The Lattice Overparametrization Paradigm for the Machine Learning of Lattice Operators[⋆]

Diego Marcondes[1,2][0000−0002−6087−4821] and Junior Barrera[1][0000−0003−0439−0475]

[1] Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, São Paulo, Brazil.
[2] Department of Electrical and Computer Engineering, Texas A&M University, College Station, USA

Abstract. The machine learning of lattice operators has three possible bottlenecks. From a statistical standpoint, it is necessary to design a constrained class of operators based on prior information with low bias, and low complexity relative to the sample size. From a computational perspective, there should be an efficient algorithm to minimize an empirical error over the class. From an understanding point of view, the properties of the learned operator need to be derived, so its behavior can be theoretically understood. The statistical bottleneck can be overcome due to the rich literature about the representation of lattice operators, but there is no general learning algorithm for them. In this paper, we discuss a learning paradigm in which, by overparametrizing a class via elements in a lattice, an algorithm for minimizing functions in a lattice is applied to learn. We present the stochastic lattice descent algorithm as a general algorithm to learn on constrained classes of operators as long as a lattice overparametrization of it is fixed, and we discuss previous works which are proves of concept. Moreover, if there are algorithms to compute the basis of an operator from its overparametrization, then its properties can be deduced and the understanding bottleneck is also overcome. This learning paradigm has three properties that modern methods based on neural networks lack: control, transparency and interpretability. Nowadays, there is an increasing demand for methods with these characteristics, and we believe that mathematical morphology is in a unique position to supply them. The lattice overparametrization paradigm could be a missing piece for it to achieve its full potential within modern machine learning.

Keywords: lattice overparametrization · discrete morphological neural networks · image processing · mathematical morphology · U-curve algorithms · stochastic lattice descent

---

1   Algebraic representations of operators

Let $(\mathcal{L}, \leq)$ be a complete lattice. A lattice operator $\psi : \mathcal{L} \to \mathcal{L}$ is a mapping from $\mathcal{L}$ into itself, and we denote by $\Psi = \mathcal{L}^{\mathcal{L}}$ the set of all lattice operators in $\mathcal{L}$. The collection Ψ inherits the complete lattice structure of $\mathcal{L}$ by considering the pointwise partial order. Let $\Omega \subset \Psi$ be a complete sublattice of Ψ.

An algebraic representation of $(\Omega, \leq)$ is any complete lattice $(\Theta, \leq)$ such that there exists a lattice isomorphism $R : \Omega \to \Theta$. The element $\theta \in \Theta$ is the parameter that represents the operator $\psi_\theta = R^{-1}(\theta)$ and (R, Θ) is a parametrization of Ω. The algebraic representations are not unique and, although they are all equivalent, some have advantages over others.

A general algebraic representation of a lattice operator ψ is through its kernel, as proposed in[3] [5]. Let $\Theta_{\mathcal{K}} = \mathcal{P}(\mathcal{L})^{\mathcal{L}}$ be the collection of all maps $\mathcal{F}$ from $\mathcal{L}$ to $\mathcal{P}(\mathcal{L})$ equipped with the pointwise partial order

$$\mathcal{F}_1 \leq \mathcal{F}_2 \iff \mathcal{F}_1(Y) \subset \mathcal{F}_2(Y) \ \ \forall Y \in \mathcal{L}$$

for $\mathcal{F}_1, \mathcal{F}_2 \in \Theta_{\mathcal{K}}$, and consider the lattice isomorphism $R_{\mathcal{K}} : \Omega \to \Theta_{\mathcal{K}}$ given by

$$R_{\mathcal{K}}(\psi)(Y) = \mathcal{K}(\psi)(Y) = \{X \in \mathcal{L} : Y \leq \psi(X)\} \qquad (Y \in \mathcal{L}).$$

See [5, Proposition 6.1] for a proof that $R_{\mathcal{K}}$ is a lattice isomorphism.

The operators in specific lattices, such as finite lattices, and subclasses of operators in general lattices, such as upper semi-continuous operators [6], have a minimal algebraic representation by the maximal intervals lesser or equal to the kernel. Formally, for $\psi \in \Psi$ let

$$\boldsymbol{A}(\mathcal{K}(\psi)) = \{[\alpha, \beta] : [\alpha, \beta] \leq \mathcal{K}(\psi)\}$$

be the intervals[4] which are lesser or equal to the kernel of ψ. The basis of ψ is defined as the maximal intervals in $\boldsymbol{A}(\psi)$, that is

$$\begin{aligned}
\boldsymbol{B}(\psi) &= \operatorname{Max}\left(\boldsymbol{A}(\mathcal{K}(\psi))\right) \\
&= \{[\alpha, \beta] \in \boldsymbol{A}(\psi) : [\alpha', \beta'] \in \boldsymbol{A}(\psi), [\alpha, \beta] \leq [\alpha', \beta'] \implies [\alpha, \beta] = [\alpha', \beta']\}
\end{aligned}$$

