Sparse Hybrid Linear-Morphological Networks

Konstantinos Fotopoulos[1], Christos Garoufis[2], Petros Maragos[1,2,3]
[1]Robotics Institute, Athena Research Center, 15125 Maroussi, Greece
[2]School of ECE, National Technical University of Athens, Greece
[3]HERON–Center of Excellence in Robotics, Athens, Greece
k.fotopoulos@athenarc.gr, cgaroufis@mail.ntua.gr, maragos@cs.ntua.gr

***Abstract*—We investigate hybrid linear-morphological networks. Recent studies highlight the inherent affinity of morphological layers to pruning, but also their difficulty in training. We propose a hybrid network structure, wherein morphological layers are inserted between the linear layers of the network, in place of activation functions. We experiment with the following morphological layers: 1) maxout pooling layers (as a special case of a morphological layer), 2) fully connected dense morphological layers, and 3) a novel, sparsely initialized variant of (2). We conduct experiments on the Magna-Tag-A-Tune (music auto-tagging) and CIFAR-10 (image classification) datasets, replacing the linear classification heads of state-of-the-art convolutional network architectures with our proposed network structure for the various morphological layers. We demonstrate that these networks induce sparsity to their linear layers, making them more prunable under L1 unstructured pruning. We also show that on MTAT our proposed sparsely initialized layer achieves slightly better performance than ReLU, maxout, and densely initialized max-plus layers, and exhibits faster initial convergence.**

***Index Terms*—mathematical morphology, morphological neural networks, neural network pruning, sparsity**

I. Introduction

Deep neural networks (DNNs) achieve state-of-the-art performance in domains like computer vision, natural language processing, and audio understanding. Their success stems from their ability to learn hierarchical representations from large datasets. However, as DNNs grow in size, they demand significant computational and memory resources, making deployment on edge devices challenging [4]. This has driven research into sparsity and pruning techniques [1].

Pruning techniques aim to remove redundant weights from a network while preserving its predictive accuracy. Various methods, such as structured and unstructured pruning, weight quantization, and low-rank approximations, have been explored to enhance efficiency. L1 unstructured pruning [16], for instance, removes weights based on their magnitude, leading to sparser models that require fewer computations. Despite the promise of such methods, designing inherently sparse neural networks remains an open challenge.

Mathematical morphology [11], [18], [22], [27] is a powerful set- and lattice-theoretic methodology, and provides a large variety of efficient nonlinear signal operators which have been widely used in signal processing for many tasks including denoising, connected filters, geometric feature extraction, representation, shape analysis, segmentation, remote sensing, object detection and recognition [10], [11], [18], [20], [22], [26]–[28]. Its arithmetic is based on max-plus and min-plus operations, called dilations and erosions, and is closely related to tropical algebra [2], [5]. These operations form the basis for morphological neural networks [23]–[25], which replace standard inner products with these nonlinear alternatives.

A key property of morphological layers is their high affinity to pruning. Since they emphasize extremal values rather than summations over many parameters, they inherently favor sparsity. Studies [6], [9], [31] show that morphological networks are able to maintain accuracy at significantly lower parameter counts. However, integrating morphological operations into neural net learning pipelines has been challenging due to their non-differentiability [7], [9], [23]. To address this, we propose a hybrid linear-morphological structure, wherein the activation functions, typically intertwined with fully connected linear layers, are replaced with explicitly sparse morphological layers. This design exploits the pruning effect invoked by morphological layers [6], maintaining trainability by avoiding over-dependence on them.

Contributions: Our contributions can be summarized as follows: 1) We propose replacing the ReLU activation layers of linear networks and/or the pooling layers of maxout networks by explicitly sparse morphological layers, creating a hybrid linear-morphological network that balances trainability and prunability. 2) By replacing the linear classification head of state-of-the-art convolutional networks [30] with the aforementioned network structure, and evaluating our approach on the Magna-Tag-A-Tune [14] (MTAT) and CIFAR-10 [13] datasets, i) we demonstrate that our proposed network is competitive with ReLU-based networks and improves upon maxout and max-plus block networks [31] (in fact, it achieves the best performance on MTAT) while also exhibiting faster initial convergence, and ii) more importantly, we show that our explicitly sparse max-plus block networks naturally induce sparsity in their linear layers, making them significantly more prunable under L1 unstructured pruning, and suitable for use cases where a lower parameter count is required.

Related work: Various works have explored morphological networks and their applications with varying levels of success. Some works have focused on replacing the max-pooling operations with morphological operations [7]. Others have

The research work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "2nd Call for H.F.R.I. Research Projects to support Faculty Members & Researchers" (Project Number:2656, Acronym: TROGEMAL).

