ADVANCES IN MORPHOLOGICAL NEURAL NETWORKS: TRAINING, PRUNING AND ENFORCING SHAPE CONSTRAINTS

Nikolaos Dimitriadis[1,*] *and Petros Maragos*[2]

[1] École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[2] School of ECE, National Technical University of Athens (NTUA), Athens, Greece
[1]nikolaos.dimitriadis@epfl.ch , [2]maragos@cs.ntua.gr

ABSTRACT

In this paper, we study an emerging class of neural networks, the Morphological Neural networks, from some modern perspectives. Our approach utilizes ideas from tropical geometry and mathematical morphology. First, we state the training of a binary morphological classifier as a Difference-of-Convex optimization problem and extend this method to multiclass tasks. We then focus on general morphological networks trained with gradient descent variants and show, quantitatively via pruning schemes as well as qualitatively, the sparsity of the resulted representations compared to FeedForward networks with ReLU activations as well as the effect the training optimizer has on such compression techniques. Finally, we show how morphological networks can be employed to guarantee monotonicity and present a softened version of a known architecture, based on Maslov Dequantization, which alleviates issues of gradient propagation associated with its "hard" counterparts and moderately improves performance.

***Index Terms*—** Tropical Geometry, Morphological Neural Networks, Monotonicity, Pruning, Maslov Dequantization

1. INTRODUCTION

During the last decade, Neural Networks have been the focal point of machine learning research, especially in the dawn of the Deep Learning era. Most architectures utilize the multiply-accumulate scheme of the linear perceptron that feeds into a nonlinearity. An alternative approach lies on the use of morphological neurons, first introduced by Davidson and Hummer [1]. This approach was extended by Ritter and Sussner, where a simple network consisting of a single hidden layer was proposed for binary classification tasks resulting in a decision boundary parallel to the axes [2]. This limitation was addressed in two major ways, either by extending the architecture to a second hidden layer, where numerous such hyperplanes can be learned allowing the solution of arbitrary (binary) classification tasks [3] or by adding the option of hyperplane rotation [4].

Recently, the field of tropical geometry has been associated with this class of morphological networks [5, 6, 7, 8, 9, 10]. Tropical geometry studies *piecewise linear (PWL)* surfaces whose arithmetic is governed by a tropical semiring, where ordinary addition is replaced by the maximum or minimum and ordinary multiplication is replaced by ordinary addition. We refer to these algebraic structures as $(\max, +)$ and $(\min, +)$ semirings, respectively. These two semirings are dual and linked via the isomorphism $\phi(x) = -x$.

Ritter, Sussner, and Diza-de-Leon introduced the term of morphological networks by replacing addition and multiplication with maximum and addition [11, 12]. This process is called *tropicalization* and yields a path towards tropical mathematics. This connection was exploited by Charisopoulos and Maragos who explored these networks via the tropical prism [5, 6]. A similar class, the $\min - \max$ classifiers, was studied by Yang and Maragos in the Probably Approximately Correct (PAC) context and was associated with Boolean functions as a lattice-based generalization [13]. Pessoa and Maragos proposed a hybrid neuron consisting of morphological and linear terms where the output is the combination of a classical and a morphological perceptron [14]. The aforementioned works addressed training problems stemming from the non-differentiability of the morphological operators and proposed training algorithms specific to the models.

Neural Networks have also been proposed to address monotonicity constraints. Archer and Wang proposed a neural network for binary classification tasks, where the training algorithm diminishes the weights of the samples which violate the monotonicity constraints resulting in a network with only positive weights [15]. Other researchers proposed imposing positive weights in a single hidden network that feeds into a sigmoid at the output stage. However, Velikova, Daniels, and Feelders showed that this approach requires K hidden layers to approximate a K-dimensional monotonic surface [16]. On the other hand Sill proposes a network architecture that guarantees monotonicity for the output [17], which is extended by Daniels and Velikova to incorporate partially monotone functions [18] and can be viewed as a special case of the $\min - \max$ networks [13].