in which $\leq$ above is the partial order in $(\Theta_{\mathcal{K}}, \leq)$. Under certain conditions, of which more details may be found in [4,5,6], it follows that

$$\psi = \vee \left\{\lambda_{[\alpha, \beta]} = \overline{\alpha} \wedge \overline{\beta} : [\alpha, \beta] \in \boldsymbol{B}(\psi)\right\} \qquad (1)$$

in which

$$\overline{\alpha}(X) = \vee \{Y \in \mathcal{L} : \alpha(Y) \leq X\} \quad \text{and} \quad \overline{\beta}(X) = \vee \{Y \in \mathcal{L} : X \leq \beta(Y)\}$$

for $X \in \mathcal{L}$. Decomposition (1) is called sup-generating decomposition of ψ and it has a dual inf-generating decomposition.

---

[3] We are calling kernel what [5] defined as left-kernel.
[4] See [5] for the formal definition of interval in this context.

Assuming that (1) holds for all $\psi \in \Omega$, denote by

$$\Theta_{\boldsymbol{B}} = \{\text{Max}\left(\boldsymbol{A}(\mathcal{F})\right) : \mathcal{F} \in \Theta_{\mathcal{K}}\}$$

the maximal intervals associated to each $\mathcal{F} \in \Theta_{\mathcal{K}}$ and consider the map $R_{\boldsymbol{B}} : \Theta_{\mathcal{K}} \to \Theta_{\boldsymbol{B}}$ given by

$$R_{\boldsymbol{B}}(\mathcal{F}) = \text{Max}\left(\boldsymbol{A}(\mathcal{F})\right).$$

It follows that $(\Theta_{\boldsymbol{B}}, \leq)$ is a complete lattice isomorphic to $(\Theta_{\mathcal{K}}, \leq)$ with partial order

$$\boldsymbol{B}_1 \leq \boldsymbol{B}_2 \iff \forall [\alpha, \beta] \in \boldsymbol{B}_1, \exists [\alpha', \beta'] \in \boldsymbol{B}_2 : [\alpha, \beta] \leq [\alpha', \beta']$$

in which the partial order on the right-hand side is that of $(\Theta_{\mathcal{K}}, \leq)$. From now on, we assume that Ω is a subclass of operators on $\mathcal{L}$ with a basis representation.

Specific classes of operators may have other algebraic representations. For instance, when $\mathcal{L} = \mathcal{P}(E)$ and $(E, +)$ is an Abelian group, then the class of translation invariant (t.i.) and locally defined lattice operators (i.e., W-operators), which in this case are set operators, can also be represented by a characteristic Boolean function. Denoting by Ψ_W the class of t.i. set operators locally defined within a window $W \in \mathcal{P}(E)$ and by $\mathfrak{B} = \{0, 1\}^{\mathcal{P}(W)}$ the Boolean functions in $\mathcal{P}(W)$, we consider the lattice isomorphism $R_{\mathfrak{B}} : \Psi_W \to \mathfrak{B}$ given by

$$R_{\mathfrak{B}}(\psi)(X) = \begin{cases} 1, & \text{if } o \in X \\ 0, & \text{otherwise} \end{cases} \qquad (X \in \mathcal{P}(W))$$

in which o is the zero element E. See [8] for more details.

Clearly, the isomorphisms may be composed to obtain isomorphisms between distinct algebraic representations and all algebraic representations are equivalent. For example, $R_{\mathfrak{B}, \boldsymbol{B}} : \mathfrak{B} \to \boldsymbol{B}$ given by $R_{\mathfrak{B}, \boldsymbol{B}} = R_{\boldsymbol{B}} \circ R_{\mathcal{K}} \circ R_{\mathfrak{B}}^{-1}$ is an isomorphism between $(\mathfrak{B}, \leq)$ and $(\boldsymbol{B}, \leq)$. The isomorphisms defined so far are illustrated in Figure 1.

From an algebraic perspective, the basis representation has some advantages over other representations, since algebraic properties of an operator may be deduced from its basis. For example, in the case of W-operators, the intervals in the basis of increasing operators are of form $[A, W]$ for $A \in \mathcal{P}(W)$; the basis of extensive increasing operators contains the interval $[o, W]$; and the basis of an increasing anti-extensive operator is such that $o \in A$ for all lower limits A of the intervals in its basis (see [21] for more details). Hence, reducing an operator to its basis representation is enough to verify its mathematical properties.

2   Lattice overparametrization

An algebraic representation R is an isomorphism between a class Ω and a parametric set Θ. Such a representation is obtained by departing from a fixed Ω and defining an isomorphism $R : \Omega \to \Theta$. This is done in [4,5,6]. Another family of representations may be obtained by departing from a Θ and defining an onto

map $\tilde{R} : \Theta \to \Omega$, so each parameter $\theta \in \Theta$ represents an operator $\psi_\theta = \tilde{R}(\theta) \in \Omega$ and for each $\psi \in \Omega$ there exists *at least one* $\theta \in \Theta$ such that $\psi = \psi_\theta$.

