Optimizing Memory Footprint for Radar-Based Human Activity Recognition on Resource-Constrained Devices

Arash Heidari[1], Lorin Werthen-Brabants[2], Tom Dhaene[3], Ivo Couckuyt[4]
IDlab, Ghent University - imec, Ghent, Belgium
{[1]Arash.Heidari, [2]Lorin.WerthenBrabants, [3]Tom.Dhaene, [4]Ivo.Couckuyt}@UGent.be

***Abstract*— Human activity recognition is useful for tasks like patient fall detection. While radar offers privacy-preserving sensing, deploying predictive networks on edge devices faces challenges due to their memory requirements. Previous attempts to address this through hyperparameter tuning often lead to unwanted accuracy decline. On the other hand, pruning is used to reduce memory requirements of deep neural networks, but its effectiveness is limited for compact neural networks such as the Split BiRNN. The Split BiRNN architecture segregates the forward and backward computations across different components enabling quick initial predictions on the edge device and a refined prediction executed in the cloud. Our study enhances a prominent pruning algorithm to reduce memory requirements, making both models more compact. This not only reduces hardware costs but also enhances evaluation speed. Our findings suggest that pruning can sufficiently compact the network while maintaining its accuracy.**

***Keywords*— Human-Activity Recognition, Pruning, Memory Footprint Reduction, Radar Sensors.**

I. Introduction

Real-time Human Activity Recognition (HAR) in healthcare settings, particularly for geriatric patients, is crucial for prompt detection of incidents like falls, which can lead to severe injuries if not addressed promptly [1], [2]. While video surveillance is a commonly used option, it poses privacy concerns and is ineffective in low-light conditions. Wearable devices offer another solution but wearables may be forgotten or lost. Compact radar devices offer a cost-effective alternative. They preserve privacy, penetrate walls, and perform well in low-light conditions [3]. Radar systems transmit electromagnetic radio signals that will be reflected by objects. If these objects are moving, the frequency of the signal shifts. This phenomenon is known as the Doppler effect. By analyzing the time it takes for the reflected signals to return, along with their angles and speeds, a Micro-Doppler (MD) signature [4] can be generated. This signature provides a vector of values representing the speeds of moving objects detected by the radar. An example of Micro-Doppler is depicted in Figure 1.

Split BiRNNs [5], [6] handle multiple radar streams efficiently and enable real-time monitoring of human activities in diverse environments, such as hospitals with multiple rooms. A Split BiRNN operates in two stages:

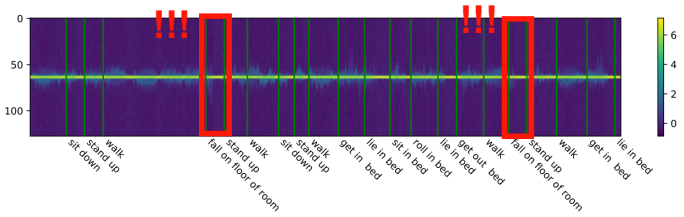

Fig. 1. An example of Micro-Doppler over time, with falling events marked with red rectangles. This is the data used to train the neural networks used in this paper.

1) **On the edge:** a lightweight model to compute class predictions on incoming radar frames, facilitating real-time streaming of results.
2) **In the cloud:** intermediate computations from the edge model are refined through a backward model and corrects inaccuracies.

The Split BiRNN technique, designed for deployment on edge devices, inherently leverages the advantages of decreased memory usage and energy consumption. Various methodologies, including Knowledge Distillation [7], Quantization [8], and Pruning [9], aim to further mitigate memory footprint and computational complexity. Knowledge Distillation involves training a smaller neural network, termed the Student, to emulate a larger reference network, the Teacher. However, its efficacy diminishes for smaller Teacher networks, prompting our focus on pruning techniques. In this study, we analyse different pruning techniques on small-scale neural networks targeting memory-constrained devices. Additionally, we refine an existing pruning technique to optimize memory utilization effectively.

