Paper Number: 3819

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

Contributions

Binary Morphological Classifiers trained via Difference-of-Convex optimization	$\longrightarrow$	Extended to multiclass problems
Sparsity of Morphological Neural Nets	$\longrightarrow$	Showed quantitatively and qualitatively superior compression ability compared to ReLU FeedForward nets
Monotonic function approximation	$\longrightarrow$	Improved with softened morphological operators via Maslov Dequantization

Background concepts

Morphological Operators for Vectors

$$\text{Dilation:} \quad \delta_{\mathbf{w}}(\mathbf{x}) = w_0 \vee \left(\bigvee w_i + x_i\right)$$

$$\text{Erosion:} \quad \varepsilon_{\mathbf{m}}(\mathbf{x}) = m_0 \wedge \left(\bigwedge m_i + x_i\right)$$

Softmax and Softmin Scalar Operations via Maslov Dequantization [1]

($h > 0$: temperature parameter)

$$\max: \quad x \vee y \quad \longrightarrow \quad x \vee_h y = h \log(e^{xh} + e^{yh}) \quad : \text{softmax}$$

$$\min: \quad x \wedge y \quad \longrightarrow \quad x \wedge_h y = -h \log(e^{-xh} + e^{-yh}) \quad : \text{softmin}$$

Morphological Operators for Vectors ↘

Softened Morphological operators

Softmax and Softmin scalar operations ↗

Training Morphological Networks via Convex-Concave Procedure

Training for Binary Classification Problems

Dilation-Erosion Perceptron **DEP** combines dilation and erosion terms. Training can be formulated as a **Difference-of-Convex** program [2]:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N} v_i \max\{0, \xi_i\} \\ \text{subject to} \quad & \lambda \underbrace{\delta_{\mathbf{w}}(\mathbf{x}_i)}_{convex} + (1-\lambda) \underbrace{\varepsilon_{\mathbf{m}}(\mathbf{x}_i)}_{concave} \geq -\xi_i \quad \forall \mathbf{x}_i \in \mathcal{P}, \\ & \lambda \delta_{\mathbf{w}}(\mathbf{x}_i) + (1-\lambda) \varepsilon_{\mathbf{m}}(\mathbf{x}_i) \leq +\xi_i \quad \forall \mathbf{x}_i \in \mathcal{N} \end{aligned}$$

Extending to Multiclass Problems

1. Use or **reduced ordering** alleviates partial ordering flaw of lattice-based DEP → **r-DEP**
2. Extension to multiclass problems with **one-versus-one** approach:
   - $K > 2$ classes → $\frac{K(K-1)}{2}$ distinct classifiers
   - Used Bagging Classifier with RBF kernels
3. Training via CCP [3]: comparable results to similar nets trained with gradient descent
4. Training via CCP [3] is **robust**: variation is much lower compared to gradient descent variants

	MNIST	FashionMNIST
$n = 5$	97.72± 0.01	88.21±0.01
$n = 10$	97.72± 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.

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

Pruning Morphological Neural Nets

1. Studied **sparsity** of Dense Morphological Neural Networks [4]
2. Morphological Neural Networks have **superior compression capabilities** compared to FeedForward networks with ReLU activations (FF-ReLU)
3. Morphological Neural Networks can retain performance with *only* 1% of weights
4. Optimizer plays a role. SGD results in sparser representations than Adam

