Optimizing Vision Transformers: Leveraging Max and Min Operations for Efficient Pruning

This is the author's version of the article that has been presented at IEEE AICAS2024
The editorial version of the paper is available at http://dx.doi.org/10.1109/AICAS59952.2024.10595859

Optimizing Vision Transformers: Leveraging Max and Min Operations for Efficient Pruning

Philippe Bich*, Chiara Boretti*, Luciano Prono*, Fabio Pareschi*,‡,
Riccardo Rovatti†,‡, and Gianluca Setti§,‡
*DET, Politecnico di Torino, Italy - Email: {name.surname}@polito.it
†DEI, ‡ARCES, University of Bologna, Italy - Email: {riccardo.rovatti}@unibo.it
§CEMSE, King Abdullah University of Science and Technology (KAUST), Saudi Arabia - Email: gianluca.setti@kaust.edu.sa

***Abstract*—The research on Deep Neural Networks (DNNs) continues to enhance the performance of these models over a wide spectrum of tasks, increasing their adoption in many fields. This leads to the need of extending their usage also on edge devices with limited resources, even though, with the advent of Transformer-based models, this has become an increasingly complex task because of their size. In this context, pruning emerges as a crucial tool to reduce the number of weights in the memory-hungry Fully Connected (FC) layers. This paper explores the usage of neurons based on the Multiply-And-Max/min (MAM) operation, an alternative to the conventional Multiply-and-Accumulate (MAC), in a Vision Transformer (ViT). This enhances the model prunability thanks to the usage of Max and Min operations. For the first time, many MAM-based FC layers are used in a large state-of-the-art DNN model and compressed with various pruning techniques available in the literature. Experiments show that MAM-based layers achieve the same accuracy of traditional layers using up to 12 times less weights. In particular, when using Global Magnitude Pruning (GMP), the FC layers following the Multi-head Attention block of a ViT-B/16 model, fine-tuned on CIFAR-100, count only 560000 weights if MAM neurons are used, compared to the 31.4 million that remain when using traditional MAC neurons.**

I. Introduction

Deep Neural Networks (DNNs) have become the predominant tool for solving multiple tasks, ranging from Natural Language Processing [1] to Augmented Reality [2], [3], across various sectors such as medicine [4], robotics [5] and precision agriculture [6]. However, the increasing size of these neural models poses a challenge, especially with the growing demand to deploy these networks on resource-constrained edge devices.

In the realm of computer vision, Fully Connected (FC) networks were eclipsed by the introduction of Convolutional Neural Networks (CNN) that started with LeNet [7] and continued with more recent models like ResNet [8] and Inception [9]. Notably, CNNs are typically less resource-intensive in terms of memory footprint compared to FC-based networks. Nevertheless, the advent of Visual Transformers (ViTs) [10] in this domain has reintroduced the use of models containing predominantly FC networks, which are characterized by a large number of weights. To contain the memory footprint of these models, it is crucial to employ compression techniques, such as structured [11], [12] or unstructured pruning [13], [14].

The general idea behind unstructured pruning consists in the removal of weights from the model limiting the impairment of its approximation capabilities. This is typically achieved by assigning a score to each weight, with the lowest-scoring weights being removed. The score can be the magnitude of the weight or it may depend on its influence on the model's cost function (i.e., on the gradient with respect to the weight), involving also a dependency on the network's input. Score comparisons can be performed *locally* [15], i.e., among the interconnections of the same DNN layer, or *globally* [16], i.e., among the interconnections of the whole DNN model. Pruning techniques can be applied post-training [17], during training [18] or before training by analyzing the values of the weights after their initialization [19]. Some techniques, called one-shot methods [20], [21], can be applied to a pre-trained network without requiring further training. In contrast, other approaches necessitate multiple training cycles [16], [22] and, given the substantial training cost required for large neural networks, this factor could make some of these strategies, despite their efficiency, mostly impractical.