In this paper, we propose various improvements and extensions of previous works in the context of morphological networks. In Section 3 we extend a training process for morphological networks based on Difference-of-Convex optimization to tackle general multiclass classification tasks. Section 4 explores the compression abilities of Dense Morphological Networks [19] and draws favorable conclusions in comparison with their linear counterparts. In Section 5 we study how Sill's network architecture can be leveraged to guarantee shape properties for the output function, such as monotonicity, and present a softened version of the network via Maslov Dequantization which addresses training issues and improves performance. More details can be found in [20].

2. BACKGROUND CONCEPTS

For $\mathbf{x} \in \mathbb{R}^n$, the tropical max-plus polynomial $p_\vee$ is defined as the maximum of many affine terms: $p_\vee(\mathbf{x}) = \max_i\{\mathbf{a}_i^\top \mathbf{x} + b_i\} =$

[*] This work was performed when N.Dimitriadis was at NTUA.

[‡] The work of P. Maragos was co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH-CREATE-INNOVATE (Project: "e-Prevention", code: T1EDK-02890).

"Proceedings of 46th Int'l Conf. on Acoustics, Speech, and Signal Processing (ICASSP-2021), Toronto, June 2021"

$\bigvee_i \mathbf{a}_i^\top \mathbf{x} + b_i$. The max-plus polynomial generates convex PWL surfaces. Its min-plus equivalent arises from replacing $\max$ $(\vee)$ with $\min$ $(\wedge)$ and produces concave PWL surfaces. The max-plus and min-plus polynomials are related to the dilation (δ) and erosion (ε) operators of mathematical morphology, respectively.

The tropical polynomials yield PWL surfaces, also referred as hinging hyperplanes by Wang and Sun [21], which are a collection of affine terms jointed by edges where multiple terms *dominate*, i.e. are maximizers or minimizers. These edges form the tropical hypersurface and correspond to the boundaries among local affine functions, generating "hard" surfaces. By approximating the $\min$ and $\max$ operators, a softened version with no hard edges is achieved via the Maslov Dequantization:

Definition 1 (Maslov Dequantization [22, 23, 24])**.** Let $x, y \in \mathbb{R}$ and $h > 0$. The transformation $x \vee_h y = h \log(e^{x/h} + e^{y/h})$ defines the Maslov Dequantization of the max operator, yielding its soft approximation. Similarly, the Maslov Dequantization of the $\min$ operator is $x \wedge_h y = -h \log(e^{-x/h} + e^{-y/h})$. As $h \to 0$, the soft versions approach the hard ones: $\lim_{h \to 0} x \vee_h y = x \vee y$ and $\lim_{h \to 0} x \wedge_h y = x \wedge y$. For small positive values of h, these approximations are part of the Log-Sum-Exp family, used in convex analysis and recently linked to tropical polynomials [25, 26]. We use the reciprocal of h, the hardness parameter $\beta = h^{-1}$.

3. TRAINING MORPHOLOGICAL NETWORKS VIA CONVEX-CONCAVE PROCEDURE

Let us consider the task of classifying the pattern $\mathbf{x}_i \in \mathbb{R}^n, i = 1, 2, \dots, N$ to two distinct classes, $\mathcal{N}$ for negative $(y_i = -1)$ and $\mathcal{P}$ for positive $(y_i = +1)$. Given an input $\mathbf{x} \in \mathbb{R}^n$ and weight vectors $\mathbf{w}, \mathbf{m} \in \mathbb{R}^{n+1}$, dilation $\delta_\mathbf{w}$ and erosion $\varepsilon_\mathbf{m}$ compute the following activations respectively:

$$\delta_\mathbf{w}(\mathbf{x}) = w_0 \vee \left(\bigvee w_i + x_i \right) \tag{1}$$

$$\varepsilon_\mathbf{m}(\mathbf{x}) = m_0 \wedge \left(\bigwedge m_i + x_i \right) \tag{2}$$