Code is available on https://github.com/kostfoto/Sparse-Morph.git.

introduced hybrid linear-morphological architectures, with the aim of alleviating the problem of their training [21], [29], [31].

The most closely related works are [6], [31]. [31] introduces the max-plus block—a combination of a linear and morphological layer. They show performance gains over [3] and suggest potential for pruning. Our approach also forms max-plus blocks but differs in key ways: 1) Sparsity is explicitly imposed on our morphological layers, aiding in training and pruning. 2) We use a different topology; instead of replacing linear layers to keep parameter count constant—hindering test performance—we insert additional morphological layers in place of activations and/or maxout pooling, maintaining model performance. Unlike [31], which uses transfer learning on CIFAR, we train successfully from scratch. 3) [31] prunes the weights only of the morphological layer, thereby indirectly deactivating only some linear neurons, effectively proving that the morphological layers are prunable. In contrast, we prune primarily the weights of the linear layers, showing that the morphological layers induce sparsity to the linear layers.

[6] focuses primarily on pruning, demonstrating that morphological layers are highly prunable, disregarding performance. We, on the other hand, i) have given emphasis to the performance of the networks, by assuming a different topology and sparsely initializing the morphological layers, and ii) demonstrate that the inclusion of morphological layers makes the rest of the layers more prunable.

II. Preliminaries

Tropical (minmax) algebra studies the tropical semirings, encompassing both the max-plus and min-plus semirings [2], [5], [17], [19]. The max-plus semiring $(\mathbb{R}_{\max}, \vee, +)$ is the set $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$ equipped with the binary operations $\vee$ (maximum), and $+$ (ordinary addition), while the min-plus semiring $(\mathbb{R}_{\min}, \wedge, +)$ is the set $\mathbb{R}_{\min} = \mathbb{R} \cup \{+\infty\}$ equipped with the binary operations $\wedge$ (minimum) and $+$. Within tropical algebra we can define matrix operations. For example, for compatible matrices $\mathbf{A}, \mathbf{B}$, their max-plus multiplication $\boxplus$ is defined by $(\mathbf{A} \boxplus \mathbf{B})_{ij} = \bigvee_k a_{ik} + b_{kj}$, and their min-plus multiplication $\boxplus'$ is defined by $(\mathbf{A} \boxplus' \mathbf{B})_{ij} = \bigwedge_k a_{ik} + b_{kj}$, where $\bigvee_k$ and $\bigwedge_k$ denote maximum and minimum over k.

Mathematical morphology is well-defined on complete lattices [11], i.e. partially ordered sets in which every subset has a supremum and an infimum. Morphological operations map vectors and signals between complete lattices using two fundamental transformations: dilations (that distribute over suprema) and erosions (that distribute over infima). Shift-invariant signal dilations (erosions) correspond to nonlinear max-plus (min-plus) convolutions. These max-plus and min-plus operations connect mathematical morphology with tropical algebra.

In this paper, we focus on dilations and erosions defined on the set $\overline{\mathbb{R}}^n$ of finite discrete-time signals (i.e. vectors $\mathbf{x} = [x_i]$), where $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$. This forms a complete lattice when equipped with the partial order $\mathbf{x} \preceq \mathbf{y} \Leftrightarrow x_i \leq y_i, \forall i \in [n]$. For given weight vectors $\mathbf{w}, \mathbf{m} \in \mathbb{R}^n$ a dilation $\delta_{\mathbf{w}}$ and erosion $\varepsilon_{\mathbf{m}}$ from $\overline{\mathbb{R}}^n$ to $\overline{\mathbb{R}}$ can be defined as follows: $\delta_{\mathbf{w}}(\mathbf{x}) = \bigvee_{i\in[n]}(x_i + w_i) = \mathbf{w}^\top \boxplus \mathbf{x}$ and $\varepsilon_{\mathbf{m}}(\mathbf{x}) = \bigwedge_{i\in[n]}(x_i + m_i) = \mathbf{m}^\top \boxplus' \mathbf{x}$. A max-plus Morphological Perceptron (MP) [25] is simply a biased vector dilation, i.e. $\mathrm{MP}_{\mathbf{w}}(\mathbf{x}) = w_0 \vee \delta_{\mathbf{w}[1:]}(\mathbf{x}) = w_0 \vee \bigvee_{i\in[n]}(x_i + w_i)$. Similarly, a min-plus MP is a biased vector erosion. These perceptrons resemble linear perceptrons, but with summation replaced by max or min and multiplication replaced by ordinary addition. Another variant, the Dilation-Erosion Perceptron (DEP), trained using the Convex-Concave Procedure [3], takes a convex combination of a dilation and an erosion.