When $\tilde{R}$ is not injective, $(\tilde{R}, \Theta)$ is an overparametrization of Ω by the parameters in Θ since a same operator can be represented by more than one parameter. If $(\Theta, \leq)$ is a lattice, we say that $(\tilde{R}, \Theta)$ is a lattice overparametrization of Ω. Since $\tilde{R}$ is not an isomorphism, the partial relation in Θ is not equivalent to that in Ω. The basis of the operator represented by θ is given by $\tilde{R}_{\boldsymbol{B}}(\theta) = (R_{\boldsymbol{B}} \circ R_{\mathcal{K}} \circ \tilde{R})(\theta)$.

As an example, assume that $\mathcal{L} = \mathcal{P}(E)$ and $(E, +)$ is an Abelian group. For a finite subset $W \in \mathcal{P}(E)$ let

$$\Omega = \{\epsilon_A \vee \epsilon_B \vee \epsilon_C : A, B, C \in \mathcal{P}(W); \{[A, W], [B, W], [C, W]\} \text{ is maximal}\} \quad (2)$$

be the class of t.i. operators locally defined within W that can be written as the supremum of three erosions. We note that $\boldsymbol{B}(\epsilon_A \vee \epsilon_B \vee \epsilon_C) = \{[A, W], [B, W], [C, W]\}$ and Ω is actually the class of the increasing W-operators with at most three elements in their basis[5]. By making $\Theta = \mathcal{P}(W)^3$ and $\tilde{R}((A, B, C)) = \epsilon_A \vee \epsilon_B \vee \epsilon_C$ we have a lattice overparametrization of Ω since $\tilde{R}((A, B, C)) = \psi$ for all $(A, B, C) \in \mathcal{P}(W)^3$ satisfying $\mathcal{K}(\psi) = [A, W] \cup [B, W] \cup [C, W]$. By lifting the restriction that the intervals $\{[A, W], [B, W], [C, W]\}$ are maximal, we depart from an algebraic representation of Ω to a lattice overparametrization by a Boolean lattice.

A lattice overparametrization may be useful for representing a constrained class of operators defined via the composition, supremum, and infimum of operators that can be parametrized by elements in a lattice. A special case is when the operators can be written as combinations of erosions and dilations with structural elements in a lattice. In [22] we proposed the discrete morphological neural networks (DMNN) to represent constrained classes of W-operators via the composition, supremum and infimum of W-operators, which are an example of overparametrizations of a class of operators. In special, the canonical DMNN are those in which the W-operators computed in the network can be written as the supremum, infimum, complement, or composition of erosions and dilations with a same structuring element, an example of which is the class in (2) (see Example 5.8 in [22]). The canonical DMNN are a specific example of lattice overparametrization.

The main advantage of considering a lattice overparametrization is the possibility of applying general, efficient algorithms to learn operators in a constrained class. This is the case since the lattice $(\Theta, \leq)$ is known and can be chosen with desired computational properties, so minimizing a function in it may be more efficient than doing so in $(\Omega, \leq)$, specially when Ω is not a lattice. We further discuss the advantages of considering a lattice overparametrization to learn lattice operators in Section 4.

---

[5] Observe that if some of the elements A, B, C are equal, then $|\boldsymbol{B}(\epsilon_A \vee \epsilon_B \vee \epsilon_C)| < 3$.

3   The machine learning of lattice operators

The general framework for learning lattice operators consists of a class Ω, a sample $\mathcal{D}_N = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$ of N pairs of input and output elements X and Y in $\mathcal{L}$, in which Y is obtained by a possibly random transformation of X, and a loss function $\ell : \mathcal{L}^2 \times \Psi \to \mathbb{R}^2$ which evaluates the *error* $\ell((X, Y), \psi)$ incurred when $\psi(X)$ is applied to approximate Y, for each pair $(X, Y) \in \mathcal{L}^2$ and operator $\psi \in \Psi$.

It is assumed that the pairs in $\mathcal{D}_N$ are sampled from an unknown, but fixed, statistical distribution P over $\mathcal{L}^2$. Each $\psi \in \Psi$ has a mean expected error under distribution P defined as $L(\psi) = \mathbb{E}_P [\ell((X, Y), \psi)]$, in which the expectation is over a random vector (X, Y) with distribution P. A target operator of Ψ is a minimizer of L in Ψ and a target operator of Ω is a minimizer of L in Ω. We denote the target operators by $\psi^\star$ and $\psi_\Omega^\star$, respectively, and they satisfy $L(\psi^\star) \leq L(\psi), \forall \psi \in \Psi$, and $L(\psi_\Omega^\star) \leq L(\psi), \forall \psi \in \Omega$. For the sake of the argument, we assume that both target operators exist and are unique.

