been put to use for careful implementation of monitoring and diagnosis scenarios in medical environments [7], [8]. In terms of human activity monitoring, Convolutional Neural Networks (CNN) based models have been broadly used for classification and detection tasks. Convolutional Neural Networks extract spatial features of an image to achieve its desired target. This aspect of the CNNs has been utilized over the years for time series signal assessment. Windowed images at different

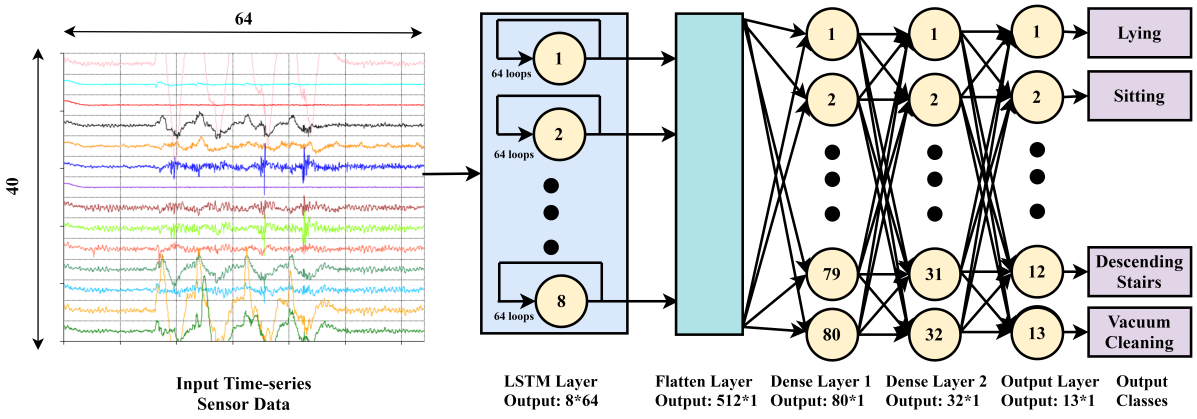

Fig. 1: Proposed LSTM based network architecture for human activity monitoring which consists of 1 LSTM layer, 2 dense layers and 1 output (softmax) layer to classify 12 different activities from each other.

samples of real-time signals serve as input to the models which helps the model to figure out the spatial information associated with the data [9]. However, signals associated with physical activity contain concentrated regions of fluctuation or sequence when activity occurs and the windowing process for the CNN architecture takes into account of these sequences during training. This sequential nature of physiological signals allows recurrent neural networks (RNNs) to be particularly useful in this regard. RNNs have the ability to capture temporal information linked with the multimodal signals to create generalized models. This leads to the use of LSTM networks for human activity recognition. In [10] LSTMs were used to concatenate positive time direction (forward state) and negative time direction (backward state) while also granting provisions for residual connections between stacked cells to alleviate the vanishing gradient issue. This improved both the temporal and spatial dimension recognition rate. The characteristic of LSTMs to avoid long dependencies gives them the edge over basic RNN architectures for any sort of classification task and thus in this work, an LSTM based model architecture has been proposed. Most of these software implementations are not suitable for low power hardware deployment and even though there are LSTM accelerators for domain specific applications, to our knowledge there is no implementation of an energy-efficient LSTM hardware model for physical activity monitoring. The only justification we found for FPGA implementation concerning HAR wearable devices comes from [11] which addresses the adaptation issue of wearable HAR systems for low cost FPGAs.

III. Background Overview

A. Long Short Term Memory Networks (LSTM)

Long Short Term Memory (LSTM) network is the subset of the basic Recurrent Neural Network (RNN) architecture. Traditional RNNs can not process large sequences when tanh or ReLU activation function is used. To overcome this issue RNN networks have been evolved to have two different variations i.e. GRU (Gated Recurrent Unit) and

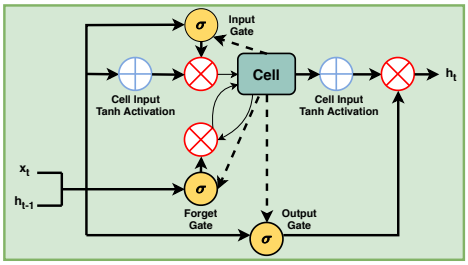

Fig. 2: An LSTM Cell which contains input, output, and forget gates. Here x_t is the input, h_{t-1} is the output of the previous state and h_t is the output of the current state.