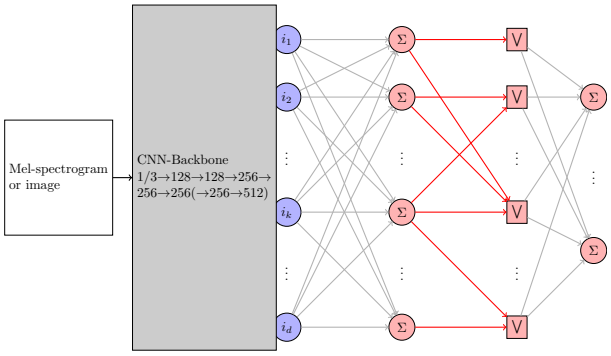

Fig. 1. Proposed network structure, preceded by a CNN backbone, for the case of MTAT and CIFAR; ReLU activations have been replaced by a sparse morphological layer. Each MP has on average 2 input weights (red).

MPs can be treated as building blocks for the construction of more complex networks, termed as Morphological Neural Networks (MNNs) [23], [24]. A max-plus MP-based network is recursively defined as $\mathbf{x}^{(n)} = f^{(n)}(\mathbf{w}_0^{(n)} \vee \mathbf{W}^{(n)} \boxplus \mathbf{x}^{(n-1)})$, where $\mathbf{w}_0^{(n)}$ is the bias vector, $\mathbf{W}^{(n)}$ is the weight matrix, and $f^{(n)}$ is an activation function.

III. Proposed method

Before we proceed with our proposed method, we study ReLU activated and maxout networks, showing how these can be expressed by combining a linear and a morphological layer.

ReLU networks: A fully connected ReLU activated network can be recursively defined as follows:

$$\mathbf{x}^{(n)} = \max(\mathbf{A}^{(n)}\mathbf{x}^{(n-1)} + \mathbf{b}^{(n)}, \mathbf{0}), \tag{1}$$

with the maximum operation applied element-wise. We have:

$$x_i^{(n)} = 0 \vee ((\mathbf{A}^{(n)}\mathbf{x}^{(n-1)})_i + b_i^{(n)}) \vee \bigvee_{j\neq i}((\mathbf{A}^{(n)}\mathbf{x}^{(n-1)})_j - \infty).$$

We may write this network in the following equivalent form:

$$\mathbf{y}^{(n)} = \mathbf{A}^{(n)}\mathbf{x}^{(n-1)},$$
$$\mathbf{x}^{(n)} = \mathrm{diag}_{\mathrm{mp}}(b_i^{(n)}) \boxplus \mathbf{y}^{(n)} \vee \mathbf{0} = \mathbf{W}_{\mathrm{ReLU}}^{(n)} \boxplus \mathbf{y}^{(n)} \vee \mathbf{0},$$

where $\mathrm{diag}_{\mathrm{mp}}$ denotes the max-plus diagonal matrix; i.e. off-diagonal elements equal to $-\infty$. The above form shows that any biased ReLU activated fully connected linear layer can be written as an unbiased linear layer, followed by a zero-biased max-plus morphological layer with diagonal weight matrix.

Maxout networks: A fully connected maxout network [8] with a pooling of K (number of affine terms) and N_n maxout units in the n-th layer can be recursively defined as follows:

$$\mathbf{x}^{(n)} = \max_{k\in[K]}(\mathbf{A}_k^{(n)}\mathbf{x}^{(n-1)} + \mathbf{b}_k^{(n)}),$$

where the maximum operation is applied element-wise. Define

$$\mathbf{A}^{(n)} = \left[(\mathbf{A}_1^{(n)})^\top, \cdots, (\mathbf{A}_K^{(n)})^\top\right]^\top,$$

$$\mathbf{w}^{(n)} = \left[(\mathbf{b}_1^{(n)})^\top, \cdots, (\mathbf{b}_K^{(n)})^\top\right]^\top.$$

Then, the network can equivalently be defined as follows:

$$\mathbf{g}^{(n)} = \mathbf{A}^{(n)}\mathbf{x}^{(n-1)} + \mathbf{w}^{(n)},$$
$$x_i^{(n)} = \max_{k\in[K]} (g_{i+(k-1)N_n}^{(n)}). \tag{2}$$

This form shows that maxout networks are effectively a linear layer followed by max-pooling. Notice that we may write

$$x_i^{(n)} = \max_{k\in[K]} ((\mathbf{A}^{(n)}\mathbf{x}^{(n-1)})_{i+(k-1)N_n} + w_{i+(k-1)N_n}^{(n)})$$
$$= \max_{j=i+(k-1)N_n} ((\mathbf{A}^{(n)}\mathbf{x}^{(n-1)})_j + w_j^{(n)})$$
$$\vee \max_{j\neq i+(k-1)N_n} ((\mathbf{A}^{(n)}\mathbf{x}^{(n-1)})_j - \infty)$$