Defining

$$L_{\mathcal{D}_N}(\psi) = \frac{1}{N} \sum_{i=1}^{N} \ell((X_i, Y_i), \psi)$$

as the mean empirical error of $\psi \in \Psi$ in sample $\mathcal{D}_N$, the empirical risk minimization paradigm propose as an estimator for $\psi_\Omega^\star$ the operator that minimizes $L_{\mathcal{D}_N}$ in Ω:

$$\hat{\psi} = \arg\min_{\psi \in \Omega} L_{\mathcal{D}_N}(\psi) = \arg\min_{\theta \in \Theta} L_{\mathcal{D}_N}(\psi_\theta) \tag{3}$$

in which Θ is any representation, algebraic or otherwise, of Ω. The quality of the estimator $\hat{\psi}$ is measured by $L(\hat{\psi})$, which is called its generalization error, and assesses how it is expected to perform on data not in the sample, but generated by the same unknown distribution P.

The goal of learning is to obtain an estimator such that $L(\hat{\psi}) \approx L(\psi^\star)$ so its generalization quality is close to the best possible. On the one hand, it is necessary to have $L(\psi_\Omega^\star) \approx L(\psi^\star)$ for otherwise there is a systematic bias in the learning process since $\hat{\psi}$ cannot generalize better than $\psi_\Omega^\star$. On the other hand, if Ω is chosen as a class of complex operators, or as $\Omega = \Psi$, then, even if $\psi_\Omega^\star$ is as good as or equal to $\psi^\star$, if the sample size is not great enough, there may be a complex operator $\hat{\psi}$ in Ω that completely fits the data, so it has zero empirical error, but that does not generalize very well. When this happens, we say overfitting occurred. Actually, if $\Omega = \Psi$ and Ψ has infinite VC dimension, which is a measure of the complexity of a class of operators [29], not even an infinite sample suffices to guarantee that $L(\hat{\psi}) \approx L(\psi^\star)$. This is the usual bias-variance trade-off in machine learning [1].

Hence, we have the following statistical bottleneck for learning lattice operators:

(**B1**) *To fix a class of operators with low bias and relative low complexity*

The recipe to circumvent **(B1)** is the core of mathematical morphology: to design a class of operators based on prior information about the practical problem and on the mathematical properties of lattice operators. Geometrical and topological properties of the transformation applied to X_i to obtain Y_i in sample $\mathcal{D}_N$ are identified, and based on them a class Ω of lattice operators is designed via the mathematical morphology toolbox. If prior information is right, so the best operator in Ω well generalizes, and Ω is not too complex, then learning is feasible and $\hat{\psi}$ is expected to well generalize. As an example, the class in (2) can be applied to a problem in which it is known that an increasing transformation was applied to X_i to obtain Y_i, and the maximum number of elements in the basis controls the complexity of Ω.

There are almost 60 years of rich literature in mathematical morphology, that we could not possibly cite here without committing huge injustices, which can be directly applied to solving **(B1)**, so it is not really a bottleneck for learning lattice operators. However, there is a second, computational, bottleneck that has not yet been overcome in general:

(B2) *To compute $\hat{\psi}$ by solving* (3)

Despite their practical success, many proposed methods for the machine learning of lattice operators in the literature are heuristics that seek to control the complexity of the class of operators relative to the sample size, but do not strongly restrict the operator class based on prior information. The ISI algorithm [17], iterative designs [18] and multiresolution designs [13,19] offer methods to control the complexity of the class based on data, however are not flexible to represent specific classes of operators, but only general classes such as filters.

Furthermore, methods such as the those based on envelope constraints [9,10] can insert sharp prior information into the learning process by projecting the operator learned by a heuristic method into a constrained class, but do not guarantee that the projected operator well approximates the target of the class. Finally, we note that methods to solve (3) for specific classes, such as stack filters [16], have been proposed, but are not general methods that can be easily extended to other classes of operators. See [7] for more details on methods for the machine learning of operators.

We propose as a general paradigm to overcoming **(B2)** the development of algorithms to efficiently minimize, or approximately minimize, a function in a lattice so (3) can be at least approximately computed whenever Ω has a lattice overparametrization $(\Theta, \leq)$. Such an algorithm would be a general optimizer for learning operators once a subclass Ω and a lattice overparametrization for it is fixed. This abstract idea, which is behind the DMNN proposed in [22], can be a paradigm for the machine learning of lattice operators based on the stochastic lattice descent algorithm (SLDA). The general framework for the machine learning of lattice operators is depicted in Figure 1.