LSTM. The main distinction between these two networks lies in the number of gates. GRU has two gates namely reset and update gates while LSTM networks consist of three gates i.e. input, output, and forget gate. This structural difference allows GRU networks to have fewer parameters, computations, and memory. However, with the increased number of gates LSTM networks have more control over accuracy. Besides, both these variants of the RNN network solve the vanishing gradient problem of basic RNN architectures. The structure of an LSTM unit is showed in Fig. 2. The input gate in the LSTM unit determines which memory content to add while forget gate figures out the previous cell state that is required for future inference. Though both these gates act in conjunction with each other, they are completely independent in terms of information correlation. Additionally, sigmoid and tanh activation functions are commonly used in LSTM networks.

B. Physical Activity Monitoring Dataset (PAMAP2)

The HAR dataset used for implementation is called PAMAP2 [12]. The PAMAP2 dataset contains 9 subjects with 12 different activities. The activities, for example, are lying, sitting, walking, running, etc. There are essentially 13 labels, one for each activity equaling the 12 activities in this case and the other label indicates the transient period between activities. The time-series signals were generated using three IMU (Inertial Measurement Units) sensors (gyroscope, accelerometer, magnetometer) and one heart rate monitor with a sampling frequency of 100 Hz

(IMU) and 9 Hz (Heart Rate Monitor) respectively. There are 52 channels but only 40 of these channels are valid according to [13]. Besides, only the first 8 subjects have significant data to perform classification and as a result subject 9 is not used during model optimization.

IV. Physical Activity Monitoring using LSTM

The model architecture takes raw time-series signals as input. But as part of preprocessing the raw signals are processed to be window images before they are fed to the model. The architecture determines correlations between sensor modalities using these window images of the raw data. Fig. 1 shows a high-level block diagram of the proposed system illustrating the deep neural network module for LSTM, and Fully Connected (FC) layers.

A. Signal Preprocessing

Raw time-series signals consist of F features with the same or different sampling frequency. To generate an image from the variables, a sliding window of size W and increment-step I is passed through all variables, creating a set of images of shape $1 \times W \times F$ (single channel image). The label associated with this image depends on the dataset. Since a single label is assigned to each image, the label of the current time step is taken as the label of the image (and the label that needs to be predicted subsequently while testing). A given image generated at time-step t has the prior states of each variable from $(t-W+1)...t$. Thus, the network can look back W prior states of each variable and given the current state of each variable, predicts the label.

B. Neural Network Architecture

The LSTM layer consists of 8 neurons and has a timestep of 64 for this case study. The other part of the input to the LSTM block is the feature which is the number of multimodal channels which is 40 for the PAMAP2 dataset. Three FC layers are utilized in this model architecture with the first and second ones containing 80 and 32 neurons, respectively. The last layer has a size equivalent to the class labels with a SoftMax activation. Tanh activation is used in the LSTM layer with a recurrent activation of hard sigmoid to counter the vanishing gradient issue and to memorize the cell states. Furthermore, FC layers are applied with ReLU activation logic which conforms to the linearity property for values greater than zero and outputs a zero for negative values. The network is optimized using the optimizer Adam which adapts the learning rate in relation to the gradient descent. Also, Categorical cross-entropy is used as the loss function. The rationale behind the number of neurons, LSTM layers, epochs and timesteps will be explained in detail in the next section.

V. Experimental Results and Model Optimization

For the experiment, the dataset for each subject was split to have 80% training data, 10% testing, and 10% validation data. In this section, the results for the experiments in relation to hyperparameter optimization have been discussed to reduce memory requirements while achieving high classification accuracy. The impact of changing the following parameters have been explored here: *1) Number of Epochs, 2) Number of filters in the LSTM layer, and 3) Choice of Timestep.*

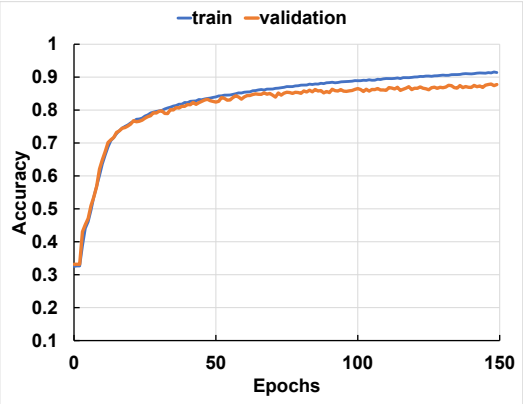

Fig. 3: Trend for classification accuracy over 150 epochs during training and validation.