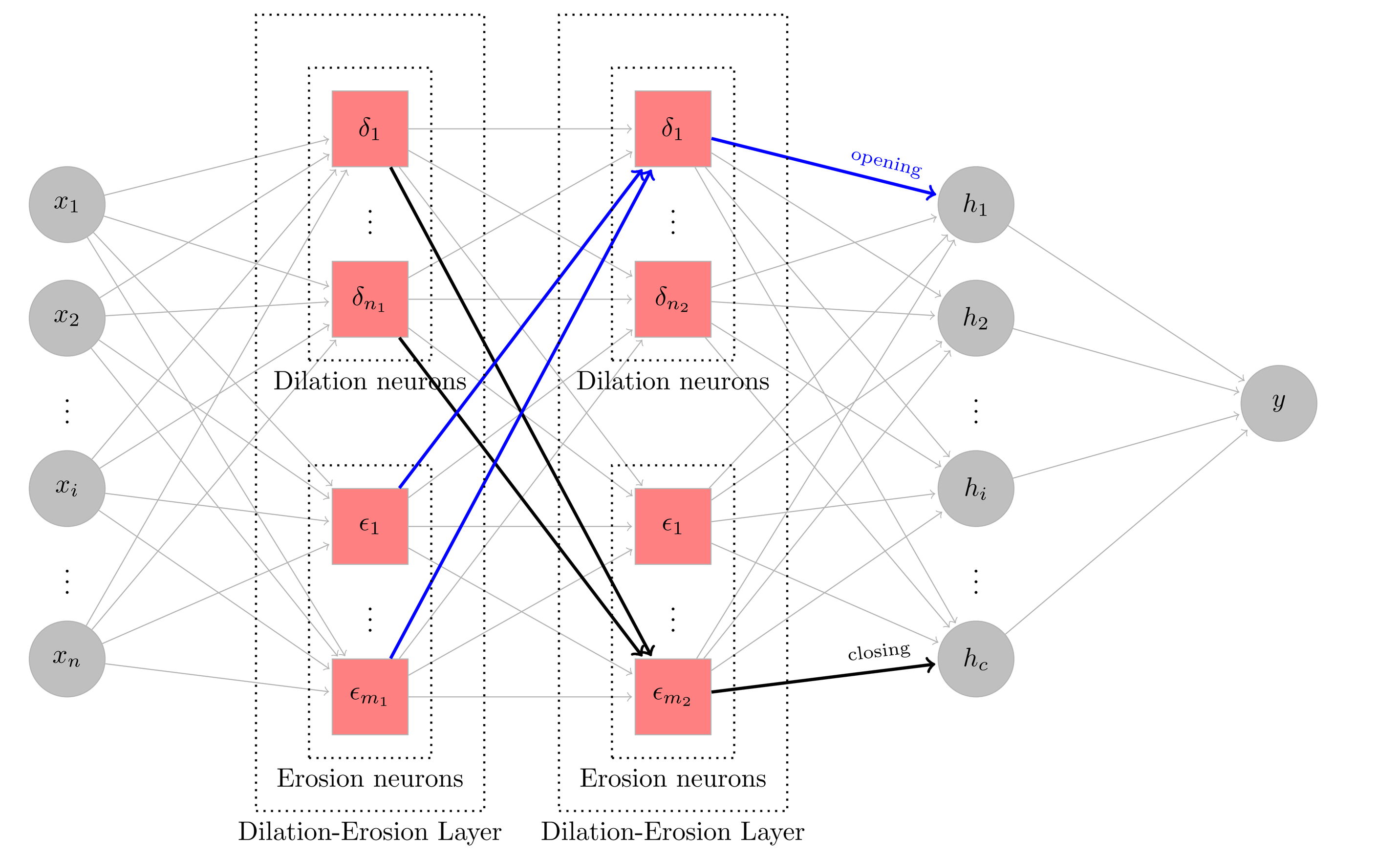
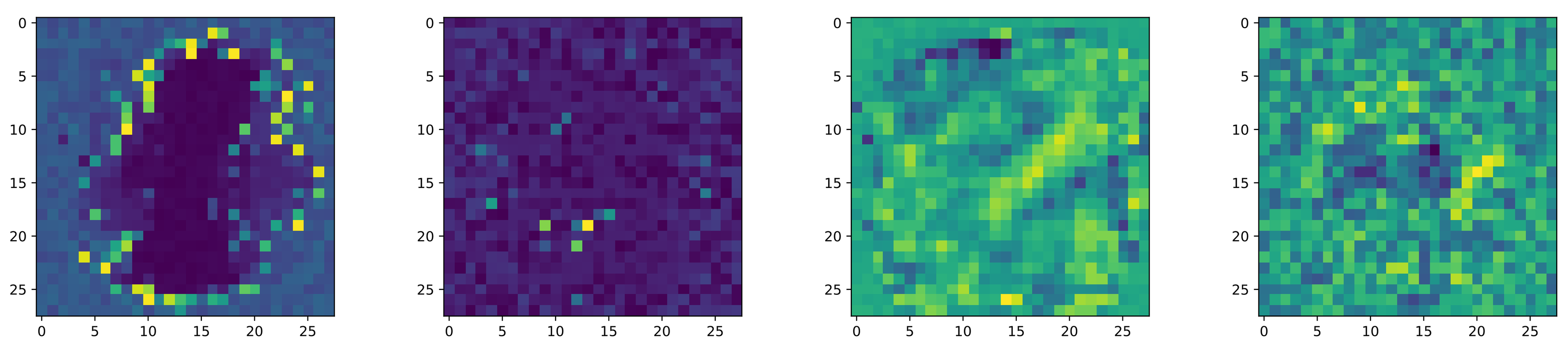


Figure 1. Dense Morphological Network with 2 hidden layers. Squares correspond to morphological neurons.