4   Stochastic lattice descent as a general learning algorithm

The U-curve algorithm was first proposed by [27] for minimizing U-shaped functions in Boolean lattices, and was then improved by [3,14,26]. It has also been applied to solve other problems in mathematical morphology [25]. Inspired by this algorithm and by the success of stochastic gradient descent algorithms for minimizing overparametrized functions in $\mathbb{R}^d$, such as the regularized empirical error of a neural network, we propose the SLDA to learn operators in a class with lattice overparametrization $(\Theta, \leq)$.

Informally, the SLDA performs a greedy search of a lattice to minimize an empirical error. At each step, n neighbors of an element are sampled and the empirical error on a fixed sample batch of the operator represented by each sampled neighbor is calculated. The algorithm jumps to the sampled neighbor with the least empirical error on the sample batch. The algorithm starts again from this new element, by sampling n neighbors and calculating their empirical error on a new sample batch. This process goes on for a predetermined number of epochs. An epoch ends when all sample batches have been considered, and the algorithm returns the element visited at the end of an epoch with the least empirical error on the whole sample. We now formally define the SLDA.

For each $\theta \in \Theta$, let $N(\theta)$ be a *neighborhood* of θ in $(\Theta, \leq)$. If Θ is countable, then $N(\theta)$ may be composed by the elements of Θ at distance one from θ. When Θ is uncountable and $d(\theta, \theta')$ is a distance measure, with $d(\theta, \theta') = \infty$ whenever $\theta \not\leq \theta'$ and $\theta' \not\leq \theta$, then one could consider $N(\theta) = \{\theta' : d(\theta, \theta') < \delta\}$ for a fixed $\delta > 0$. Assume that, given θ and a constant n, there exists an algorithm which samples n elements from $N(\theta)$. If $N(\theta)$ is a finite set, then the elements may be sampled uniformly, while if it is countable or uncountable then other statistical distributions should be considered.

The SLDA is formalized in Algorithm 1. The initial point $\theta \in \Theta$, a batch size[6] b, the number n of neighbors to be sampled at each step, and the number of training epochs is fixed. The initial point is stored as the point with minimum empirical error visited so far. For each epoch, the sample $\mathcal{D}_N$ is randomly partitioned in N/b batches $\{\tilde{\mathcal{D}}_b^{(1)}, \ldots, \tilde{\mathcal{D}}_b^{(N/b)}\}$. For each batch $\tilde{\mathcal{D}}_b^{(j)}$, n neighbors of θ are sampled and θ is updated to a sampled neighbor with the least empirical error $L_{\tilde{\mathcal{D}}_b^{(j)}}$, that is calculated on the sample batch $\tilde{\mathcal{D}}_b^{(j)}$. Observe that θ is updated at each batch, so during an epoch, it is updated N/b times.

At the end of each epoch, the empirical error $L_{\mathcal{D}_N}(\theta)$ of θ on the whole sample $\mathcal{D}_N$ is compared with the error of the point with the least empirical error visited so far at the end of an epoch, and it is stored as this point if its empirical error is lesser. After the predetermined number of epochs, the algorithm returns the point with the least empirical error on the whole sample $\mathcal{D}_N$ visited at the end of an epoch. For finite lattices, if $b = N$ and n is equal to the number of neighbors of θ, i.e., $n = n(\theta) = |N(\theta)|$, then Algorithm 1 reduces to the (deterministic) lattice descent algorithm.

---

[6] We assume that N/b is an integer to easy notation. If this is not the case, the last batch will contain less than b points.

Algorithm 1 Stochastic lattice descent algorithm for learning lattice operators.

Ensure: $\theta \in \Theta, n, b, Epochs$
1: $L_{min} \leftarrow L_{\mathcal{D}_N}(\psi_\theta)$
2: $\widehat{\theta} \leftarrow \theta$
3: **for** run $\in \{1, \dots, \text{Epochs}\}$ **do**
4:     $\{\tilde{\mathcal{D}}_b^{(1)}, \dots, \tilde{\mathcal{D}}_b^{(N/b)}\} \leftarrow \text{SampleBatch}(\mathcal{D}_N, b)$
5:     **for** $j \in \{1, \dots, N/b\}$ **do**
6:         $\tilde{N}(\theta) \leftarrow \text{SampleNeighbors}(\theta, n)$
7:         $\theta \leftarrow \theta'$ s.t. $\theta' \in \tilde{N}(\theta)$ and $L_{\tilde{\mathcal{D}}_b^{(j)}}(\psi_{\theta'}) = \min\{L_{\tilde{\mathcal{D}}_b^{(j)}}(\psi_{\theta''}) : \theta'' \in \tilde{N}(\theta)\}$
8:     **if** $L_{\mathcal{D}_N}(\psi_\theta) < L_{min}$ **then**
9:         $L_{min} \leftarrow L_{\mathcal{D}_N}(\psi_\theta)$
10:        $\widehat{\theta} \leftarrow \theta$
11: **return** $\widehat{\theta}$

An implementation of Algorithm 1 for a finite lattice has been done in [22] and good results were obtained in a simple binary image transformation problem. We note that in order for the algorithm to work for uncountable lattices, the statistical distribution applied to sample the neighbors should be chosen in a way to give a meaningful probability to chains in which the error decreases. The challenge of doing so is defining such a distribution without computing the error on the chains, what is computationally unfeasible. An implementation of the SLDA, or a modification of it, for uncountable lattices is currently an open problem.