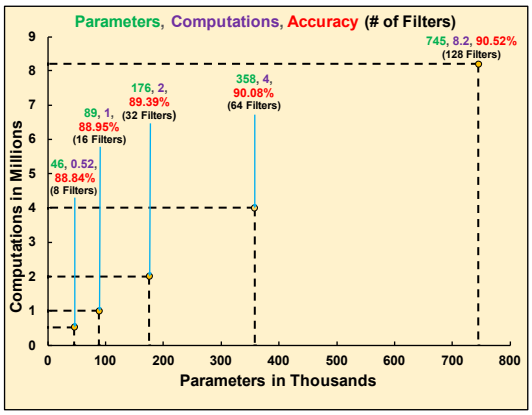

Fig. 4: Trend for computations, parameters and accuracy with different filter sizes. 8 filters provide suitable trade-off in this implementation.

A. Number of Epochs

The first step towards optimization involved figuring out the number of epochs required to train the model. In this case, we used a preliminary model without unoptimized parameters to see the trend for loss and accuracy at different epochs. The model was trained for 150 epochs while monitoring the validation results for accuracy and loss to find out the desired epoch value. According to this hypothesis, Fig. 3 shows that accuracy becomes stable after 100 epochs. Therefore, we decided to tune all our parameters for 100 epochs.

B. Number of filters in the LSTM layer

The number of filters for the LSTM layer is an important parameter that needs further attention. This is significant in the sense that the number of weights generated, and the amount of computations involved in the model is largely dependent on the choice of this parameter. Besides, the number of weights and computations directly affects memory and energy consumption, respectively. In this case,

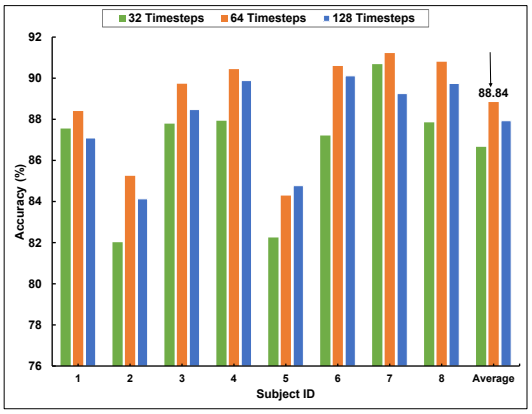

Fig. 5: Variation of accuracy for timestep size of 32, 64 and 128 for each subject. Timestep of 64 is optimal for this HAR case study.

the model was tested with several filter configurations. The higher the number of filters the better was the accuracy. However, with large filter sizes, the computations and parameters also increase which is not suitable for low power hardware implementation. Hence, an ideal filter size that presents good classification accuracy with a lower number of computations is required. From Fig. 4 it is evident that the choice of 8 filters provides the best trade-off (46k parameters, 0.52 million computations, 88.84% accuracy) for the memory and computation constraints involved.

C. Choice of Timestep

In deep neural networks that have LSTM modules, the temporal information hinges on the timestep. Timestep acts as the number of sequences the LSTM block looks back to decide for future prediction. Hence, the timestep is a particularly important parameter that determines the structure of the whole model. The choice of timestep varies from application to application and for human activity recognition problems, the timestep should be selected such that the relevant sequence of time-series information for each activity is available within the chosen timestep length. As mentioned in the previous sections, the sampling frequency for the sensors of this dataset was 100 Hz which means there are 100 samples in 1 second of data. Following this, if a timestep of size 128 is selected there will be overlap between sequences of information for corresponding classes which is not desired and with a timestep size of 32, the sequence will be too small for the LSTM model to make an accurate enough prediction. Thus, a timestep size of 64 for is chosen for this case study which provides the best accuracy given the constraints as evident by Fig. 5.