The Dilation-Erosion Perceptron (DEP) results from their convex combination and can be thought of as a feedforward neural network with a single hidden layer consisting of two neurons, a dilation and an erosion. A similar architecture is the Maxout Network [27] that consists of the maximum of affine expressions which correspond to general tropical polynomials. Charisopoulos and Maragos formulate the problem of training a DEP classifier [5] as:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} v_i \max\{0, \xi_i\} \\ \text{s.t.} \quad & \lambda \delta_\mathbf{w}(\mathbf{x}_i) + (1-\lambda)\varepsilon_\mathbf{m}(\mathbf{x}_i) \geq -\xi_i \quad \forall \mathbf{x}_i \in \mathcal{P}, \\ & \lambda \delta_\mathbf{w}(\mathbf{x}_i) + (1-\lambda)\varepsilon_\mathbf{m}(\mathbf{x}_i) \leq +\xi_i \quad \forall \mathbf{x}_i \in \mathcal{N} \end{aligned} \tag{3}$$

where ξ_i are slack variables which ensure that only misclassified patterns are taken into account for the objective function and the variables v_i correspond to a weighting scheme proposed in [5] that penalizes patterns with greater chances of being outliers. The above formulation corresponds to a Difference-of-Convex optimization problem, since the dilation term is convex, whereas the erosion term is concave. Various methods have been proposed to tackle such problems, with many focusing on the Fenchel Conjugate. We use a heuristic called Convex-Concave Procedure (CCP), proposed by Yuille and Rangarajan [28] and extended in [29, 30].

Valle proposes a greedy algorithm where the dilation and the erosion perceptrons are trained separately and combined later by minimizing the average hinge loss [31]. This method allows the inclusion of a regularization term $C\|\mathbf{u} - \mathbf{r}\|_1$ in the objective function, where $\mathbf{u} = \mathbf{w}$ or $\mathbf{u} = \mathbf{m}$ and $\mathbf{r}$ is a reference term.

The Dilation-Erosion Perceptron suffers from a major flaw as a lattice-based model, it presupposes a partial ordering both on the features and the classes. By simply inverting the classes $\mathcal{N} \rightleftarrows \mathcal{P}$, the performance of the classifier might severely drop [31]. A way to counteract this behavior lies on the use of reduced morphological operators based on a reduced ordering:

Definition 2. Let R be a nonempty set, $\mathcal{L}$ be a complete lattice and $\rho : R \to \mathcal{L}$ be a surjective mapping. A reduced ordering, or r-ordering, is defined as: $\mathbf{x} \leq_\rho \mathbf{y} \Leftrightarrow \rho(\mathbf{x}) \leq \rho(\mathbf{y}), \forall \mathbf{x}, \mathbf{y} \in R$.

The DEP as well as its reduced variant, denoted as **r-DEP**, are binary classifiers. The formulation of the classification problem 3 is reminiscent of Support Vector Machines (SVMs). Various methods have been proposed regarding the extension of SVM-based classifiers to multiclass problems. Notably, there are two major approaches: *one-versus-the-rest* and *one-versus-one*. Let us consider a problem with $K > 2$ classes and N datapoints. In the former approach, the positive class consists of the elements of $\mathcal{C}_k$ and the negative class consists of the datapoints from all other classes $\mathcal{C}_{-k}$. A straightforward issue with this approach lies on the imbalance of the datasets, since for a relatively uniform distribution of datapoints among classes, we have $|\mathcal{C}_k| \simeq \frac{N}{K} \ll |\mathcal{C}_{-k}| \simeq \frac{(K-1)N}{K}$.

We employ the *one-versus-one* approach on the `MNIST` and `FashionMNIST` [32] datasets. A single reduced Dilation-Erosion Perceptron is used for every pair of classes and the final output on a single element is the majority (hard) vote of all classifiers. Thus, $\frac{K(K-1)}{2} = \frac{10 \times 9}{2} = 45$ distinct classifiers must be trained. A bagging classifier is evaluated for various numbers, denoted as n, of Radial Basis Function (RBF) kernel estimators. The results are presented in Table 1 and are comparable with those achieved using traditional methods of training (morphological) neural networks, presented in the next section. However, these methods concern models with many more parameters as the networks are both denser and deeper. Moreover, training via CCP is robust; repeating the experiments 10 times showed that the variation of the method is much lower than methods based on stochastic optimization such as gradient descent variants. We believe that with further experimentation, concerning both the selection of the kernel type as well as the number of kernel used, performance can be improved.