Hence, we may write this network in the following form:

$$\mathbf{y}^{(n)} = \mathbf{A}^{(n)}\mathbf{x}^{(n-1)},$$
$$\mathbf{x}^{(n)} = \begin{pmatrix}\mathbf{W}_1^{(n)} & \cdots & \mathbf{W}_K^{(n)}\end{pmatrix} \boxplus \mathbf{y}^{(n)} = \mathbf{W}_{\text{maxout}}^{(n)} \boxplus \mathbf{y}^{(n)},$$

where

$$\mathbf{W}_k^{(n)} = \text{diag}_{\text{mp}}(b_{ki}^{(n)}).$$

The above form shows that any fully connected maxout linear layer can be written as an unbiased linear layer, followed by an unbiased max-plus morphological layer with a weight matrix formed by the concatenation of multiple diagonal matrices.

Max-plus block: Introduced in [31], the max-plus block (with the inclusion of a morphological bias) generalizes the aforementioned structures by relaxing the constrained forms of the weight matrix and bias. Analytically, a network of max-plus blocks can be recursively defined as follows:

$$\mathbf{y}^{(n)} = \mathbf{A}^{(n)}\mathbf{x}^{(n-1)},$$
$$\mathbf{x}^{(n)} = \mathbf{W}^{(n)} \boxplus \mathbf{y}^{(n)} \vee \mathbf{w}_0^{(n)}.$$

The weight matrices $\mathbf{A}^{(n)}, \mathbf{W}^{(n)}$ and bias $\mathbf{w}_0^{(n)}$ are general weight matrices and bias. From our previous analysis, we see that the max-plus block network is a generalization of ReLU and maxout networks, and hence it immediately follows that it is a universal approximator [31]. Notice that a max-plus block is effectively a maxout layer, where the linearities are shared across all outputs, achieving greater effective pooling.

Our method: Our proposal is twofold: 1) We propose a new means of constructing hybrid linear-morphological topologies, which maintains test performance and is amendable to pruning, and, more importantly, 2) we propose a new constraint on the form of the max-plus block, where the morphological layer is explicitly defined to be sparse.

Proposal 1): Most existing works focus on replacing linear layers with morphological layers, in order to keep parameter count constant. For example, in [31], the authors replace the final linear layer with a max-plus layer, turning a linear-ReLU-linear topology to a linear-ReLU-morphological one. By contrast, we leave the linear layers as are, and replace activations with additional morphological layers, turning linear-ReLU-linear topologies into linear-morphological-linear ones. Specifically, if we are given a linear-ReLU layer as defined in (1) with n_{out} outputs, i.e. $\mathbf{A}^{(n)} \in \mathbb{R}^{\text{n}_{\text{out}}\times\text{n}_{\text{in}}}$, then we get rid of the ReLU layer and the linear bias, and add a morphological layer with $\mathbf{W}^{(n)} \in \mathbb{R}^{\text{n}_{\text{out}}\times\text{n}_{\text{out}}}, \mathbf{w}_0^{(n)} \in \mathbb{R}^{\text{n}_{\text{out}}}$, forming a max-plus block. If we are given a maxout layer as defined in (2) with n_{out} outputs and pooling of K, i.e. $\mathbf{A}^{(n)} \in \mathbb{R}^{K\text{n}_{\text{out}}\times\text{n}_{\text{in}}}$, then we get rid of the maxout pooling layer and the linear bias, set a new linear layer with $\widetilde{\mathbf{A}}^{(n)} \in \mathbb{R}^{\text{n}_{\text{out}}\times\text{n}_{\text{in}}}$ and no linear bias, and a new morphological layer with $\mathbf{W}^{(n)} \in \mathbb{R}^{\text{n}_{\text{out}}\times\text{n}_{\text{out}}}, \mathbf{w}_0^{(n)} \in \mathbb{R}^{\text{n}_{\text{out}}}$, forming a max-plus block.

Proposal 2): The max-plus block of [31] defines a dense morphological layer, where each output has an effective pooling equal to the dimension of the input of the morphological layer. Instead, we propose that the morphological layer of the max-plus block be explicitly sparse. Specifically, for a morphological layer with n_{out} outputs and inputs (as defined in Proposal 1), we explicitly initialize all but $K \cdot \text{n}_{\text{out}}$ weights to $-\infty$, which renders them inactive throughout training. K is the effective pooling we wish each output to have on average. This means that we have the same effective pooling as a maxout network, but with a reduced linear layer of size n_{out} instead of $K \cdot \text{n}_{\text{out}}$ (see Proposal 1). In practice, we take $K = 2$. We should note that Proposal 2 eliminates the vast majority of additional parameters that Proposal 1 introduces.