D. Model Weights Quantization

Reducing model size by applying quantization of the model weights is a popular method. Quantization reduces the complexity of the model and reuses the cache in a better and efficient way with the lower precision weights. Quantization is also power efficient as low precision data movement is more efficient than the higher precision data [14]. Therefore, 8-bit quantization was applied with the help

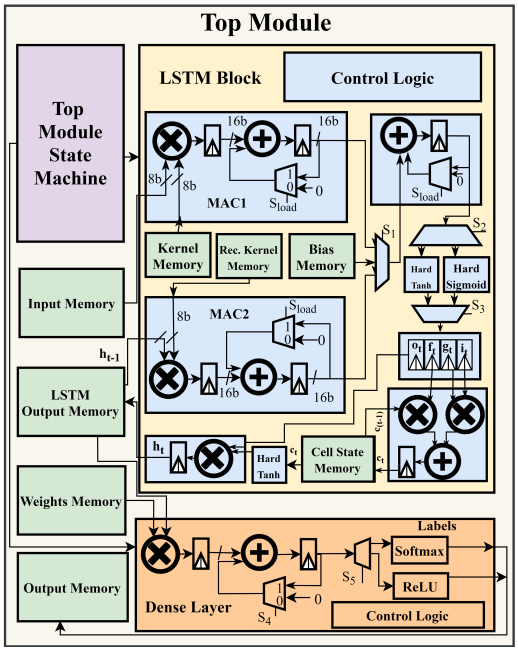

Fig. 6: Block diagram of hardware architecture which includes LSTM and dense layer block and also a top level state machine which controls all the blocks.

of tensorflow-lite post-training quantization. This resulted in an accuracy of 87.2% which represents a 1.7% drop than the original 16-bit model. The main task of quantizing a tensor is to calculate the two parameters scalar and displacement for value range mapping, by solving the set of equations:

$$float_{min} = (fixed_{min} - displacement)/scaler. \quad (1)$$

$$float_{max} = (fixed_{max} - displacement)/scaler. \quad (2)$$

The $fixed_{min}$ / $fixed_{max}$ are known from the target data type nature, but $float_{min}$ / $float_{max}$ are still needed to solve for scalar and displacement. The trained model only includes values in those weight tensors after a regular training, i.e. it only has min / max data for those tensors, so it can only be post-training weight quantized. Here floating Point weights are in 0.0 - 1.0 range and they are quantized to integer 8, so 0.0 becomes -128 and 1.0 becomes 127. However, from the **??** it can be seen that compared to 16 bit precision model, 8-bit precision model has less memory requirement.

VI. Hardware Architecture Design

Fig. 6 depicts the hardware architecture of our LSTM based human activity recognition algorithm. The main components of the hardware architecture includes the LSTM block, dense layer (FC), memories and the top module state machine. The Keras platform was utilized to perform offline training to obtain the trained weights. The floating-point arithmetic is complex and costly in terms of area, time and power in hardware and Keras by default gives weights in floating-point format. Therefore, the weights were converted into 8-bit fixed-point format. The input data for

TABLE I: Comparison of 16-bit precision and 8-bit precision hardware implementation

Data Precision/ Performance metrics	16 bit	8 bit	Improvement
Platform	Artix7 100t	Artix7 100t	-
Accuracy (%)	88.9	87.2	-
Frequency (MHz)	78.6	160	2 ×
Latency (ms)	1.4	0.8	1.8 ×
Power (mW)	144	82	1.8 ×
Energy (mJ)	207.4	67.2	3.1 ×
Energy Efficiency (GOP/s/W)	2.5	7.7	3.1 ×

the hardware model was also converted into 8-bit fixed point format. The **LSTM block** performs the LSTM layer operation. The computation needed in the LSTM layer is shown in [15] which is matrix-vector multiplication. For reducing the memory storage of weights, the kernel weights and recurrent kernel weights were concatenated into two different memories. Equations were merged in a way that it requires less computational block. As shown in Fig. 6, the LSTM block consists of two multiplication and accumulation (MAC) modules, one accumulator, few registers, multipliers, adders and four memories to store kernel weights, recurrent kernel weights, biases, and cell state memory. *Hard Tanh* and *Hard Sigmoid* activation functions are used dynamically inside this block in order to reduce the hardware complexity defined by the following equations:

$$hardtanh(x) = \begin{cases} -1, & x < -1 \\ x, & -1 \leq x \leq 1 \\ 1, & x > 1 \end{cases} \quad (3)$$

$$hardsigmoid(x) = \begin{cases} 0, & x < -2.5 \\ 0.2 \times x + 0.5, & -2.5 \leq x \leq 2.5 \\ 1, & x > 2.5 \end{cases} \quad (4)$$

The output of this block is stored into *LSTM Output Memory*.