	`MNIST`	`FashionMNIST`
$n = 5$	$\mathbf{97.72 \pm 0.01}$	$\mathbf{88.21 \pm 0.01}$
$n = 10$	$\mathbf{97.72 \pm 0.01}$	88.07 ± 0.01
$n = 15$	97.67 ± 0.01	88.11 ± 0.01
$n = 20$	97.64 ± 0.01	88.12 ± 0.01

Table 1: Results of Bagging *multiclass* r-DEP with n RBF kernels.

4. PRUNING MORPHOLOGICAL NEURAL NETS

We extend the network of the previous section both in terms of density and depth. This family of networks is called Dense Morphological [19]. An example is presented in Fig. 1. By letting the network search for hidden representations of the input data, the major problem of constructing a surjective mapping ρ for the multiclass r-DEP is alleviated. Compared to the CCP formulation, this approach allows the parallelization of training and the use of optimized deep learning libraries that take advantage of GPUs, resulting in faster training. In this case, the models are trained with standard gradient descent methods. We study two variants: (mini-batch) Stochastic Gradient Descent (SGD) [33] and Adaptive Momentum Estimation (Adam) [34]. Our focus lies *not* on achieving the highest possible accuracy, but on showing the compression ability of morphological networks compared to traditional ones. To this end, we apply pruning techniques to evaluate the ability of the various networks to retain information with a fraction of the original nodes.

Regarding the hidden layer, the models studied include: only dilation neurons (denoted as δ), only erosion neurons (ε), a mixed network with both types of neurons (δ, ε) as well as a Feedforward neural networks with ReLU activations (FF-ReLU) for comparison.

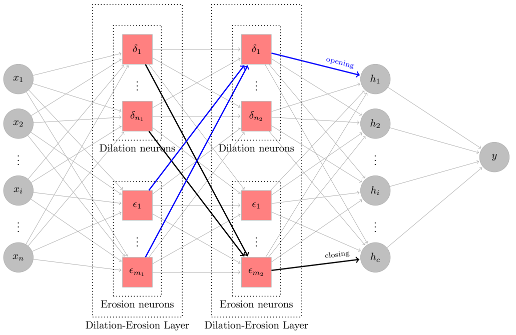
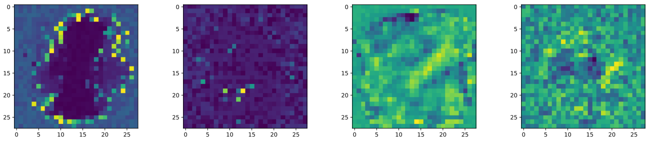


Fig. 1: Dense Morphological Network with 2 hidden layers. Square nodes correspond to morphological operators, either min or max. The output layer is fully connected.

All models are structurally identical, consisting of 400 neurons in the hidden layer. The experiments focus on visual recognition tasks and the datasets selected remain the same as in Section 3. The training lasts 50 epochs. After experimentation, we conclude that the best learning rates for Adam and SGD are $\eta = 0.001$ and $\eta = 0.09$, respectively. The results are presented in the first rows of each panel (one for each dataset) in Table 2.

We apply a simple pruning scheme based on the ℓ_1 norm [35] to keep salient features and discard unimportant ones. The method concerns the weights of the hidden layer where the bulk of parameters is. Furthermore, the output layer is a fully connected one and, thus, does not offer room for improvement. Various amounts of units are pruned and the performance of the remaining network is evaluated on the test set (see other rows of Table 2). The results show that the morphological network has superior compression capabilities than its more traditional counterpart. To be more specific, shades of red are used in Table 2 to showcase the degree of deterioration in accuracy, while green communicates the *absence* of performance loss between the unpruned net and the one using only 1% of the parameters (in the hidden layer). In both datasets and for both optimizers, the morphological network outperforms FF-ReLU, with the effect more evident on the more complex dataset, `FashionMNIST`. Another insight is that in both types of networks, the (mini-batch) SGD outperforms Adam in terms of sparsity even though it lacks slightly behind performance-wise on the full (unpruned) network.