While many pruning strategies have been developed, only little effort has been put into changing the inner structure of the neurons to increase the performance of the already existing techniques. This concept has been explored in [23], where the authors introduced the Multiply-And-Max/min (MAM) neuron which leverages Max and Min operations to replace the summation used by traditional Multiply-and-ACcumulate (MAC) neurons to compute their output. In [23], the application of MAM neurons to a single layer of a custom FC autoencoder showed an accuracy similar to the one obtained with MAC neurons with an enhanced prunability of the network. Starting from that, in this paper:

- we evaluate the performance of MAM neurons for solving classical computer vision classification tasks using MNIST [7], Fashion-MNIST [24], and CIFAR-100 [25] datasets;
- we substitute multiple traditional FC layers with MAM-based FC layers in a state-of-the-art ViT model;
- we leverage the vanishing contributions technique introduced in [23] to avoid the retraining of the ViT model.
- we prune MAM neurons with methods in [17] together with additional state-of-the-art techniques such as [18] and [26] and we show how DNNs with MAM neurons outperform DNNs using only MAC neurons in terms of prunability;

The rest of the paper is structured as follows. In Section II, we provide a brief summary of the structure of MAM neurons. Then, in Section III, we present the pruning techniques used to prune both MAC and MAM neurons. In Section IV, we show

This is the author's version of the article that has been presented at IEEE AICAS2024
The editorial version of the paper is available at http://dx.doi.org/10.1109/AICAS59952.2024.10595859

the results on a simple FC-based network and on a ViT-B/16 model [10]. Finally, the conclusion is drawn.

II. The MAM neuron

In a standard DNN, the output of a traditional FC layer based on MAC neurons is defined as a column vector $\boldsymbol{y} \in \mathbb{R}^M$ and computed as

$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \tag{1}$$

$$\boldsymbol{y} = f(\boldsymbol{z}). \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is the input column vector, $\boldsymbol{W} \in \mathbb{R}^{M\times N}$ is the weights matrix, $\boldsymbol{b} \in \mathbb{R}^M$ is the bias column vector and $f(\cdot)$ is the activation function, which is applied element-wise to $\boldsymbol{z}$ to obtain the output vector $\boldsymbol{y}$. The MAM neuron, presented in [23], maintains the same map operation (i.e., multiply) contained in (1) that can be isolated by defining the *weighted inputs* matrix $\boldsymbol{V} \in \mathbb{R}^{M\times N}$ whose elements are evaluated as

$$v_{ij} = w_{ij}x_j \quad \text{with } i \in \{1,..,M\},\ j \in \{1,..,N\} \tag{3}$$

where v_{ij} and w_{ij} are the scalars at row i and column j of $\boldsymbol{V}$ and $\boldsymbol{W}$, respectively, and x_j is the j-th element of the input vector $\boldsymbol{x}$. With this notation, the equation (1) of a traditional MAC-based layer can be simply indicated as

$$z_i = \sum_{j=1}^{N} v_{ij} + b_i \quad \text{with } i \in \{1,..,M\} \tag{4}$$

where z_i is the i-th element of vector $\boldsymbol{z}$. Similarly, the output of a FC MAM-based layer is defined as

$$z_i = \mathop{\mathcal{M}}_{j=1}^{N} v_{ij} + b_i \quad \text{with } i \in \{1,..,M\} \tag{5}$$

where the reduce operation $\mathcal{M}$ is a suitable operator defined as

$$\mathop{\mathcal{M}}_{j=1}^{N} v_{ij} \triangleq \max_{j\in\{1,..,N\}} v_{ij} + \min_{j\in\{1,..,N\}} v_{ij}. \tag{6}$$

It is important to emphasize that, given an input, the output z_i of a MAM neuron depends only on two weights. In other words, whereas in traditional neurons the output is computed using all the inputs and all the weights, in a MAM neuron the output depends only on two inputs and two weights that are selected by Max and Min operators and may be different when changing the input. Yet, the probability of the weights being selected is not uniform and many weights may have a very low probability of contributing to the output. This intuition may stand as an intuitive explanation of why MAM neurons present better prunability properties. Figure 1 summarizes how, starting from the same matrix $\boldsymbol{V}$, MAC and MAM neurons compute their output.

A. The vanishing contributions method: a tool for integrating MAM neurons in large pre-trained neural models

During the training of MAM neurons, given a single input instance, for each row of matrix $\boldsymbol{V}$ only a couple of gradients are actually propagated backward. This dramatically slows down the training process. In [23], authors propose the *vanishing contributions technique* to speed up the training. This technique consists in starting the training with a standard

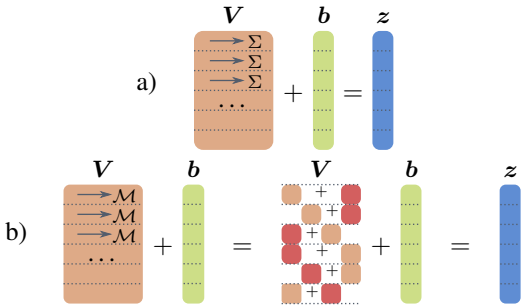

Fig. 1. Starting from the weighted input matrix $\boldsymbol{V}$, a MAC neuron computes its output z_i using the summation (a) while a MAM neuron uses the $\mathcal{M}$ operator (b) that sums only the maximum and the minimum values of each row of $\boldsymbol{V}$.