Our method is illustrated in Fig. 1, where the ReLU activations of a linear-ReLU-linear topology have been replaced by a sparse morphological layer with $K = 2$. On average, each MP has 2 active inputs; in contrast to maxout layers, the number of active inputs differs throughout each unit, whereas the same input can be fed into multiple units.

TABLE I
TEST PERFORMANCE OF DIFFERENT METHODS ON MTAT

Method	ROC-AUC ↑	PR-AUC ↑
ReLU	0.9149 ± 0.0005	0.4632 ± 0.0023
Maxout	0.9148 ± 0.0003	0.4626 ± 0.0021
Zhang et al. [31]	0.5000 ± 0.0000	0.0653 ± 0.0000
Dense-Morph	0.9127 ± 0.0013	0.4553 ± 0.0010
Sparse-Morph (ours)	**0.9152** ± 0.0002	**0.4646** ± 0.0020

IV. EXPERIMENTS

Experimental Setup: Our goal is threefold: 1) to ensure that including morphological layers, particularly sparse ones, in state-of-the-art architectures does not degrade test performance, 2) to show that they contribute to accelerated training, and 3) to examine whether maxout, max-plus block, and our sparse max-plus block networks improve network prunability.

We conduct experiments on the MTAT dataset [14], a widely used benchmark in music tagging [30], containing 25,863 annotated 29-sec song excerpts. Following prior work [15], we evaluate on the top 50 tags using the default splits. We also evaluate our method on CIFAR-10 [13], which consists of 60,000 32×32 RGB images, distributed into 10 classes.

TABLE II
RESULTS ON CIFAR-10

Pruning ratio		Test Accuracy (%)			
r_2	r_1 (params)	ReLU	Maxout	Dense-Morph	Sparse-Morph
	Original	**78.98** ± 0.76	78.17 ± 0.86	77.78 ± 0.72	78.32 ± 0.50
0.7	0.7 (41380)	69.57 ± 3.38	34.89 ± 6.62	29.83 ± 11.16	**74.13** ± 2.14
0.7	0.8 (28273)	55.03 ± 6.49	23.25 ± 4.86	21.61 ± 8.35	**66.10** ± 4.02
0.7	0.9 (15166)	27.78 ± 5.84	15.41 ± 5.24	17.14 ± 4.02	**34.16** ± 10.41
0.8	0.7 (40868)	62.36 ± 3.83	34.42 ± 4.77	24.83 ± 9.63	**72.40** ± 2.33
0.8	0.8 (27761)	43.33 ± 5.99	22.08 ± 3.79	19.66 ± 8.48	**62.51** ± 4.60
0.8	0.9 (14654)	20.54 ± 6.58	14.24 ± 4.31	17.03 ± 6.46	**30.73** ± 8.90
0.9	0.7 (40356)	44.77 ± 4.86	35.34 ± 3.23	19.61 ± 6.45	**70.31** ± 3.02
0.9	0.8 (27249)	26.80 ± 4.56	23.76 ± 1.42	14.68 ± 4.34	**58.20** ± 7.24
0.9	0.9 (14142)	14.91 ± 3.93	15.85 ± 3.78	13.93 ± 3.81	**28.57** ± 12.26
0.95	0.7 (40100)	36.49 ± 1.45	32.21 ± 2.39	22.11 ± 7.63	**67.36** ± 2.85
0.95	0.8 (26993)	22.53 ± 4.74	23.85 ± 2.78	17.42 ± 5.79	**54.92** ± 5.93
0.95	0.9 (13886)	13.27 ± 3.04	16.87 ± 5.54	14.53 ± 3.92	**25.88** ± 7.63

Our base model is the short-chunk CNN [30], which processes log-mel spectrograms (3.69 sec length – 96 mel bands – 512-sample windows, 256-sample hop) and consists of a 7-layer convolutional backbone followed by a two-layer linear-ReLU-linear classification head, which we replace with a linear-morphological-linear design. The first FC layer (512 neurons) receives a 512-dimensional input, and the final layer has 50 output neurons. The intermediate morphological layer has size 512×512 with mostly inactive parameters. For CIFAR, we reduce the number of convolutional layers to five.

We compare five classification heads: 1) ReLU-based MLP – a standard linear-ReLU-linear structure as in [30], 2) Maxout-based MLP – replacing the first ReLU layer with a maxout layer (pooling factor $K = 2$), 3) Zhang et al. [31] - replacing the final linear layer with a morphological layer, as proposed in [31] for their CIFAR architecture, 4) Dense-Morph-based MLP, replacing the first linear-ReLU layer with a dense linear-morphological layer (Proposal 1), and 5) Sparse-Morph-based MLP, which further constrains the morphological layer to be sparse (Proposals 1 & 2). We incorporated Batch Normalization in all classification heads, with the exception of (4) on MTAT, where its removal led to smoother training.