These observations can be explained qualitatively by plotting the activations of the hidden layer (without bias terms). In Fig. 2, one of the 400 nodes is selected for 4 different networks studied and the flattened input is reshaped to a 28×28 grid for displaying purposes. The lighter color corresponds to a higher value. The activations of the morphological networks are in stark difference with those of their traditional counterparts, since high values characterize only a small percentage of the parameters. Intuitively, this suggests that by keeping only those parameters, performance should not be substantially affected. Given the similar performance of the full networks, Fig. 2 shows, in conjunction with the results of Table 2, that morphological networks utilize the same input data more efficiently. Finally, the Adaptive Momentum Estimation results in higher performance, but utilizes more connections than SGD. Thus, the pruning effect is more severe on models trained via Adam.



(a) (δ, ε) - Adam

(b) (δ, ε) - SGD

(c) FF-ReLU - Adam

(d) FF-ReLU - SGD

Fig. 2: Examples of hidden layer activations for various models (`MNIST` dataset). (a) & (b) correspond to a morphological network (dilation neurons), whereas (c) & (d) correspond to FF-ReLU.

5. MONOTONICITY CONSTRAINTS

We examine how morphological network architectures can be leveraged to guarantee monotonicity, where the output of the network does not decrease with with the increase of the input. Without loss of generality, we consider only monotonically increasing functions. Sill [17] proposes a min-pooling layer preceded by max-affine terms for learning monotonic functions. The network is presented in Fig. 3 and produces the following output for the pattern $\mathbf{x} \in \mathbb{R}^n$:

$$y = f(\mathbf{x}) = \bigwedge_{k \in [K]} \bigvee_{j \in [J]} \{\mathbf{w}_{k,j}^\top \mathbf{x} + b_{k,j}\} \tag{4}$$

A general monotonic surface is neither convex nor concave. As stated in Section 2, max-plus (min-plus) terms construct convex (concave) outputs and their combination yields a more general function, which can approximate any monotone function to an arbitrary degree of accuracy [17, Theorem 3.1]. Monotonicity constraints are enforced by limiting the weight vector to nonnegative values $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ via a function with a positive image, such as an exponential transformation $w_i = e^{z_i}$, $z_i \in \mathbb{R}$ [17] or $w_i = z_i^2$, which allows for flat surfaces [16].

The output is a PWL approximation of the input data. From a mathematical morphology viewpoint, the application of max term (dilation) followed by a min (erosion) yields an output similar to a closing. Each max term consists of J hyperplanes and constructs a group and Sill describes as *active* the group that determines the output for pattern $\mathbf{x} \in \mathbb{R}^n$. In the case of strictly positive weights, the transformation is reversible and corresponds to a morphological opening $x = f^{-1}(y) = \bigvee_{k \in [K]} \bigwedge_{j \in [J]} \{w_{k,j}^{-1}(y - b_{k,j})\}$ [36]. Sill uses a gradient descent training algorithm variant where the gradient for each hyperplane is computed by the error of the patterns corresponding to the active hyperplane at each iteration. We propose a softening of the morphological operators max and min via Maslov Dequantization (see Def. 1), which alleviates the undifferentiability of the the min and max operators. More specifically, the morphological operators are selective in the sense that a single element x_i of the input vector $\mathbf{x}$ is solely responsible for the output. Thus, in the backpropagation step, only x_i's parameters are updated. In the context of the Sill network, this implies that only the active hyperplane's weights are updated for a given pattern. Given low initialized weight parameters, we observed that the output of the network