We argue that, in general, it is not computationally feasible to apply the SLDA directly on lattice $(\Omega, \leq)$. On the one hand, since $(\Theta, \leq)$ is known a priori, for any $\theta \in \Theta$ the set $N(\theta)$ is known, so the complexity of sampling n neighbors should be that of sampling from a known statistical distribution, which is usually very low. On the other hand, if the SLDA was applied directly on $(\Omega, \leq)$, fixed a $\psi \in \Omega$, the computation of its neighborhood in $(\Omega, \leq)$ would be problem-specific and could have a great complexity. Therefore, suboptimally minimizing the empirical error in Θ via the SLDA should be less computationally complex than doing so in Ω. Furthermore, it is possible to learn on a poset $(\Omega, \leq)$ as long as it has a lattice overparametrization. In this case, minimizing the empirical error in $(\Omega, \leq)$ is a constrained optimization problem, while minimizing it in the lattice $(\Theta, \leq)$ is an unconstrained one which ought to be more efficiently solved.

We also note that the SLDA could be applied to the case in which Θ is a poset possibly contained in a lattice. In this case, the complexity of the algorithm could increase significantly due to the restrictions on $N(\theta)$. For example, sampling n neighbors of an element in a Boolean lattice is trivial, while sampling n neighbors which are also in a set of elements (the poset Θ) may be quite complex, specially when $\Theta \cap N(\theta)$ needs to be computed. In other cases, Θ being a poset may not meaningfully increase the complexity (see the application in [22]).

5   Degrees of prior information and hierarchical SLDA

When one has strong prior information about the properties that $\psi^\star$ satisfies, then he can properly fix a constrained Ω and, having a lattice overparametrization of Ω, he can in principle approximately compute (3). However, when strong prior information is not available, Ω may be too complex, so overfitting occurs, or the lattice Θ may be too complex, so high computational resources are needed. Either way, if one can decompose Θ into a lattice $(\mathbb{L}(\Theta), \subset)$ of subsets of Θ then he can apply an algorithm analogous to the SLDA to minimize a validation error in $(\mathbb{L}(\Theta), \subset)$ to select a subset $\hat{\Theta} \subset \Theta$, which represents a constrained class $\{\psi_\theta : \theta \in \hat{\Theta}\} \subset \Omega$, and then learn an operator in it. This is a specific instance of learning via model selection and is also represented in Figure 1 (see [24] for a formal definition of learning via model selection).

We proposed in [23] a hierarchical SLDA in the context of the unrestricted sequential DMNN proposed in [22] to represent W-operators. The class represented by these DMNN is composed of all operators that can be represented via the composition of d W-operators locally defined in $W_1, \ldots, W_d$, which is overparametrized by the Boolean characteristic functions of the W-operators. The set of possible sequences of Boolean functions is a Boolean lattice, and hence this is a lattice overparametrization. Since this class is quite complex, it is prone to overfit the data, so we propose a SLDA to select the windows of the W-operators, what is equivalent to creating equivalence classes on the characteristic functions' domain. Each possible sequence of windows defines a subset of Θ and varying all possible windows generates a lattice $(\mathbb{L}(\Theta), \subset)$ of subsets of Θ. This is an example where it is possible to learn lattice operators without strong prior information, and we refer to [23] for more details.

We are currently working on more general methods to learn lattice operators via a hierarchical SLDA in contexts where prior information is not available.

6   Control, transparency and interpretability

The lattice overparametrization paradigm for the machine learning of lattice operators has by design three important properties that modern learning methods lack in general: control, transparency and interpretability. Due to the extensive knowledge about lattice operators and the mathematical morphology toolbox, the practitioner has all the resources necessary to design Ω to fulfill its needs, so he has complete control over the class of operators. This is clear in the case of canonical DMNN in the context of set operators [22], which can represent any class of operators that can be decomposed via supremum, infimum, complement, and composition of erosions and dilations.

All the steps of the machine learning are transparent: the practitioner knows the properties of the operators in Ω since he can compute the basis of each one via $\tilde{R}_{\boldsymbol{B}}$; he knows the lattice overparametrization, which he chose; and he can trace the path of the SLDA and inspect the choices of the algorithm at each step. By monitoring the properties of θ each time it is updated, one can make sense of

Fig. 1. The lattice isomorphisms between representations of $(\Omega, \leq)$. The dashed lines represent an isomorphism that holds when the operators in Ω have a basis representation. The dotted lines represent isomorphisms that hold for t.i. and locally defined set operators when $\mathcal{L} = \mathcal{P}(E)$ and $(E, +)$ is an Abelian group. The orange arrows represent frameworks for the learning of lattice operators via the SLDA and via model selection based on data.

a possible logic that the algorithm is following. This monitoring may be that of the basis $\tilde{R}_{\boldsymbol{B}}(\theta)$ or of the values of the parameters θ in case they have semantic information.