		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. Accuracy of pruned networks on the `MNIST` and `FashionMNIST` datasets. Models: δ → only dilation neurons, ε → only erosion, (δ, ε) → split equally, FF-ReLU → FeedForward NN with ReLU. green indicates the *absence* of performance loss between the unpruned net and the one using only 1% of the parameters, shades of red showcase the degree of (severe) deterioration in accuracy



(a) (δ, ε) - Adam (b) (δ, ε) - SGD (c) FF-ReLU - Adam (d) FF-ReLU - SGD

Figure 2. Hidden layer activations for various models (`MNIST` dataset).

Enforcing Monotonicity Constraints

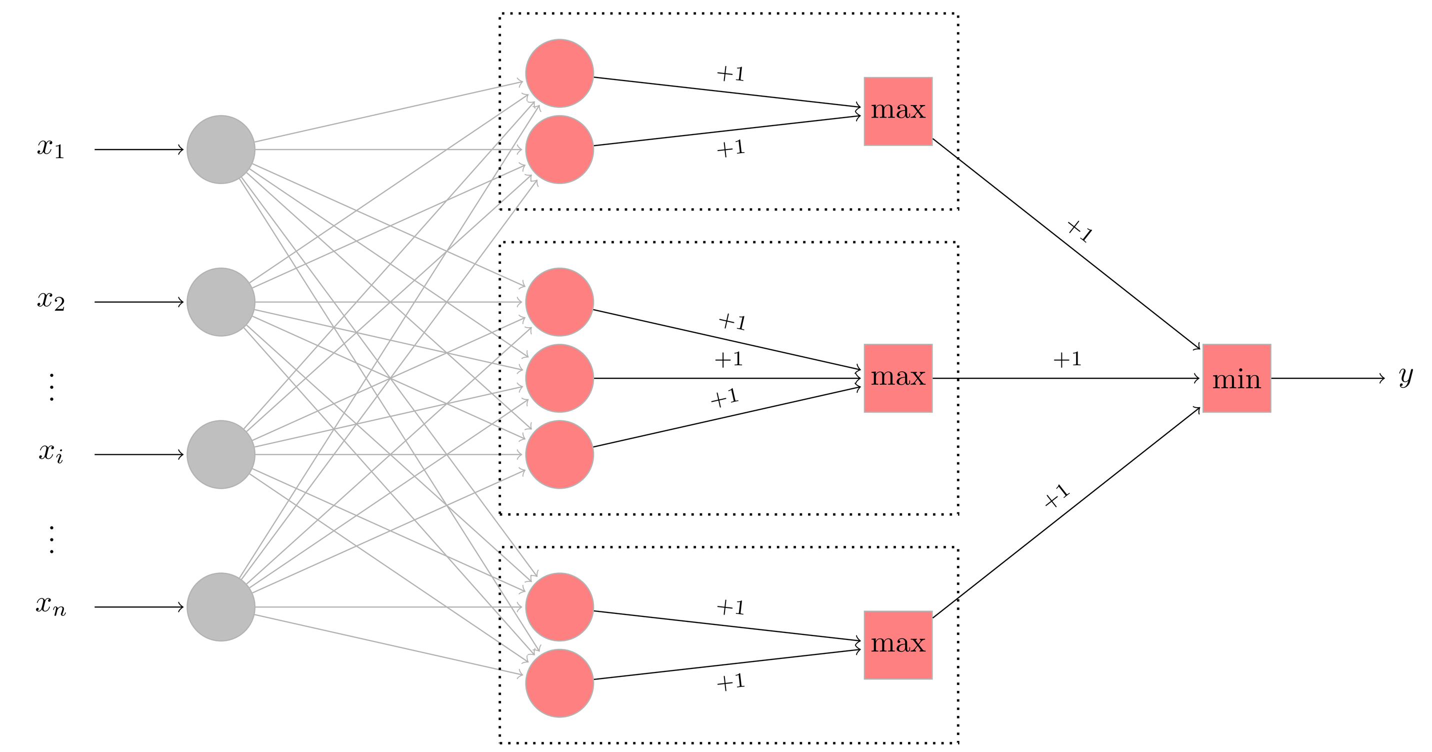


Figure 3. Monotonic network [5]. The gray edges correspond to nonnegative weights.