II. Network Pruning

Pruning algorithms try to mitigate computational complexity, inference time, and memory footprint by selectively removing specific connections or neurons which contribute the least to the prediction of the neural network, as formally articulated by the following optimization problem:

$$\begin{aligned} \arg\min_{p} \quad & \mathcal{A}(\mathcal{N}(W)) - \mathcal{A}(\mathcal{N}_p(W_p)) \\ \text{where} \quad & \mathcal{N}_p(W_p) = P(\mathcal{N}(W)) \end{aligned} \tag{1}$$

where the pruning strategy $P(\cdot)$ seeks to modify an unpruned neural network $\mathcal{N}(W)$ characterized by weights W with minimum loss in efficiency, $\mathcal{A}(\cdot)$, conventionally quantified as the accuracy.

Pruning algorithms are broadly classified into static and dynamic pruning [10]. Static pruning involves post-training modifications to the neural network, selectively removing connections (*unstructured pruning*) or neurons (*structured pruning*), possibly followed by retraining to refine accuracy [11]. This type of pruning does not alter the network's behavior during inference. In contrast, dynamic pruning algorithms are designed to exert real-time control over the behaviour of the neural network by deactivating specific connections or neurons during the inference process. However, dynamic pruning methods typically incur a computational overhead, limiting their practicality for edge device implementation [10].

It is widely acknowledged that the magnitude of a weight exhibits a direct correlation with its significance in a neural network. Consistent with this observation, one prevalent pruning criterion involves the removal of small weights as follows [12]:

$$P_s(\mathcal{N}(W), T) = \mathcal{N}_p(W') \tag{2}$$
$$\text{where } W'_i = \begin{cases} W_i & \text{if } |W_i| > T \\ 0 & \text{otherwise} \end{cases}$$

Setting a high threshold value could significantly reduce accuracy, resulting in a sub-optimal pruned neural network. To mitigate this issue, it is advisable to include a retraining phase for the adjusted network weights [13]. Furthermore, iterative pruning has shown to yield better performance [14].

Alternatively, The Lottery Ticket Hypothesis (LTH) [15], argues that initial weight initialization plays a more crucial role in pruning. LTH trains the model until a desired accuracy is reached, followed by removing weights with least contribution. Remaining weights are reinitialized, the pruning threshold is increased, and the process iterates.

An alternative paradigm involves pruning neurons rather than connection-based pruning. Here, the objective is to reduce the final architectural footprint while upholding accuracy. As unstructured pruning yields sparsity within neural networks, it introduces computational complexities. In contrast, structured pruning does not cause similar computational complexities or sparsity issues.

III. Proposed Pruning Algorithm

The predominant focus of pruning algorithms has been directed towards mitigating the computational complexity associated with large-scale neural networks, and their utility when applied to smaller counterparts remains constrained. In light of these considerations, the present study introduces a modified pruning algorithm, augmented by a retraining process, specifically tailored to facilitate substantial connection pruning in context of HAR, detailed in Algorithm 1.

Algorithm 1 Proposed Magnitude-Based Pruning with Retraining P_r

Input Trained neural network $\mathcal{N}$ with weights W, pruning threshold T, outer retraining limit O, inner retraining limit I and threshold increment step δ.

$t \leftarrow$ a small value smaller than T (e.g. δ)
for w **in** W **do** ▷ Initial pruning step
  if $|w| < t$ **then**
    $w \leftarrow 0$
while $t < T$ **do**
  for $o \in \{1, ..., O\}$ **do** ▷ Outer retraining loop
    for $i \in \{1, ..., I\}$ **do** ▷ Inner retraining loop
      $W \leftarrow \text{Retrain}(\mathcal{N}(W))$
    for w **in** W **do** ▷ Intermediate pruning step
      if $|w| < t$ **then**
        $w \leftarrow 0$
  $t \leftarrow t + \delta$

In this algorithm, instead of removing or masking connections below the pruning threshold, they are directly assigned a zero value. During retraining, these connections remain active and can be updated or influence preceding layer connections. Retraining involves back-propagation for I iterations followed by pruning at each step. This process repeats for a total of O iterations followed by incrementally increasing the pruning threshold by δ until the threshold reaches its final value T. The proposed pruning algorithm relies on two pivotal hyperparameters, I and O, controlling the extent of pruned connections and resulting network accuracy. Parameter I determines the number of back-propagation iterations between each pruning operation, while O governs the total count of back-propagation and pruning phases before adjusting the pruning threshold, making the complexity of the algorithm an order of $I \cdot O$.