Training is performed from scratch. For CIFAR-10 we train using Adam for 10 epochs with a learning rate of 0.001 and a random 80-20 train-validation split. For MTAT we train for 100 epochs, using Adam [12] with a learning rate of 0.0001 for the first 80 epochs, and SGD with Nesterov momentum (0.9) and a learning rate of 0.001 for 20 epochs. Weight decay is set to 0.0001. The model with the lowest validation loss is selected for testing. We train 5 models for each method, reporting the mean and standard error of the ROC-AUC, PR-AUC scores (MTAT) or classification accuracy (CIFAR-10) in the testing split; for MTAT, excerpt-wise scores are obtained by averaging the per-spectrogram network outputs.

Test Performance: First, we verify that the inclusion of morphological layers does not significantly degrade test accuracy. We report on the results of fully-trained, non-pruned networks in MTAT. Table I shows that models that follow Proposal 1 achieve comparable results to ReLU and Maxout networks. In addition, the model following both Proposals 1 and 2, i.e. our Sparse-Morph-based, marginally achieves the best performance out of all the models.

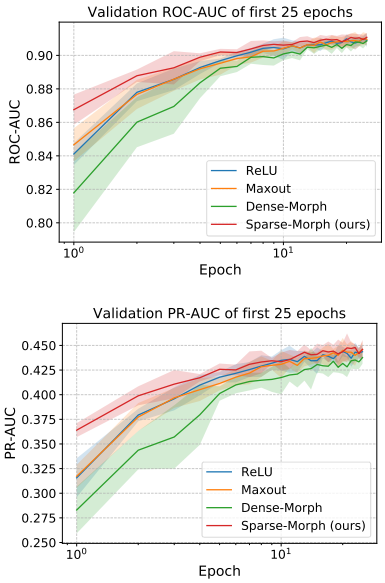

Fig. 2. Error plot of validation ROC-AUC and PR-AUC scores of different methods on MTAT for the first 25 epochs of training; x-axis in log-scale.

Pruning Experiments: We evaluate prunability via unstructured L1 pruning on the FC layers. Networks following Proposal 1 are pruned based on L1 norm (linear layers) or absolute magnitude (morphological layers, with pruned weights set to $-\infty$). We prune the last linear layer with a pruning ratio of r_2 and the remaining layers of the classification head with r_1. To ensure equal parameter counts, we make the following adjustments: 1) For the maxout network, whose linear layer is twice the size of the ReLU network, we prune at a pruning ratio $r_1' = 1 - (1 - r_1)/2$, and an additional 512 parameters due to the biases. 2) For the Dense-Morph network, which has 1 linear and 1 morphological FC layer, we prune each at a pruning ratio $r_1' = 1 - (1 - r_1)/2$. 3) For the Sparse-Morph network, which has $2 \cdot 512$ additional parameters, we prune this many additional parameters from the first linear layer.

Tables II & III show that our sparse max-plus block networks achieve significantly better pruning performance than ReLU networks. This also holds true for the Maxout and Dense-Morph networks in MTAT (where Sparse-Morph network performs best in ROC-AUC, while Maxout and Sparse-Morph achieve similar PR-AUC scores), as well as in CIFAR under high pruning ratios; for other ratios, ReLU outperforms other morphological variants. Since weights are mostly (in fact, for the Maxout and Sparse-Morph networks, solely) pruned from the preceding linear layers, it is implied that morphological layers are not only sparse themselves but also induce sparsity in adjacent linear layers.

Speed of Convergence: Lastly, we compare the initial convergence speed of our models (Fig. 2) in MTAT. We observe that our Sparse-Morph model achieves faster validation score improvements in the first 25 epochs; in contrast, Dense-Morph is slower to converge than the ReLU-based baseline. This could prove useful in hyperparameter optimization, where the models are trained only for a few epochs.