MAC-based neuron and gradually transitioning to a MAM-based one during the initial epochs. This approach enables more weight updates in the early training stages and it is controlled by a parameter β, initially set to 1 and linearly decreased, epoch by epoch, to reach $\beta = 0$. Consequently, the neuron output can be expressed during training with an affine combination of (4) and (5) as

$$z_i = \beta \sum_{j=1}^{N} v_{ij} + (1-\beta) \mathop{\mathcal{M}}_{j=1}^{N} v_{ij} + b_i \quad \text{with } i \in \{1,..,M\} \tag{7}$$

In this work we want to highlight that this technique not only has the advantage of accelerating the convergence of the network during training, but it makes the application of MAM neurons to pre-trained large neural models possible. More specifically, it enables the smooth substitution of standard MAC-based neurons in a pre-trained model with MAM-based ones, limiting the resources needed to retrain the DNN. This is of great use when dealing with structures that require a very long training process and a large amount of data, such as the models based on Transformers.

III. State-of-the-art pruning techniques

In this section, we introduce the pruning techniques used to compress both MAC and MAM neurons. All these methods are being applied in Section IV-A to prune a small and easily trainable FC network, to show that MAM neurons can rely on less weights compared to traditional neurons to obtain the same accuracy, even by using multiple different pruning approaches. However, only the compression strategies that do not require a complete training of the model are being applied to the ViT model in Section IV-B – the full training process of a ViT model is in fact impractical for most of its users. In selecting these techniques, we draw the methods from [17], supplementing them with some more recent approaches introduced in [18] and [26]. A summary of the selected methods follows.

1) Magnitude-based pruning: each weight is scored according to its magnitude. Following the idea that the smaller in magnitude a weight, the less it can influence the output, the values with the lower magnitude are pruned. We use two versions of this method, namely the Global Magnitude Pruning (GMP) and the Layer-wise Magnitude Pruning (LMP). With

This is the author's version of the article that has been presented at IEEE AICAS2024
The editorial version of the paper is available at http://dx.doi.org/10.1109/AICAS59952.2024.10595859

LGP all layers are constrained to be pruned by the same amount, while GMP has no constraints.

2) Gradient-based pruning: each weight is scored taking into consideration the statistics of the input data fed into the DNN. More precisely, given a set of inputs (i.e., the validation dataset), the score of a weight is computed as the average of the gradient of the loss function with respect to that weight multiplied by the magnitude of the weight. Similarly to GMP and LMP, we will use Global Gradient-based Pruning (GGP) and Layer-wise Gradient-based Pruning (LGP).

3) Alternating Compressed / DeCompressed Training: presented in [18], the AC/DC technique is a method to obtain a sparse model based on the training approach. With this, one selects in advance a percentage of weights to be removed and alternates, during training, phases in which the model uses all its weights with others where the model is pruned. The resulting sparse model is then further pruned using GMP. To use this technique with MAM neurons, we start from the obtained MAC sparse model, we convert the traditional neurons to MAM neurons, and we prune them with GMP. This technique requires the full training of the model.

4) Parameter-free Differentiable Pruning: PDP is a novel technique presented in [26], it proposes a training-time pruning scheme based on the usage of soft pruning masks that do not require the storage of additional learnable parameters. As for AC/DC, it requires a full training process.

IV. Experiments and Results

In this section, we present some comparisons between the pruning performance with MAC and MAM layers. For the comparisons, we use the standard computer vision datasets MNIST, Fashion-MNIST and CIFAR-100. First, we show the results on a small FC-based network that can be easily trained so that every pruning method presented in Section III can be applied. Then, we compare the performance of MAM and MAC neurons on ViT, using the pruning techniques that do not require full training of the model (GMP, LMP, GGP and LGP).

A. Simple FC-based network

The model is composed of two hidden FC layers containing 784 and 256 neurons, respectively, each followed by the ReLU activation function. These two layers are tested either with MAC-based or MAM-based neurons. The final classification layer (composed of 10 neurons) is kept MAC-based. The hidden layers contain 266 240 weights, which are 99% of the total number of learnable parameters.