$$y = f(\mathbf{x}) = \bigwedge_{k \in [K]} \bigvee_{j \in [J]} \{\mathbf{w}_{k,j}^\top \mathbf{x} + b_{k,j}\}, \qquad \mathbf{w}_{k,j} \in \mathbb{R}_+^n \; \forall k \in [K], j \in [J]$$

- Used **softened morphological operators**
- **Active** group: affine term that determines the output for pattern $\mathbf{x} \in \mathbb{R}^n$
- "Hard" operators → $1-1$ correspondence between active group and output
  → only active hyperplane gets updated
  → a small fraction of hyperplanes dominate the training
- "Soft" operators alleviate undifferentiability → better approximation

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 [5]	0.00305	0.01107	0.02401	0.0390
Smooth Sill Net [ours]	0.00294	0.00938	0.02302	0.0386

Table 3. RMS error of monotonic regression methods for function $f(x) = x^3 + x + \sin x, x \in [-4, 4]$ scaled to $[-1, 1]$ and corrupted with additive i.i.d zero-mean Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

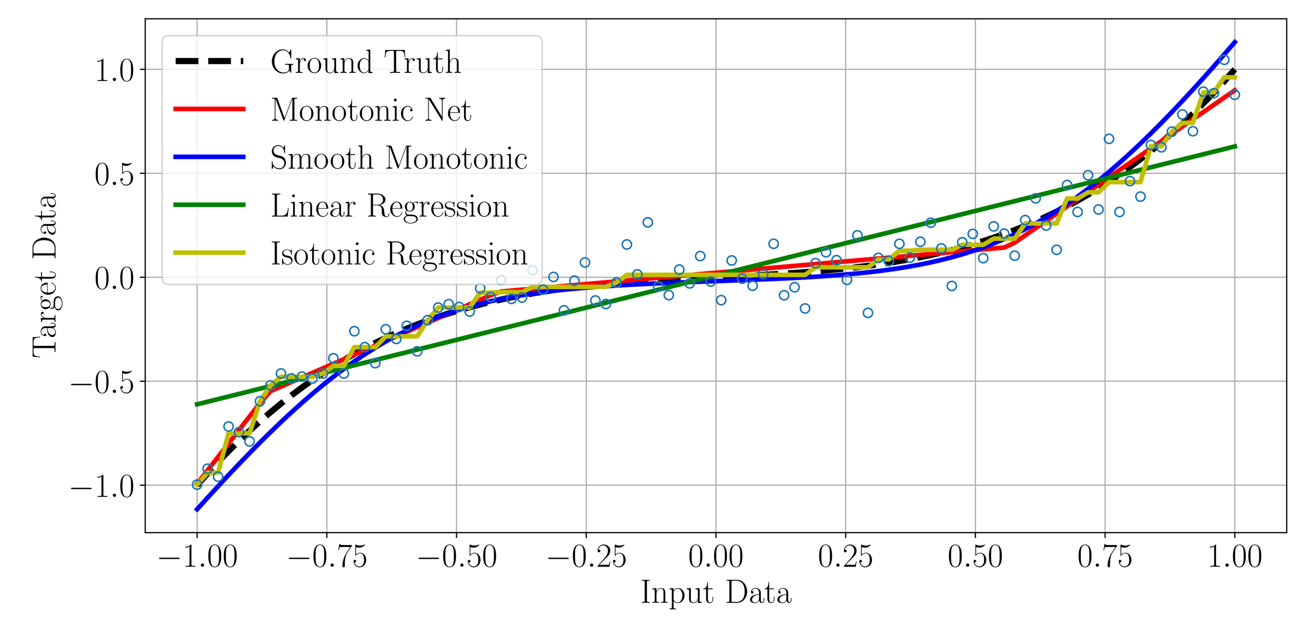


Figure 4. Comparison of monotonic regression methods
Smooth Monotonic is ours.

References

[1] G. L. Litvinov, "Maslov dequantization, idempotent and tropical mathematics: A brief introduction," *Journal of Mathematical Sciences*, vol. 140, no. 3, pp. 426–444, 2007.

[2] V. Charisopoulos and P. Maragos, "Morphological Perceptrons: Geometry and Training Algorithms," in *Mathematical Morphology and Its Applications to Signal and Image Processing (Proc. ISMM 2017)*, vol. 10225 of *LNCS*, pp. 3–15, Springer, 2017.

[3] A. L. Yuille and A. Rangarajan, "The Concave-Convex Procedure," *Neural computation*, vol. 15, no. 4, pp. 915–936, 2003.

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

[5] J. Sill, "Monotonic Networks," in *Adv. in NeurIPS*, 1998.