		Adaptive Momentum Estimation				Stochastic Gradient Descent			
	p	δ	ε	(δ,ε)	FF-ReLU	δ	ε	(δ,ε)	FF-ReLU
MNIST	100%	97.62	96.17	97.95	98.13	94.86	93.36	96.07	98.16
	75%	97.62	96.18	97.93	98.15	94.86	93.36	96.07	98.12
	50%	97.62	96.22	97.90	98.17	94.86	93.37	96.07	98.08
	25%	97.62	96.09	97.87	97.51	94.86	93.40	96.06	98.01
	10%	97.62	95.78	97.74	93.38	94.86	93.38	96.09	96.67
	7.5%	97.62	95.42	97.76	90.17	94.86	93.38	96.10	95.56
	5%	97.62	94.51	97.66	83.39	94.86	93.40	96.10	92.96
	2.5%	97.62	93.43	97.37	68.93	94.86	93.39	96.09	80.48
	1%	97.62	91.17	97.08	44.22	94.86	93.38	96.08	58.07
FashionMNIST	100%	86.31	86.82	88.32	88.82	82.06	85.23	86.21	87.79
	75%	86.30	86.81	88.30	88.88	82.00	85.23	86.21	87.75
	50%	86.22	86.80	88.33	88.18	82.05	85.25	86.20	87.19
	25%	85.95	86.85	88.31	82.15	81.90	85.26	86.28	84.35
	10%	85.58	86.27	88.05	65.89	81.67	85.27	86.23	73.22
	7.5%	85.47	86.15	87.99	57.93	81.63	85.27	86.21	63.95
	5%	85.37	85.81	87.76	49.12	81.52	85.24	86.22	47.73
	2.5%	84.91	85.47	87.56	42.48	81.14	85.26	86.22	38.84
	1%	81.14	84.86	86.85	28.13	80.68	85.27	86.18	35.46

Table 2: Performance of pruned networks on the MNIST and FashionMNIST datasets for various model architectures.

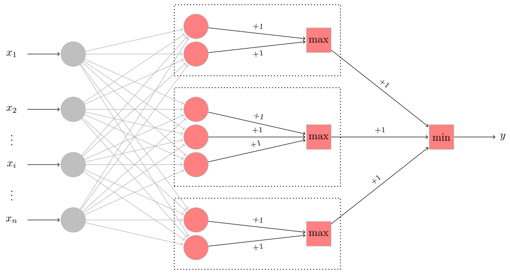


Fig. 3: Monotonic network. The gray edges correspond to nonnegative weights.

might result to a poor approximation of the data when using Adam [34] as our training algorithm (without Sill's hyperplane assignment of patterns), since the updated hyperplanes dominate the groups, not allowing the remaining (hyperplanes) to update their weights. One method to alleviate this problem is using a gain parameter G to magnify the initialized weights. This way, the weights of the dominating hyperplane get diminished during the backpropagation step, allowing the other hyperplanes to dominate in the next epochs. One way to circumvent this issue completely is the proposed use of softened morphological operators that do not have this problematic one-to-one correspondence between a single input element and the output.

We use a simple example to illustrate this method. We consider the strictly increasing function $f(x) = x^3 + x + \sin x, x \in [-4, 4]$. We scale both the domain and the image of f to $[-1, 1]$. We sample 100 observations uniformly and corrupt them with additive i.i.d zero-mean Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. We have used Glorot uniform initialization for all network parameters [37]. The training lasts 1000 epochs using Adam with $\eta = 0.01$. We use isotonic regression [38] for comparison. The results are presented in Table 3 for $K = J = 5$ and the outputs of the various methods for $\sigma = 0.15$ in Fig. 4. For the monotonic net, we select gain parameter $G = 20$ for the initialization of the weights. For the smooth monotonic net, we select hardness parameter $\beta = 5$.

From Table 3, we conclude that the smooth monotonic net outperforms the other methods for all noise levels σ. Moreover, its training procedure is less involved than Sill's assignment of patterns to hyperplanes during each step or the use of an arbitrary gain parameter G for initializing the weights.