IV. Experiments

Experimental investigations are conducted on the trained Split BiRNN [5], the custom-designed neural network architecture tailored for human activity recognition, applied to the PARRad dataset [16]. The input data are Micro-Doppler over time, with 128 Doppler bins and a sampling frequency of 11.11Hz. The neural network model is split into two components: the *Edge model* exclusively employs unidirectional GRU layers, reducing memory usage for edge device implementation, while the *Cloud model* incorporates bidirectional GRU layers for enhanced accuracy, suitable for cloud execution. The architecture is depicted in Figure 2, consisting of convolutional and GRU layers, totaling 19,436 parameters, with 13,004 on the edge model and the rest on the cloud model. We focus on pruning techniques for the GRU and fully connected layers followed by a comparison of the retained accuracy across different pruning techniques.

We initiate our analysis by exploring two pruning methodologies. First, we explore the simple magnitude-based pruning without retraining. Then, we assess the impact of

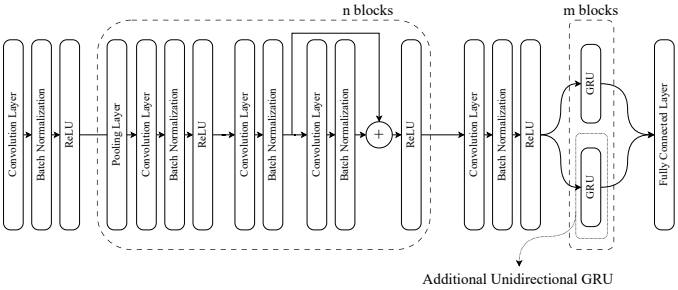

Fig. 2. Architecture of the split BiRNN model being used for HAR.

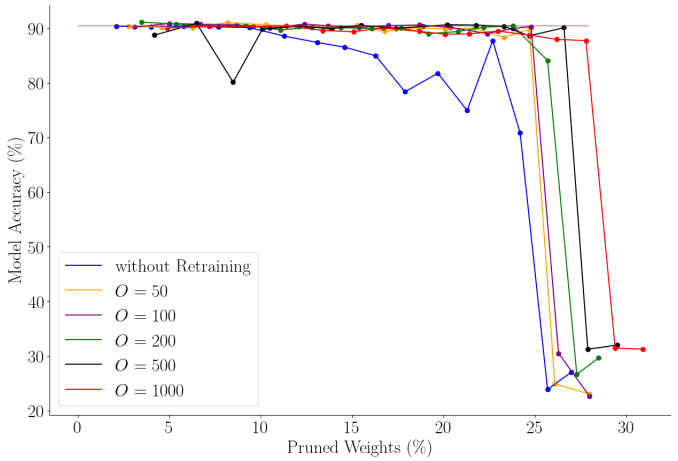

Fig. 3. Effect of O while $I = 1$ on the cloud model. The horizontal line designates the accuracy of the unpruned model. Each data point in the figure signifies the concluding outcomes for the respective pruning threshold.

pruning followed by retraining using Algorithm 1 with $I = 1$. In the latter approach, pruned weights are completely removed from the network, akin to conventional pruning techniques with retraining. To conduct these experiments, we initialize the pruning threshold t at 0.01, subsequently incrementing it by $\delta = 0.01$ at each step until the accuracy drops significantly. The obtained results are presented in Figure 3.

The experimental results confirm the expected trend: omission of retraining results in decreasing model accuracy as more weights are pruned. Increasing the parameter O allows for more extensive weight pruning before a significant accuracy drop occurs. Notably, when the pruning threshold reaches 0.15, model accuracy decreases to approximately 25%. This suggests that a threshold of $T = 0.14$ is the practical limit for $I = 1$. However, increasing O raises the risk of overfitting and may lead to decreased accuracy. Empirical findings indicate optimal performance of the pruning algorithm with $I = 1$ occurs at $O = 500$ and $T = 0.14$.