Finally, the mathematical properties of the learned operator $\psi_{\hat{\theta}}$ are completely known, since it suffices to compute its basis $\tilde{R}_{\boldsymbol{B}}(\hat{\theta})$ from which its properties can be deduced. From these properties, it is possible to explain what the operator is doing, foresee cases in which it might not properly work and obtain insights about the relation between X and Y.

We note that these three properties are present in a learning framework only if $\tilde{R}_{\boldsymbol{B}}$ can be computed, for otherwise, if one cannot reduce an operator to its basis, then he may not be able to deduce its properties. This is a possible bottleneck to this learning paradigm:

(B3) *To compute $\tilde{R}_{\boldsymbol{B}}(\theta)$ for $\theta \in \Theta$*

For canonical DMNN in the context of set operators this is not a bottleneck since results in [8] allow computing the basis for each $\theta \in \theta$ (see [22, Remark 5.2] for more details). Moreover, results in [21] present general algorithms to compute the basis of many classes of set operators. Having these kinds of results for more general lattices is needed to overcome **(B3)**.

7 Conclusion

In the last decades, neural networks (NN) have been the main paradigm in image processing and its outstanding performance overshadowed mathematical morphology (MM), that has been relegated in favor of them (see the discussion in [2]). To this date, there has been no definitive method that brought MM to the deep learning era and many attempts try to insert MM into NN, as if NN were the *golden standard* and MM was only a second class *tool*. This last fact is completely false: in the context of lattice operators, NN do not have any advantage over MM from a theoretical perspective, and do not have the essence of MM, which control, transparency and interpretability are a part of. Indeed, needless to say, neural networks do not have, in general, any of these three properties: one does not have control over the class it represents, its learning algorithm and behavior is opaque, and its results are hardly interpretable.

Indeed, learning methods based on neural networks have been proposed in the last decades in the form of morphological neural networks [11,12,28]. More recently, they have been studied in the context of convolutional neural networks [15,20]. Although these methods do not suffer from **(B2)**, since they can be efficiently trained, they are opaque and as much a black-box as usual neural networks. In special, it is not trivial to insert constraint into them to achieve **(B1)** and once they are trained it is not possible to solve **(B3)** efficiently. Therefore, MNN do not address all the bottlenecks, but could maybe be adapted to fit the paradigm proposed in this paper. This is a current line of research.

However, NN do have great advantages from a practical standpoint, since they obtain good results and can be efficiently trained, and a great part of this success appears to be due to the possibility of proper learn in an overparametrized context. The paradigm proposed in this paper asks the following question: what if we could consider overparametrization to learn, but do not lose control and semantic understanding? To this day, there is no way of doing so with neural networks, but we argued in this paper that it is possible with MM as long as bottlenecks **(B1)**, **(B2)** and **(B3)** are overcome.

Since solving **(B1)** has been the purpose of MM for decades, it is necessary to overcome only **(B2)** and **(B3)**. The latter can be done by extending results such as those in [8,21] to general lattices, and the former can be overcome via implementations of algorithms analogous to the SLDA. We believe one should value MM and embrace aspects of NN, such as overparametrization, that can enhance MM without losing its spirit, instead of embracing NN and trying to insert MM into it.

The works in [22] and [23] are proofs of concept of the paradigm we discussed in this paper that, we believe, could be the guide for research in the machine learning of mathematical morphology in the deep learning era. The potential of such a line of research is enormous, since even if the performance of these methods come only close to that of neural networks, they may be preferred since they are controllable, transparent, and interpretable by design. Nowadays, there is an increasing demand for methods with these characteristics, and we believe that MM is in a unique position to supply them. The lattice overparametrization paradigm could be a missing piece for MM to achieve its full potential within modern machine learning.