σ	0.05	0.1	0.15	0.2
Linear Reg.	0.0236	0.03077	0.04827	0.0505
Isotonic Reg.	0.0042	0.01112	0.02557	0.0417
Sill Net	0.00305	0.01107	0.02401	0.0390
Smooth Sill Net	**0.00294**	**0.00938**	**0.02302**	**0.0386**

Table 3: RMS error of monotonic regression methods with noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

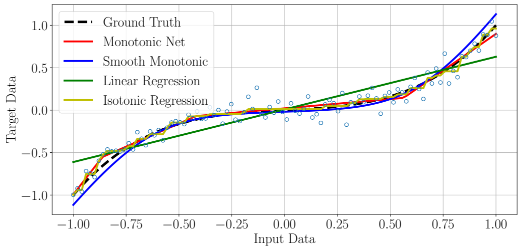


Fig. 4: Comparison of monotonic regression methods

6. CONCLUSION

Tropical geometry and mathematical morphology offer the mathematical foundations to analyze neural networks with min and max terms. In this work, we studied such networks with respect to their training, their compression ability and a method of enforcing monotonicity constraints. To this end, we extended a training algorithm based on nonconvex optimization to general (multiclass) classification tasks. Also, we explored the ability of (dense) morphological networks to efficiently construct hidden representations of the input data and retain information with minimal or zero accuracy loss despite heavy pruning and showed how they outperformed their linear equivalents. Finally, we used Maslov Dequantization in a known architecture for monotonic regression, in order to improve convergence and accuracy.

References

[1] Davidson, J. L. and Hummer, F., “Morphology neural networks: An introduction with applications”, *Circuits, Systems and Signal Processing*, vol. 12, pp. 177–210, 1993.

[2] Ritter, G. X. and Sussner, P., “An Introduction to Morphological Neural Networks”, in *Proc. 13th Int’l Conf. Pattern Recognition*, 1996.

[3] Sussner, P., “Morphological Perceptron Learning”, in *Proc. of 1998 IEEE Int’l Symposium on Intelligent Control*, 1998.

[4] Barmpoutis, A. and Ritter, G. X., “Orthonormal Basis Lattice Neural Networks”, in *Computational Intelligence Based on Lattice Theory*, Springer, 2007, pp. 45–58.

[5] Charisopoulos, V. and Maragos, P., “Morphological Perceptrons: Geometry and Training Algorithms”, in *Mathematical Morphology and Its Applications to Signal and Image Processing (Proc. ISMM 2017)*, ser. LNCS, vol. 10225, Springer, 2017.

[6] Charisopoulos, V. and Maragos, P., “A Tropical Approach to Neural Networks with Piecewise Linear Activations”, *arXiv*, 2018.

[7] Zhang, L., Naitzat, G., and Lim, L.-H., “Tropical Geometry of Deep Neural Networks”, in *Proc. ICML*, 2018.

[8] Smyrnis, G., Maragos, P., and Retsinas, G., “Maxpolynomial Division with Application To Neural Network Simplification”, in *Proc. ICASSP*, 2020.

[9] Smyrnis, G. and Maragos, P., “Multiclass Neural Network Minimization via Tropical Newton Polytope Approximation”, in *Proc. ICML*, 2020.

[10] Zhang, Y. *et al.*, “Max-Plus Operators Applied to Filter Selection and Model Pruning in Neural Networks”, in *Mathematical Morphology and Its Applications to Signal and Image Processing (Proc. ISMM 2019)*, ser. LNCS, vol. 11564, Springer, 2019.

[11] Ritter, G. X., Sussner, P., and Diza-de-Leon, J., “Morphological Associative Memories”, *IEEE Trans. Neural Networks*, vol. 9, pp. 281–293, 1998.

[12] Ritter, G. X. and Urcid, G., “Lattice Algebra Approach to Single-Neuron Computation”, *IEEE Trans. Neural Networks*, vol. 14, pp. 282–295, 2003.

[13] Yang, P.-F. and Maragos, P., “Min-max Classifiers: Learnability, Design and Application”, *Pattern Recognition*, vol. 28, pp. 879–899, 1995.

[14] Pessoa, L. F. and Maragos, P., “Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition”, *Pattern Recognition*, vol. 33, pp. 945–960, 2000.