In the subsequent experiments, we maintain O at a fixed value of 100 and vary I to assess the pruning algorithm's performance. Obtained results are depicted in Figure 4. Increasing I leads to more pruned weights without accuracy loss, similar to increasing O. Notably, higher I values correspond to higher final pruning thresholds (T), with $I = 100$ achieving a threshold of 0.19. Despite this, the accuracy

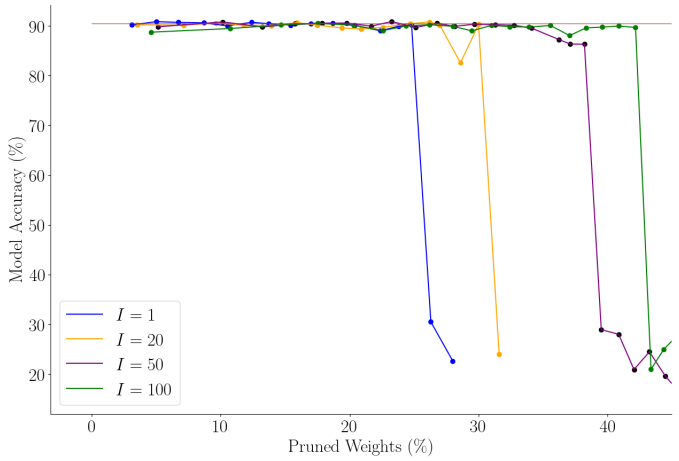

Fig. 4. Effect of I while $O = 100$ on the cloud model. The horizontal line designates the accuracy of the unpruned model. Each data point in the figure signifies the concluding outcomes for the respective pruning threshold.

trade-off remains minimal; for example, with $I = 100$, over 42% of connections in the cloud model can be pruned with negligible accuracy loss.

Our empirical results demonstrate that, with both I and O set to 100, a total of 8,198 connections can be pruned, while the remaining 11,223 connections exhibit superior performance compared to the unpruned edge model, albeit with a marginal accuracy reduction of 0.8% relative to the unpruned cloud model. Moreover, the pruned model features a reduced connection count in comparison to the unpruned edge model. Given that the model operates at a hardware level, the number of pruned connections exhibits a linear correlation with the ensuing memory footprint.

In order to provide a comprehensive evaluation, we incorporated two additional pruning algorithms for comparative analysis. The first algorithm, known as the Lottery Ticket Hypothesis and recognized as a state-of-the-art pruning technique, yielded a model accuracy of 12.2%. This result was obtained with only a relatively small fraction of weights pruned, specifically 17.8% of the connections, indicating that its suitability for the HAR task on small neural networks is severely constrained. We also explore the structured pruning, which, while less effective in terms of pruned connections compared to connection-level pruning, eliminates the need to preserve sparsity, simplifying computational overhead. However, structured pruning only pruned less than 10% of connections before a significant accuracy decline.

The comparative analysis of various methods is consolidated in Table 1. This table presents the accuracy alongside the percentage of pruned weights before reaching a notable decline in network accuracy. Additionally, an essential criterion, denoted as the *Pruning Effectiveness Index (PEI)*, is introduced, where $\Delta\mathcal{A} = \mathcal{A}(\mathcal{N}(W)) - \mathcal{A}(\mathcal{N}_p(W_p))$, n_p are the number of pruned weights and n the total number of weights of the original network:

Table 1. Comparison between state-of-the-art pruning techniques and proposed method

Method	Connections Pruned (%)		Accuracy (%)		PEI	
	Cloud	Edge	Cloud	Edge	Cloud	Edge
No Pruning	0.00	0.00	90.43	81.49	0.00	0.00
Lottery Ticket Hypothesis	12.13	8.97	71.67	63.18	0.10	0.07
Unstructured Pruning	22.70	20.40	87.72	74.77	0.22	0.19
Unstructured Pruning with Retraining	27.80	25.00	87.72	79.21	0.27	0.24
Structured Pruning	8.90	11.10	88.81	**80.40**	0.09	0.11
Proposed Method	**42.18**	**39.43**	**89.69**	80.18	**0.42**	**0.39**

$$\mathrm{PEI} = (1 - \Delta\mathcal{A})\frac{n_p}{n}. \tag{3}$$

This index represents a combined measure of both accuracy and the percentage of pruned weights. A higher value of this index signifies a more effective algorithm in achieving a harmonious balance between accuracy preservation and model size reduction.

V. Conclusion

In this study, we aimed to reduce the memory requirements of a neural network tailored for human activity recognition to enable its deployment on edge devices. We focused on static pruning methods due to their compatibility with edge devices, excluding dynamic pruning algorithms due to computational overhead. While some pruning methods, like the Lottery Ticket Hypothesis, were less effective, our novel approach of zeroing out pruned connections enhanced pruning efficacy with minimal accuracy loss.