We train the network on two different classical computer vision benchmarks, MNIST and FashionMNIST. These datasets contain 10 classes and 70 000 b/w images of size 28x28 each. Since both datasets comprise only one training set with 60 000 images and a test set with the remaining 10 000 images, we randomly remove 5000 images from the training set to create a validation set. Each pixel in the image is normalized between 0 and 1, and data augmentation is performed to improve the performance of the network. In particular, images are randomly shifted, rotated and scaled. The model undergoes training for 50 epochs, using Adam optimizer with a starting learning rate of 0.001 and cross-entropy loss. With MAM neurons, the first 5 epochs are used to complete the vanishing contributions transition. Table I shows the training results, where MAC and MAM neurons have a comparable performance.

TABLE I
Accuracy on the test set of the simple FC-based network when trained with MAC or MAM neurons

	MNIST		**Fashion-MNIST**	
	MAC	MAM	MAC	MAM
Accuracy	99.03%	99.02%	90.22%	90.03%

TABLE II
Percentage of remaining weights (at 3% accuracy loss, that is 96.03% for mnist and 87.22% for Fashion-mnist) in a simple FC network using MAM and MAC layers after pruning

	MNIST		**Fashion-MNIST**	
	MAC	MAM	MAC	MAM
GMP	34.23%	2.80%	38.14%	3.70%
LMP	31.83%	2.20%	38.14%	3.40%
GGP	19.92%	2.90%	39.04%	2.60%
LGP	19.62%	2.80%	47.35%	2.90%
AC/DC@90	7.61%	2.70%	7.61%	3.30%
AC/DC@95	2.90%	1.90%	3.40%	**2.01%**
PDP@85	-	-	14.31%	4.30%
PDP@90	7.11%	2.60%	-	-
PDP@95	3.60%	**1.88%**	-	-

After training, the models are pruned. LMP and GMP are the most straightforward pruning methods to apply since they do not require further training of the model or gradient evaluation. On the contrary, GGP and LGP require also the computation of the gradient, making them slightly more complex methods to apply. On average, the MAM model is compressed 12 times more compared to the MAC model with GMP and LMP and 11.3 times more with GGP and LGP. For example, considering that each weight occupies 4 bytes (single floating-point precision), the size of the entire model (i.e., also considering the non pruned last layer) trained on Fashion-MNIST and pruned with GMP, is of 51.8 KB when using MAM neurons while it is of 437.3KB when using MAC neurons.

AC/DC and PDP are training techniques that require the full training of the model with the goal of achieving a predefined target sparsity. The resulting sparse models obtained with AC/DC are then pruned with GMP while the ones obtained with PDP are then pruned with LMP as suggested by the original papers [18], [26]. We test AC/DC with a target sparsity of 90% and 95% on both MNIST and Fashion-MNIST. PDP is tested with 85% of sparsity on Fashion-MNIST and 90% and 95% on MNIST. Both AC/DC and PDP methods require to choose some hyperparameters. In this work, for AC/DC we choose to train the model for 50 epochs starting with 3 epochs of warm-up followed by alternating compressed and decompressed training phases (21 phases in total which last 2 epochs each) with 5 final epochs of fine-tuning. Similarly, for

This is the author's version of the article that has been presented at IEEE AICAS2024
The editorial version of the paper is available at http://dx.doi.org/10.1109/AICAS59952.2024.10595859

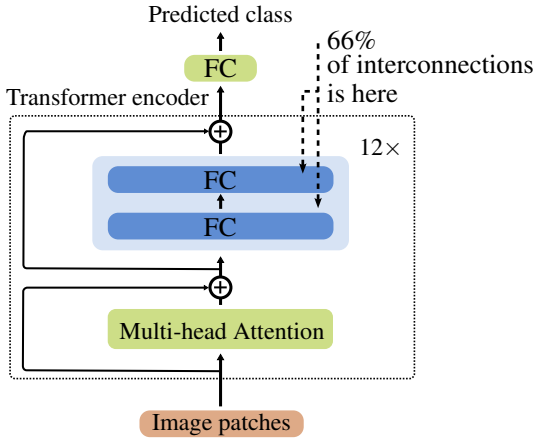

Fig. 2. Simplified representation of the ViT-B/16 model we use. We substitute the blue FC MAC-based layers with MAM-based layers in each encoder for a total of 24 MAM-based layers. These layers contain more than 56 million weights which are 66% of the total number of trainable parameters (which is 84 million).

