Training morphological neural networks with gradient descent: some theoretical insights

Samy Blusseau[1[0000−0003−0294−8172]]

Mines Paris, PSL University, Centre for mathematical morphology (CMM), Fontainebleau, France

Abstract. Morphological neural networks, or layers, can be a powerful tool to boost the progress in mathematical morphology, either on theoretical aspects such as the representation of complete lattice operators, or in the development of image processing pipelines. However, these architectures turn out to be difficult to train when they count more than a few morphological layers, at least within popular machine learning frameworks which use gradient descent based optimization algorithms. In this paper we investigate the potential and limitations of differentiation based approaches and back-propagation applied to morphological networks, in light of the non-smooth optimization concept of Bouligand derivative. We provide insights and first theoretical guidelines, in particular regarding initialization and learning rates.

Keywords: Morphological neural networks · Nonsmooth optimization · Lattice operators

1 Introduction

Morphological neural networks were introduced in the late 1980s [17,5], and have been revisited in recent years [4,18,12,6,8]. With the growing maturity of deep learning science, new exciting perspectives seem to open and give hope for significant breakthroughs.

In image processing, with the development of libraries specialized in morphological architectures [15], where basic as well as advanced operators are implemented, such as geodesical reconstruction layers [16], it is now within reach to train end to end pipelines which include morphological preprocessing and post-processing, and to use the know how of the morphological community to impose topological and geometrical constraints inside deep networks.

Furthermore, morphological networks can help investigate in practice the representation theory of lattice operators initiated by Georges Matheron [11,9,2]. Just as the universal approximation theorem for the multi-layer perceptron, the representation theorem of lattice operators with families of erosions and anti-dilations, is an existence one and is asymptotic, but does not provide any algorithm to actually exhibit such representations. Since these decompositions can be implemented as morphological layers, we may hope to learn these representations *from data*.

Yet, the optimization of morphological architectures is still slow and difficult. Despite the several contributions in this area, [12,6,1], architectures including morphological layers are often quite shallow and do not compete with the state of the art networks for image analysis. On the one hand, it may be due to the Fréchet non differentiability of the morphological layers, reason for several attempts to replace them by smooth approximations [7,8]. On the other hand, non smooth operations such as the Rectifier Linear Unit (ReLU) or the max-pooling, are commonly used in successfully trained architectures, while smooth morphological ones do not seem to solve all the optimization issues.

In this paper we investigate the potential and limitations of training morphological neural networks with differentiation-based algorithms relying on back-propagation and the chain rule. In Section 2 we introduce morphological networks, and recall in Section 3 the principles of gradient descent, back-propagation and chain rule. Section 4 presents the concept of Bouligand derivative, which is suited to morphological layers. In Section 5 we expose the possibilities and issues of this framework within the chain-rule paradigm, before concluding in Section 6.

2 Morphological networks

There is no universal definition of morphological neural networks, but most architectures that are called so, are neural networks including at least a morphological layer. In turn, a morphological layer is one computing a morphological operation such as a dilation or an erosion, or sometimes a (weighted) rank filter. In this paper we will focus on dilation and erosion layers, composed with each other or with other classical (dense or convolutional) layers.

Dilation layers. We will call dilation layer a function $\delta_W : \mathbb{R}^n \to \mathbb{R}^m$, $n, m \in \mathbb{N}^*$, defined by

$$\delta_W : \mathbf{x} = (x_1, \ldots, x_n) \mapsto \left(\max_{1 \leq k \leq n} x_k + w_{i,k} \right)_{1 \leq i \leq m} \tag{1}$$

where the $w_{i,k}$ are the real valued coefficients of a matrix $W \in \mathbb{R}^{m \times n}$, and the parameters (or *weights*) of the layer. Extended to the complete lattices $\bar{\mathbb{R}}^n$ and $\bar{\mathbb{R}}^m$, where $\bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$, δ_W is a shift invariant morphological dilation [10,3]. In practical neural architectures the input and output of a layer are usually represented as sets of vectors, called feature maps. In such a setting, each output feature map would be the supremum of dilations like δ_W, of the input feature maps. By reshaping the set of input feature maps into one input vector, and the set of output ones into one output vector, we get the equivalent formulation (1), simpler to analyze.

Erosion layers. Similarly, we will call erosion layer a function $\varepsilon_W : \mathbb{R}^m \to \mathbb{R}^n$, $n, m \in \mathbb{N}^*$, defined by

$$\varepsilon_W : \mathbf{x} = (x_1, \ldots, x_m) \mapsto \left(\min_{1 \leq k \leq m} x_k - w_{k,j} \right)_{1 \leq j \leq n}. \tag{2}$$

Note that the sign "$-$" and the transposition ($w_{k,i}$ instead of $w_{i,k}$) in the definition are meant to make $(\varepsilon_W,\ \delta_W)$ a morphological adjunction.

Morphological networks. As said earlier, in this paper any neural network including at least a morphological layer is considered a morphological network. This includes sequential compositions of dilations and erosions layers, supremum of erosion layers, infimum of dilation layers, and composition with classical layers (linear operators followed by a non-linear activation function). This also includes anti-dilations and anti-erosions, which are of the kind $\mathbf{x} \mapsto \delta_W(-\mathbf{x})$ and $\mathbf{x} \mapsto \varepsilon_W(-\mathbf{x})$. Note however that the composition $\delta_A \circ \delta_B$ of two dilation layers can be considered as one dilation layer δ_C where $C \in \mathbb{R}^{m\times n}$ is the max-plus matrix product of $A \in \mathbb{R}^{m\times p}$ by $B \in \mathbb{R}^{p\times n}$,

$$C_{ij} = \max_{1\le k\le p} A_{ik} + B_{kj}, \quad 1 \le i \le m,\ \ 1 \le j \le n. \tag{3}$$

Furthermore, the pointwise maximum of l dilation layers $\delta_{W_1}, \dots, \delta_{W_l}$ (where all W_is are the same size), is also equivalent to one dilation layer δ_{W^*} where W^* is the pointwise maximum the matrices W_is.