While pruning significantly reduced the network size for edge compatibility, there is potential for further optimization, such as exploring quantization techniques. Additionally, optimizing edge-cloud interaction to reduce bandwidth requirements through efficient communication, like quantizing forward states from edge devices, presents an intriguing direction for future research.

Acknowledgments

This work has been supported by the Flemish Government under the 'Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen' and the 'Fonds Wetenschappelijk Onderzoek (FWO)' programmes.

References

[1] G. F. Fuller, "Falls in the Elderly," *American Family Physician*, vol. 61, no. 7, p. 2159, Apr. 2000, ISSN: 0002-838X, 1532-0650.

[2] X. Yu, "Approaches and principles of fall detection for elderly and patient," in *HealthCom 2008 - 10th International Conference on e-Health Networking, Applications and Services*, Jul. 2008, pp. 42–47. DOI: 10.1109/HEALTH.2008.4600107.

[3] M. Zhao, T. Li, M. A. Alsheikh, *et al.*, "Through-Wall Human Pose Estimation Using Radio Signals," en, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Jun. 2018, pp. 7356–7365, ISBN: 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00768.

[4] V. C. Chen, F. Li, S. S. Ho, and H. Wechsler, "Micro-Doppler effect in radar: Phenomenon, model, and simulation study," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 2–21, Jan. 2006, ISSN: 0018-9251. DOI: 10.1109/TAES.2006.1603402.

[5] L. Werthen-Brabants, G. Bhavanasi, I. Couckuyt, T. Dhaene, and D. Deschrijver, "Split birnn for real-time activity recognition using radar and deep learning," *Scientific Reports*, vol. 12, no. 1, p. 7436, 2022, ISSN: 2045-2322. DOI: 10.1038/s41598-022-08240-x. [Online]. Available: https://doi.org/10.1038/s41598-022-08240-x.

[6] L. Werthen-Brabants, G. Bhavanasi, I. Couckuyt, T. Dhaene, and D. Deschrijver, "Quantifying uncertainty in real time with split birnn for radar human activity recognition," in *2022 19th European Radar Conference (EuRAD)*, 2022, pp. 173–176. DOI: 10.23919/EuRAD54643.2022.9924932.

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531.

[8] Y. Guo, "A survey on methods and theories of quantized neural networks," *CoRR*, vol. abs/1808.04752, 2018. arXiv: 1808.04752. [Online]. Available: http://arxiv.org/abs/1808.04752.

[9] M. Augasta and T. Kathirvalavakumar, "Pruning algorithms of neural networks — a comparative study," *Open Computer Science*, vol. 3, no. 3, pp. 105–115, 2013. DOI: doi:10.2478/s13537-013-0109-x. [Online]. Available: https://doi.org/10.2478/s13537-013-0109-x.

[10] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.07.045. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221010894.

[11] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1510.00149.

[12] W. Lei, H. Chen, and Y. Wu, "Compressing deep convolutional networks using k-means based on weights distribution," in *Proceedings of the 2nd International Conference on Intelligent Information Processing*, ser. ICIIP '17, Bangkok, Thailand: Association for Computing Machinery, 2017, ISBN: 9781450352871. DOI: 10.1145/3144789.3144803. [Online]. Available: https://doi.org/10.1145/3144789.3144803.

[13] B. Choi, J.-H. Lee, and D.-H. Kim, "Solving local minima problem with large number of hidden nodes on two-layered feed-forward artificial neural networks," *Neurocomputing*, vol. 71, no. 16, pp. 3640–3643, 2008, Advances in Neural Information Processing (ICONIP 2006) / Brazilian Symposium on Neural Networks (SBRN 2006), ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2008.04.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231208002002.

[14] M. Paganini and J. Z. Forde, "On iterative neural network pruning, reinitialization, and the similarity of masks," *CoRR*, vol. abs/2001.05050, 2020. arXiv: 2001.05050. [Online]. Available: https://arxiv.org/abs/2001.05050.

[15] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=rJl-b3RcF7.

[16] G. Bhavanasi, L. Werthen-Brabants, T. Dhaene, and I. Couckuyt, "Patient activity recognition using radar sensors and machine learning," *Neural Computing and Applications*, vol. 34, no. 18, pp. 16033–16048, 2022.