TABLE III
PRUNING PERFORMANCE OF DIFFERENT METHODS FOR A VARIETY OF PRUNING RATIOS r_1, r_2 ON MTAT

Pruning ratio		ReLU		Maxout		Dense-Morph.		Sparse-Morph. (ours)	
r_2	r_1 (#params)	ROC-AUC ↑	PR-AUC ↑	ROC-AUC ↑	PR-AUC ↑	ROC-AUC ↑	PR-AUC ↑	ROC-AUC ↑	PR-AUC ↑
0.8	0.8 (58111)	0.9048 ± 0.0006	0.4278 ± 0.0014	0.9116 ± 0.0014	0.4530 ± 0.0022	0.9023 ± 0.0043	0.4304 ± 0.0046	**0.9125** ± 0.0008	**0.4585** ± 0.0026
	0.9 (31897)	0.9045 ± 0.0006	0.4247 ± 0.0026	0.9100 ± 0.0015	0.4501 ± 0.0023	0.9005 ± 0.0034	0.4232 ± 0.0043	**0.9119** ± 0.0009	**0.4563** ± 0.0022
	0.95 (18790)	0.9034 ± 0.0006	0.4212 ± 0.0047	0.9068 ± 0.0010	0.4440 ± 0.0021	0.8979 ± 0.0031	0.4171 ± 0.0045	**0.9102** ± 0.0012	**0.4514** ± 0.0035
	0.98 (10925)	0.8995 ± 0.0011	0.4099 ± 0.0069	0.8988 ± 0.0015	0.4282 ± 0.0050	0.8939 ± 0.0034	0.4090 ± 0.0059	**0.9058** ± 0.0019	**0.4397** ± 0.0052
0.9	0.8 (55551)	0.8911 ± 0.0024	0.4039 ± 0.0049	0.9000 ± 0.0011	0.4466 ± 0.0024	0.8914 ± 0.0068	0.4128 ± 0.0072	**0.9074** ± 0.0028	**0.4480** ± 0.0039
	0.9 (29337)	0.8901 ± 0.0033	0.4001 ± 0.0072	0.8983 ± 0.0013	0.4427 ± 0.0028	0.8910 ± 0.0054	0.4096 ± 0.0053	**0.9064** ± 0.0030	**0.4454** ± 0.0033
	0.95 (16230)	0.8892 ± 0.0029	0.3961 ± 0.0082	0.8950 ± 0.0010	0.4363 ± 0.0020	0.8899 ± 0.0045	0.4067 ± 0.0045	**0.9046** ± 0.0032	**0.4401** ± 0.0047
	0.98 (8365)	0.8836 ± 0.0048	0.3807 ± 0.0106	0.8885 ± 0.0011	0.4214 ± 0.0036	0.8871 ± 0.0045	0.4004 ± 0.0061	**0.8993** ± 0.0041	**0.4258** ± 0.0040
0.95	0.8 (54271)	0.8647 ± 0.0017	0.3630 ± 0.0034	0.8943 ± 0.0011	**0.4392** ± 0.0030	0.8730 ± 0.0083	0.3938 ± 0.0087	**0.8965** ± 0.0058	0.4325 ± 0.0037
	0.9 (28057)	0.8627 ± 0.0041	0.3585 ± 0.0060	0.8923 ± 0.0009	**0.4344** ± 0.0035	0.8730 ± 0.0064	0.3919 ± 0.0071	**0.8952** ± 0.0061	0.4292 ± 0.0033
	0.95 (14950)	0.8602 ± 0.0047	0.3477 ± 0.0092	0.8882 ± 0.0013	**0.4271** ± 0.0047	0.8725 ± 0.0052	0.3892 ± 0.0059	**0.8930** ± 0.0066	0.4223 ± 0.0039
	0.98 (7085)	0.8506 ± 0.0061	0.3289 ± 0.0150	0.8789 ± 0.0019	**0.4072** ± 0.0042	0.8692 ± 0.0037	0.3831 ± 0.0051	**0.8866** ± 0.0074	0.4040 ± 0.0039
0.98	0.8 (53503)	0.7820 ± 0.0083	0.2732 ± 0.0044	0.8398 ± 0.0101	0.3919 ± 0.0086	0.8058 ± 0.0129	0.3401 ± 0.0149	**0.8737** ± 0.0072	**0.3951** ± 0.0086
	0.9 (27289)	0.7797 ± 0.0092	0.2680 ± 0.0063	0.8361 ± 0.0100	0.3836 ± 0.0057	0.8061 ± 0.0120	0.3402 ± 0.0114	**0.8719** ± 0.0077	**0.3917** ± 0.0088
	0.95 (14182)	0.7755 ± 0.0069	0.2606 ± 0.0119	0.8302 ± 0.0088	0.3736 ± 0.0063	0.8060 ± 0.0109	0.3382 ± 0.0088	**0.8683** ± 0.0081	**0.3807** ± 0.0080
	0.98 (6317)	0.7695 ± 0.0057	0.2479 ± 0.0093	0.8147 ± 0.0094	0.3462 ± 0.0095	0.8033 ± 0.0103	0.3317 ± 0.0063	**0.8584** ± 0.0084	**0.3565** ± 0.0084

V. CONCLUSION

We explored hybrid linear-morphological architectures and introduced a method to integrate max-plus blocks while preserving test performance. By replacing activation layers with morphological layers and sparsifying them via $-\infty$ initialization, we achieved the best test performance in MTAT and faster initial convergence, More importantly, we show that morphological layers induce sparsity in linear components, making hybrid networks—including maxout and max-plus variants—more prunable. Our findings highlight the potential of morphological layers for improving efficiency, convergence, and compression in neural networks; future work could focus in applying these principles under large-scale settings.