[15] Archer, N. P. and Wang, S., “Application of the Back Propagation Neural Network Algorithm with Monotonicity Constraints for Two-Group Classification Problems”, *Decision Sciences*, vol. 24, pp. 60–75, 1993.

[16] Velikova, M., Daniels, H., and Feelders, A., “Solving Partially Monotone Problems with Neural Networks”, *WASET, Int’l J. of Computer, Electrical, Automation, Control and Inform. Eng.*, vol. 1, pp. 4043–4048, 2006.

[17] Sill, J., “Monotonic Networks”, in *Adv. in NeurIPS*, 1998.

[18] Daniels, H. and Velikova, M., “Monotone and Partially Monotone Neural Networks”, *IEEE Trans. on Neural Networks*, vol. 21, pp. 906–917, 2010.

[19] Mondal, R., Santra, S., and Chanda, B., “Dense Morphological Network: An Universal Function Approximator”, *arXiv*, 2019.

[20] Dimitriadis, N. and Maragos, P., “Advances in the training, pruning and enforcement of shape constraints of Morphological Neural Networks using Tropical Algebra”, *arXiv*, 2020.

[21] Wang, S. and Sun, X., “Generalization of Hinging Hyperplanes”, *IEEE Trans. on Information Theory*, vol. 51, pp. 4425–4431, 2005.

[22] Litvinov, G. L., “Maslov dequantization, idempotent and tropical mathematics: A brief introduction”, *Journal of Mathematical Sciences*, vol. 140, pp. 426–444, 2007.

[23] Maragos, P., “Dynamical systems on weighted lattices: general theory”, *Mathematics of Control, Signals, and Systems*, vol. 29, pp. 1–49, 2017.

[24] Maragos, P. and Theodosis, E., “Tropical Geometry and Piecewise-Linear Approximation of Curves and Surfaces on Weighted Lattices”, *arXiv*, 2019.

[25] Calafiore, G. C., Gaubert, S., and Possieri, C., “Log-Sum-Exp Neural Networks and Posynomial Models for Convex and Log-Log-Convex Data”, *IEEE Trans. Neural Networks and Learning Systems*, vol. 31, pp. 827–838, 2019.

[26] Calafiore, G. C., Gaubert, S., and Possieri, C., “A Universal Approximation Result for Difference of Log-Sum-Exp Neural Networks”, *IEEE Trans. Neural Networks and Learning Systems*, vol. 31, pp. 5603–5612, 2020.

[27] Goodfellow *et al.*, “Maxout Networks”, in *Proc. ICML*, 2013.

[28] Yuille, A. L. and Rangarajan, A., “The Concave-Convex Procedure”, *Neural computation*, vol. 15, pp. 915–936, 2003.

[29] Lipp, T. and Boyd, S., “Variations and extension of the convex–concave procedure”, *Optimization and Engineering*, vol. 17, pp. 263–287, 2016.

[30] Shen, X. *et al.*, “Disciplined Convex-Concave Programming”, in *Proc. IEEE CDC*, 2016.

[31] Valle, M. E., “Reduced Dilation-Erosion Perceptron for Binary Classification”, *Mathematics*, vol. 8, 2020.

[32] Xiao, H., Rasul, K., and Vollgraf, R., “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms”, *arXiv*, 2017.

[33] Ruder, S., “An overview of gradient descent optimization algorithms”, *arXiv*, 2017.

[34] Kingma, D. P. and Ba, J., “Adam: A Method for Stochastic Optimization”, *arXiv*, 2014.

[35] Li, H. *et al.*, “Pruning filters for efficient convnets”, *arXiv*, 2016.

[36] Duetting, P. *et al.*, “Optimal Auctions through Deep Learning”, in *Proc. ICML*, 2019.

[37] Glorot, X. and Bengio, Y., “Understanding the difficulty of training deep feedforward neural networks”, in *Proc. 13th Int’l Conf. Artificial Intelligence & Statistics*, 2010.

[38] Barlow, R. E. and Brunk, H. D., “The Isotonic Regression Problem and its Dual”, *J. Amer. Stat. Assoc.*, vol. 67, pp. 140–147, 1972.