The second component **Dense Layer** block performs fully-connected operations. FC layers connect every neuron in input to every neuron in output and it corresponds to one matrix-vector multiplication followed by a bias offset and an activation function. This block includes a MAC (multiply - accumulator) engine, a dynamic sorting logic for SoftMax and ReLU activation function. ReLU activation function was used in the first and second dense layer, and SoftMax activation in the output dense layer for the final classification task. All weights were stored in the weight memory. At the completion of the whole computation for the dense layers, the result was stored in *Output Memory* which overwrote the immediate data from previous dense layer.

VII. Hardware Implementation and Results

The hardware architecture was implemented on the Xilinx Artix-7 100t FPGA part. Additionally, the Xilinx Vivado tool 2018.2 was used to synthesize the design. Furthermore, the on-chip memories of the Artix–7 100t device was utilized to store the weights and feature map memory. The choice of Artix–7 FPGA came from the fact that the targetted avenue was low power embedded applications. Hence, energy efficiency and power consumptions are the two parameters that needed optimization, and Artix–7 FPGA is designed for low power operations which meets the restraints of the application. The hardware setup is implemented for both 16 bit and 8-bit precision to provide a point of comparison between the two. As evident by Table I, 8 bit precision achieves better results almost on all categories with a frequency of 160 MHz. The improvement comes from the fact that with low precision the design can run faster with reduced overhead for power and utilization. Besides, both the 8-bit and 16-bit precision design meets the latency deadline of 1 second. Table II shows a detailed comparison of the implementation with related human activity recognition hardware models. As evident by the table most of the previous work dedicated to physical monitoring concentrated on CNN based architectures. [16] introduces an accelerator for 3D convolution and [5] focuses on devising an energy-efficient model for multimodal data classification. According to the table, the latency achieved by this work is significantly faster compared to both models. This LSTM based model has an improvement of $18\times$ over [5] for latency which uses the same PAMAP2 dataset that is used in this work as a case study. The serial implementation results were compared from [5] because this LSTM implementation is serial as well. A parallelized implementation of this LSTM hardware will result in better performance results which is one of the future scopes of this work. Besides, the LSTM hardware design has a power consumption of 82 mW which is suitable for embedded applications along with a small energy consumption of 67.2 μJ. Another important parameter for hardware comparison is throughput and this design has an improvement of $43\times$ over [16] and $18.2\times$ over [5] for throughput. Even though the compared works have different architectures, energy efficiency was used as a figure of merit to make a distinction among them which takes into account the performance of a design over power consumption for the computations involved. This adds credence to the fact that this design is better with an energy efficiency of 7.7 which replicates to an improvement of $5.2\times$ over [16] and $22\times$ over [5]. Table III presents a comparative analysis of this work with existing LSTM hardware architectures. [17] represents the design called DeepRNN that balances the off-chip memory bandwidth and on-chip resources to attain high performance and scalability. Another work

TABLE II: Comparison of this work with previous work related to Human Activity Recognition (HAR) implemented on FPGA

	[5]	[16]	**This Work**
Application	Human Activity Recognition	Human Activity Recognition	Human Activity Recognition
Platform	Artix7 100t	Arria 10 SX660	Artix7 100t
Accuracy (%)	98	85	87.2
Frequency (MHz)	100	150	160
Latency (ms)	14.8	35.3	0.8
Throughput (label/s)	67.5	28.4	1220
Power (mW)	116	36,000	82
Energy (mJ)	1.7	1270	67.2
Energy Efficiency (GOP/s/W)	0.4	1.5	7.7

TABLE III: Comparison of this work with previous work related to LSTM models implemented on FPGA

	[17]-DeepRNN	[18]	[17]-DeepStore	[19]	**This Work**
Application	Language Modeling	Language Modeling	Language Modeling	EEG Artifact Detection	Human Activity Recognition
Platform	Xilinx Zynq XC7Z020	Xilinx Zynq XC7Z030	Xilinx Zynq XC7Z030	Artix7 100t	Artix7 100t
Latency (ms)	-	-	-	1.2	0.8
Throughput (ms)	-	-	-	807	1220
Frequency (MHz)	142	100	142	52.6	160
Power (mW)	1800	1190	2300	109	82
Energy Efficiency (GOP/s/W)	0.4	1.9 (average)	0.5	0.5	7.7

in [17] named DeepStore is implemented for language processing and utilizes the on-chip memory of the device to achieve low memory bandwidth whereas [18] introduces an RTL (Register Transfer Level) design for LSTM with the goal of low power implementation and throughput optimization. [19] proposed a hardware implementation of LSTM on programmable logic Artix-7 FPGA to detect EEG artifacts. This work edges the previous works in terms of energy efficiency significantly with an improvement of $19.2\times$, $4.1\times$, $16.7\times$ and $15.3\times$ over [17]-DeepRNN, [17]-DeepStore, [18] and [19] respectively. Besides, this implementation has an improvement of $1.5\times$ for both latency and throughput when compared to [19]. Along with this, the power consumption of 82 mW for our work is a considerable improvement over all these designs.