PDP we train for 50 epochs with the last 10 epochs dedicated to fine-tuning (with $\epsilon = 0.05$ and $\tau = 0.0001$, more details in [26]). As reported in Table II, using these two techniques, the MAM-based model can be reduced on average 2.3 times more compared to the MAC-based one.

Table II shows the percentage of remaining weights in the hidden layers when the accuracy of the network is 3% below what is reported in Table I for MAC. The missing values in the table for PDP with 90% and 95% of target sparsity on Fashion-MNIST mean that these methods were not able to achieve the target accuracy. The table shows that, for any pruning method, MAM neurons have an advantage in terms of prunability compared to standard MAC neurons.

B. ViT-B/16 model

To assess the performance of MAM neurons in a state-of-the-art architecture, we select the ViT-B/16 model [10] represented in Figure 2. In each of the 12 encoders, we replace the two MAC-based FC layers following the Multi-head Attention block with two MAM-based layers. These layers contain 56.6 million parameters contributing 66% to the total number of learnable parameters of the model.

The MAC and the MAM-based models are both trained starting from the available MAC-based network[1] that has been pre-trained on ImageNet-21K [27]. We fine-tune this model on CIFAR-100 for 30 epochs. CIFAR-100 is a classical computer vision benchmark that consists of a total of 60 000 color images of size 32x32 divided into 10 classes. As for the other datasets, we randomly eliminate 5000 images from the training set and we use them as validation set. The same data augmentation applied to MNIST and Fashion-MNIST is performed. In the case of the MAM-based model, the initial 10 epochs are used to deploy the vanishing contributions transition. We use the Adam optimizer with a starting learning rate of $5{\cdot}10^{-5}$. At the end of the fine-tuning process, the MAC-based model achieves 92.14% of accuracy which is almost matched by the MAM-based model with 91.83%.

[1]We used the ViT-B/16 Hugging Face pre-trained model.

TABLE III
PERCENTAGE OF REMAINING WEIGHTS (AT 3% ACCURACY LOSS, THAT IS 89.14%) AND MEMORY OCCUPATION OF THE PRUNED FC LAYERS (VIT-B/16 MODEL)

	CIFAR-100			
	MAC		MAM	
	Remaining weights	Model size	Remaining weights	Model size
GMP	55.53%	125.9 MB	0.02%	1.2 MB
LMP	49.17%	111.5 MB	0.02%	1.2 MB
GGP	36.95%	84.1 MB	**0.01%**	**0.6 MB**
LGP	39.89%	90.05 MB	**0.01%**	**0.6 MB**

Since the full training of the model is not a viable option, we show the comparisons between MAC and MAM layers with the application of GMP, LMP, GGP and LGP only. The percentage of remaining weights in the pruned layers when the accuracy drops by 3% compared to the non-pruned MAC-based model is reported in Table III, together with the size in Megabyte of the FC layers. We highlight that, to account for the whole size of the DNN, the reader should consider also 115.4 MB introduced by the multi-head attention blocks. With any technique employed, the advantage of utilizing MAM-based layers is substantial, as the MAM-based pruned layers can maintain high the accuracy of the network with only 560 000 weights compared to the 20.7 million needed by MAC neurons to let the ViT-B/16 model achieve the same accuracy when pruned with GGP.

V. CONCLUSION

In this study, we have compared the prunability of MAM neurons to the one of traditional MAC neurons when pruned using state-of-the-art techniques. This comparison was conducted by integrating MAM-based layers into a simple FC-based neural network trained on classical computer vision datasets, namely MNIST and FashionMNIST. Furthermore, for the first time, MAM neurons were used in a Visual Transformer, namely ViT-B/16, which was fine-tuned on CIFAR-100. The vanishing contributions technique was also employed to prevent the impractical training from scratch of the transformer. Remaining weights in the pruned MAM and MAC-based layers together with the memory occupation of the model were measured in this context as well, highlighting the significantly greater prunability of MAM neurons.