References

1. Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.T.: Learning from data, vol. 4. AMLBook New York (2012)
2. Angulo, J.: Some open questions on morphological operators and representations in the deep learning era: A personal vision. In: Discrete Geometry and Mathematical Morphology: First International Joint Conference, DGMM 2021, Uppsala, Sweden, May 24–27, 2021, Proceedings. pp. 3–19. Springer (2021)
3. Atashpaz-Gargari, E., Reis, M.S., Braga-Neto, U.M., Barrera, J., Dougherty, E.R.: A fast branch-and-bound algorithm for u-curve feature selection. Pattern Recognition **73**, 172–188 (2018)
4. Banon, G.J.F., Barrera, J.: Minimal representations for translation-invariant set mappings by mathematical morphology. SIAM Journal on Applied Mathematics **51**(6), 1782–1798 (1991)
5. Banon, G.J.F., Barrera, J.: Decomposition of mappings between complete lattices by mathematical morphology, part i. general lattices. Signal Processing **30**(3), 299–327 (1993)
6. Barrera, J., Banon, G.J.F.: Expressiveness of the morphological language. In: Image Algebra and Morphological Image Processing III. vol. 1769, pp. 264–275. SPIE (1992)
7. Barrera, J., Hashimoto, R.F., Hirata, N.S., Hirata Jr, R., Reis, M.S.: From mathematical morphology to machine learning of image operators. São Paulo Journal of Mathematical Sciences **16**(1), 616–657 (2022)
8. Barrera, J., Salas, G.P.: Set operations on closed intervals and their applications to the automatic programming of morphological machines. Journal of Electronic Imaging **5**(3), 335–352 (1996)
9. Brun, M., Dougherty, E.R., Hirata Jr, R., Barrera, J.: Design of optimal binary filters under joint multiresolution–envelope constraint. Pattern recognition letters **24**(7), 937–945 (2003)
10. Brun, M., Hirata, R., Barrera, J., Dougherty, E.R.: Nonlinear filter design using envelopes. Journal of Mathematical Imaging and Vision **21**, 81–97 (2004)
11. Davidson, J.L., Ritter, G.X.: Theory of morphological neural networks. In: Digital Optical Computing II. vol. 1215, pp. 378–388. SPIE (1990)
12. Dimitriadis, N., Maragos, P.: Advances in morphological neural networks: training, pruning and enforcing shape constraints. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3825–3829. IEEE (2021)

13. Dougherty, E.R., Barrera, J., Mozelle, G., Kim, S., Brun, M.: Multiresolution analysis for optimal binary filters. Journal of Mathematical Imaging and Vision **14**, 53–72 (2001)

14. Estrela, G., Gubitoso, M.D., Ferreira, C.E., Barrera, J., Reis, M.S.: An efficient, parallelized algorithm for optimal conditional entropy-based feature selection. Entropy **22**(4), 492 (2020)

15. Franchi, G., Fehri, A., Yao, A.: Deep morphological networks. Pattern Recognition **102**, 107246 (2020)

16. Hirata, N.S., Barrera, J., Dougherty, E.R.: Design of statistically optimal stack filters. In: XII Brazilian Symposium on Computer Graphics and Image Processing (Cat. No. PR00481). pp. 265–274. IEEE (1999)

17. Hirata, N.S.T., Barrera, J., Terada, R., Dougherty, E.R., Talbot, H., Beare, R.: The incremental splitting of intervals algorithm for the design of binary image operators. Proceedings of the 6th ISMM pp. 219–228 (2002)

18. Hirata, N.S.T., Dougherty, E.R., Barrera, J.: Iterative design of morphological binary image operators. Optical Engineering **39**(12), 3106–3123 (2000)

19. Hirata Junior, R., Brun, M., Barrera, J., Dougherty, E.R.: Multiresolution design of aperture operators. Journal of Mathematical Imaging and Vision **16**, 199–222 (2002)

20. Hu, Y., Belkhir, N., Angulo, J., Yao, A., Franchi, G.: Learning deep morphological networks with neural architecture search. Pattern Recognition **131**, 108893 (2022)

21. Jones, R., Svalbe, I.D.: Basis algorithms in mathematical morphology. In: Advances in electronics and electron physics, vol. 89, pp. 325–390. Elsevier (1994)

22. Marcondes, D., Barrera, J.: Discrete morphological neural networks. arXiv preprint arXiv:2309.00588 (2023)

23. Marcondes, D., Feldman, M., Barrera, J.: An algorithm to train unconstrained sequential discrete morphological neural networks. arXiv preprint arXiv:2310.04584 (2023)

24. Marcondes, D., Peixoto, C.: Distribution-free deviation bounds of learning via model selection with cross-validation risk estimation. arXiv preprint arXiv:2303.08777 (2023)

25. Reis, M.S., Barrera, J.: Solving problems in mathematical morphology through reductions to the u-curve problem. In: International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing. pp. 49–60. Springer (2013)

26. Reis, M.S., Estrela, G., Ferreira, C.E., Barrera, J.: Optimal boolean lattice-based algorithms for the u-curve optimization problem. Information Sciences (2018)

27. Ris, M., Barrera, J., Martins, D.C.: U-curve: A branch-and-bound optimization algorithm for u-shaped cost functions on boolean lattices applied to the feature selection problem. Pattern Recognition **43**(3), 557–568 (2010)

28. Ritter, G.X., Sussner, P.: An introduction to morphological neural networks. In: Proceedings of 13th International Conference on Pattern Recognition. vol. 4, pp. 709–717. IEEE (1996)

29. Vapnik, V.: The nature of statistical learning theory. Springer science & business media (1999)