VIII. **Conclusion**

This work proposes and implements an RTL design for LSTM model architecture targetting physical activity monitoring with a classification deadline of one second. The model set forth for classification attains a classification accuracy of 87.2% with tuned hyperparameters for computation and memory optimization. Furthermore, the Verilog RTL demonstrates considerable improvement when compared to recent LSTM based hardware architectures. This is followed by the figure of merit in terms of energy efficiency which provides evidence that this work is at least $4.1\times$ more efficient than contemporary hardware RTL designs. In addition to these, the future goals will involve introducing ASIC performance results for the same architecture along with NVIDIA Jetson TX2 implementation of the model for energy efficiency on a GPU platform.

REFERENCES

[1] S. Seto *et al.*, "Multivariate time series classification using dynamic time warping template selection for human activity recognition," in *2015 IEEE Symposium Series on Computational Intelligence*, 2015.

[2] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[3] H.Ren *et al.*, "End-to-end scalable and low power multi-modal CNN for respiratory-related symptoms detection," in *2020 IEEE 33rd International System-on-Chip Conference (SOCC) (SOCC 2020)*, Sep. 2020, in press.

[4] M.Hosseini, H.Ren, H.Rashid, A.Mazumder, B.Prakash, and T.Mohsenin, "Neural networks for pulmonary disease diagnosis using auditory and demographic information," in *epiDAMIK 2020: 3rd epiDAMIK ACM SIGKDD International Workshop on Epidemiology meets Data Mining and Knowledge Discovery*. ACM, 2020, pp. 1–5, in press.

[5] A. Jafari *et al.*, "Sensornet: A scalable and low-power deep convolutional neural network for multimodal data classification," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 274–287, Jan 2019.

[6] A. Jafari, M. Hosseini, H. Homayoun, and T. Mohsenin, "A scalable and low power dcnn for multimodal data classification," in *2018 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. IEEE, 2018, pp. 1–6.

[7] N. Attaran *et al.*, "Embedded low-power processor for personalized stress detection," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. PP, no. 99, pp. 1–1, 2018.

[8] T. Abtahi *et al.*, "Accelerating convolutional neural network with fft on embedded hardware," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 9, pp. 1737–1749, Sept. 2018.

[9] M. Khatwani, M. Hosseini *et al.*, "Energy efficient convolutional neural networks for eeg artifact detection," in *2018 IEEE Biomedical Circuits and Systems Conference*, pp. 1–4.

[10] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, "Deep residual bidir-lstm for human activity recognition using wearable sensors," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[11] K. Basterretxea, J. Echanobe, and I. del Campo, "A wearable human activity recognition system on a chip," in *Proceedings of the 2014 Conference on Design and Architectures for Signal and Image Processing*. IEEE, 2014, pp. 1–8.

[12] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," 06 2012.

[13] A. Reiss and D. Stricker, "Creating and benchmarking a new dataset for physical activity monitoring," in *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments*, 2012, pp. 1–8.

[14] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," *arXiv preprint arXiv:1806.08342*, 2018.

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

[16] H. Fan, C. Luo, C. Zeng, M. Ferianc, Z. Que, S. Liu, X. Niu, and W. Luk, "F-e3d: Fpga-based acceleration of an efficient 3d convolutional neural network for human action recognition," in *2019 IEEE 30th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, vol. 2160. IEEE, 2019, pp. 1–8.

[17] A. X. M. Chang and E. Culurciello, "Hardware accelerators for recurrent neural networks on fpga," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–4.

[18] E. Azari and S. Vrudhula, "An energy-efficient reconfigurable lstm accelerator for natural language processing," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 4450–4459.

[19] H.-A. Rashid, N. K. Manjunath, H. Paneliya, M. Hosseini, and T. Mohsenin, "A low-power lstm processor for multi-channel brain eeg artifact detection," in *2020 21th International Symposium on Quality Electronic Design (ISQED), Accepted*. IEEE, 2020.