ACKNOWLEDGMENT

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-Generation EU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

REFERENCES

[1] D. W. Otter, J. R. Medina, and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, Feb. 2021. doi:10.1109/TNNLS.2020.2979670

[2] Q. Cheng, S. Zhang, S. Bo, D. Chen, and H. Zhang, "Augmented Reality Dynamic Image Recognition Technology Based on Deep Learning Algorithm," *IEEE Access*, vol. 8, pp. 137 370–137 384, 2020. doi:10.1109/ACCESS.2020.3012130

[3] Y. Ghasemi, H. Jeong, S. H. Choi, K.-B. Park, and J. Y. Lee, "Deep learning-based object detection in augmented reality: A systematic review," *Computers in Industry*, vol. 139, p. 103661, Aug. 2022. doi:10.1016/j.compind.2022.103661

[4] A. Esteva *et al.*, "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 24–29, Jan. 2019. doi:10.1038/s41591-018-0316-z

[5] M. Mahdavian, P. Nikdel, M. TaherAhmadi, and M. Chen, "STPOTR: Simultaneous Human Trajectory and Pose Prediction Using a Non-Autoregressive Transformer for Robot Follow-Ahead," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 9959–9965. doi:10.1109/ICRA48891.2023.10160538

[6] M. Wakchaure, B. K. Patle, and A. K. Mahindrakar, "Application of AI techniques and robotics in agriculture: A review," *Artificial Intelligence in the Life Sciences*, vol. 3, p. 100057, Dec. 2023. doi:10.1016/j.ailsci.2023.100057

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998. doi:10.1109/5.726791

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. doi:10.1109/CVPR.2016.90

[9] C. Szegedy *et al.*, "Going Deeper With Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9. doi:10.1109/CVPR.2015.7298594

[10] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Jun. 2021, arXiv:2010.11929 [cs].

[11] Y. He, X. Zhang, and J. Sun, "Channel Pruning for Accelerating Very Deep Neural Networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 1398–1406. doi:10.1109/ICCV.2017.155

[12] Z. Chen, T.-B. Xu, C. Du, C.-L. Liu, and H. He, "Dynamical Channel Pruning by Conditional Accuracy Change for Deep Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 799–813, Feb. 2021. doi:10.1109/TNNLS.2020.2979517

[13] X. Chen, J. Zhu, J. Jiang, and C.-Y. Tsui, "Tight Compression: Compressing CNN Model Tightly Through Unstructured Pruning and Simulated Annealing Based Permutation," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, Jul. 2020, pp. 1–6. doi:10.1109/DAC18072.2020.9218701

[14] A. Aswani, C. R, and A. James, "Unstructured Weight Pruning in Variability-Aware Memristive Crossbar Neural Networks," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2022, pp. 3458–3462. doi:10.1109/ISCAS48785.2022.9937284

[15] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both Weights and Connections for Efficient Neural Network," in *28th International Conference on Neural Information Processing Systems (NIPS'15)*, vol. 28, Oct. 2015.

[16] J. Frankle and M. Carbin, "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks," in *7th International Conference on Learning Representations (ICLR 2019)*, Sep. 2018.

[17] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the State of Neural Network Pruning?" *Proceedings of Machine Learning and Systems*, vol. 2, pp. 129–146, Mar. 2020.

[18] A. Peste, E. Iofinova, A. Vladu, and D. Alistarh, "AC/DC: Alternating Compressed/DeCompressed Training of Deep Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 8557–8570.

[19] C. Wang, G. Zhang, and R. Grosse, "Picking Winning Tickets Before Training by Preserving Gradient Flow," in *11th International Conference on Learning Representations (ICLR)*, Feb. 2022.

[20] S. Zhang and B. C. Stadie, "One-Shot Pruning of Recurrent Neural Networks by Jacobian Spectrum Evaluation," in *International Conference on Learning Representations (ICLR 2020)*, Mar. 2020.

[21] T. Chen *et al.*, "Only Train Once: A One-Shot Neural Network Training And Pruning Framework," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 19 637–19 651.

[22] M. Zullich, E. Medvet, F. A. Pellegrino, and A. Ansuini, "Speeding-up pruning for Artificial Neural Networks: Introducing Accelerated Iterative Magnitude Pruning," in *2020 25th International Conference on Pattern Recognition (ICPR)*, Jan. 2021, pp. 3868–3875. doi:10.1109/ICPR48806.2021.9412705

[23] P. Bich, L. Prono, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "Aggressively prunable MAM²-based Deep Neural Oracle for ECG acquisition by Compressed Sensing," in *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Oct. 2022, pp. 163–167. doi:10.1109/BioCAS54905.2022.9948676

[24] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," Sep. 2017. doi:10.48550/arXiv.1708.07747 ArXiv:1708.07747 [cs, stat].

[25] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[26] M. Cho, S. Adya, and D. Naik, "PDP: Parameter-free differentiable pruning is all you need," in *ICML 2023 Workshop on Differentiable Almost Everything: Differentiable Relaxations, Algorithms, Operators, and Simulators*, 2023.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc., 2012.