ACKNOWLEDGMENTS

This project is funded by the European Union under Horizon Europe (grant No. 101136568 - HERON).

Funded by
the European Union

REFERENCES

[1] D. Blalock, J.J. Gonzalez Ortiz, J. Frankle, and J. Guttag. What is the State of Neural Network Pruning? In *Proc. of Machine Learning and Systems 2*, 2020.

[2] P. Butkovič. *Max-linear Systems: Theory and Algorithms*. Springer, 2010.

[3] V. Charisopoulos and P. Maragos. Morphological Perceptrons: Geometry and Training Algorithms. In *Proceedings ISMM*. Springer, 2017.

[4] J. Chen and X. Ran. Deep Learning With Edge Computing: A Review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019.

[5] R. Cuninghame-Green. *Minimax Algebra*. Springer-Verlag, 1979.

[6] N. Dimitriadis and P. Maragos. Advances in Morphological Neural Networks: Training, Pruning and Enforcing Shape Constraints. In *Proc. Int'l Conf. Acoust., Speech, Signal Processing*, Toronto, 2021.

[7] G. Franchi, A. Fehri, and A. Yao. Deep Morphological Networks. *Pattern Recognition*, 102:107246, 2020.

[8] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *Proc. Int'l Conf. Machine Learning (ICML)*, 2013.

[9] R. Groenendijk, L. Dorst, and T. Gevers. Geometric Back-Propagation in Morphological Neural Networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 45(11):14045–14051, Nov. 2023.

[10] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume I. Addison-Wesley, 1992.

[11] H.J.A.M. Heijmans. *Morphological Image Operators*. Acad. Press, Boston, 1994.

[12] D. P. Kingma and J. Lei Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015.

[13] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images, 2009.

[14] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie. Evaluation of algorithms using games: The case of music tagging. In *ISMIR*, pages 387–392. Citeseer, 2009.

[15] J. Lee, J. Park, K. Luke Kim, and J. Nam. SampleCNN: End-to-end Deep Convolutional Neural Networks using Very Small Filters for Music Classification. *Applied Sciences*, 8(1):150, 2018.

[16] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. In *Proc. ICLR*, 2017.

[17] D. Maclagan and B. Sturmfels. *Introduction to Tropical Geometry*. Amer. Math. Soc., 2015.

[18] P. Maragos. Morphological Filtering for Image Enhancement and Feature Detection. In A. C. Bovik, editor, *Image and Video Processing Handbook*, pages 135–156. Elsevier Acad. Press, 2 edition, 2005.

[19] P. Maragos, V. Charisopoulos, and E. Theodosis. Tropical geometry and machine learning. *Proceedings of the IEEE*, 109(5):728–755, May 2021.

[20] F. Meyer. *Topographic Tools for Filtering and Segmentation 1 & 2*. Wiley, 2019.

[21] R. Mondal, S. Santra, S.S. Mukherjee, and B. Chanda. Morphological Network: How Far Can We Go with Morphological Neurons? In *Proc. British Machine Vision Conference (BMVC)*, 2022.

[22] L. Najman and H. Talbot, editors. *Mathematical Morphology: From Theory to Applications*. Wiley-ISTE, 2010.

[23] L. F.C. Pessoa and P. Maragos. Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition. *Pattern Recogn.*, 33:945–960, 2000.

[24] G. X. Ritter and P. Sussner. An Introduction to Morphological Neural Networks. In *Proc. Int'l Conf. Pattern Recognition (ICPR)*, 1996.

[25] G. X. Ritter and G. Urcid. Lattice algebra approach to single-neuron computation. *IEEE Trans. Neural Networks*, 14(2):282–295, Mar. 2003.

[26] P. Salembier and M. H.F. Wilkinson. Connected Operators: A review of region-based morphological image processing techniques. *IEEE Signal Processing Magazine*, pages 136–157, Nov. 2009.

[27] J. Serra. *Image Analysis and Mathematical Morphology*. Acad. Press, New York, 1982.

[28] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer, 2nd edition, 2004.

[29] M. E. Valle. Reduced Dilation-Erosion Perceptron for Binary Classification. *Mathematics*, 8(4):512, 2020.

[30] M. Won, A. Ferraro, D. Bogdanov, and X. Serra. Evaluation of CNN-based automatic music tagging models. In *Proc. SMC 2020*, 2020.

[31] Y. Zhang, S. Blusseau, S. Velasco-Forero, I. Bloch, and J. Angulo. Max-Plus Operators Applied to Filter Selection and Model Pruning in Neural Networks. In *Proceedings ISMM*. Springer, 2019.