Similarly, in theory it is pointless to compose or take the minimum of erosion layers, since such operators can be represented (and learned) as one erosion layer.

3 Optimization with gradient descent

Let us consider a classic neural network setting where a function $f_\theta : \mathbb{R}^n \to \mathbb{R}^+$ depending on a parameter $\theta = [\theta_1, \dots, \theta_L]$ is a composition of L functions

$$f_\theta \coloneqq f_{L,\theta_L} \circ f_{L-1,\theta_{L-1}} \circ \dots \circ f_{1,\theta_1}, \tag{4}$$

each f_{k,θ_k} depending on its own parameter $\theta_k \in \mathbb{R}^{p_k}$ and mapping $\mathbb{R}^{n_k}$ to $\mathbb{R}^{n_{k+1}}$, with $n_1 = n$ and $n_L = 1$ (we include the loss function as part of the last layer). Typically, we would like to find a parameter θ which minimizes the expectation $\mathbb{E}(f_\theta(X))$ where X is a random variable that models the distribution of the data we want to process[1]. In practice this can be done by applying f_θ to samples $x_1, \dots, x_N$ of X and iteratively update $\theta \leftarrow \theta + \Delta\theta$ in a way that decreases the function at the current sample, $f_{\theta+\Delta\theta}(x_i) \le f_\theta(x_i)$. Hence the change $\Delta\theta$ that is looked for is a *descent direction.*

3.1 Gradient descent

Where it exists, the gradient of a function $g : \mathbb{R}^n \to \mathbb{R}$ precisely provides a descent direction. Indeed if g is Fréchet-differentiable[2] at $x \in \mathbb{R}^n$, then

$$\forall h \in \mathbb{R}^n,\ \forall \eta \ge 0, \quad g\big(x + \eta h\big) = g(x) + \eta\big(\langle \nabla g(x), h\rangle + \epsilon(\eta)\big) \tag{5}$$

[1] Recall that f_θ is real valued since we include the loss in the last layer f_{L,θ_L}.

[2] The Fréchet derivative is just the usual derivative, which is a linear function, like $h \mapsto \langle \nabla g(x), h\rangle$ in (5).

where $\langle \cdot, \cdot \rangle$ is the inner product in $\mathbb{R}^n$ and ϵ is a function that goes to zero when η goes to zero. Hence, if $\nabla g(x) \neq 0$, for η sufficiently small $|\epsilon(\eta)| < \|\nabla g(x)\|^2$ and therefore $g\big(x - \eta \nabla g(x)\big) < g(x)$, for which $-\nabla g(x)$ is called a descent direction of g at x. Equation (5) also implies that any $h \in \mathbb{R}^n$ such that $\langle \nabla g(x), h \rangle \leq 0$ is a descent direction. Furthermore, it shows $-\nabla g(x)$ is the *steepest* descent direction: given $\eta > 0$ sufficiently small, any unit vector v verifies $g\big(x - \eta \frac{\nabla g(x)}{\|\nabla g(x)\|}\big) \leq g\big(x + \eta v\big)$.

These results can be applied to the function $g : \theta \mapsto f_\theta(x)$ for a fixed sample x, provided g is a Fréchet-differentiable (also called F-differentiable) function of θ. In that case we will note $\nabla_\theta f_\theta(x) \coloneqq \nabla g$.

3.2 Back propagation and the chain rule

To compute $\nabla_\theta f_\theta(x)$, it is sufficient to compute each $\nabla_{\theta_i} f_\theta(x)$, which can also be noted $\frac{\partial f_\theta(x)}{\partial \theta_i}$, and is the gradient of the function $g_i : \theta_i \mapsto f_\theta(x)$, x and the other parameters $\theta_j, j \neq i$, being fixed. Indeed, the gradient with respect to θ is the concatenation of the gradients with respect to the θ_is, $\nabla_\theta f_\theta(x) = [\nabla_{\theta_1} f_\theta(x), \ldots, \nabla_{\theta_L} f_\theta(x)]$.

To obtain these, the so called "chain rule" is applied, involving the (Fréchet) derivative of each layer with respect to its input variable and its derivative with respect to its parameter. The derivatives with respect to the input variables tell earlier layers (i.e. the layers that are closer to the input) how they should change their output values to eventually decrease the whole function $f_\theta(x)$. They play a role of message passing to earlier layers. The derivative with respect to a layer's parameter tells how to update this parameter in order to comply with the instruction received from later layers (that is, layers closer to the output).

More formally, we can see this in the case of f_θ as defined in (4). We note $\mathbf{x}_1 \coloneqq x$ the input variable of f_{1,θ_1} (and therefore f_θ), and $\mathbf{x}_{k+1} \coloneqq f_{k,\theta_k}(\mathbf{x}_k)$, $1 \leq k \leq L-1$. For fixed $\theta_k, \mathbf{x}_k$, we denote by $f'_{k,\theta_k}(\mathbf{x}_k;\ \cdot\ )$ and $f'_{k,\mathbf{x}_k}(\theta_k;\ \cdot\ )$ the Fréchet derivatives of the k-th layer with respect to its input variable and parameter respectively. Then the chain rule algorithm can be summarized as follows (see Figure 1).

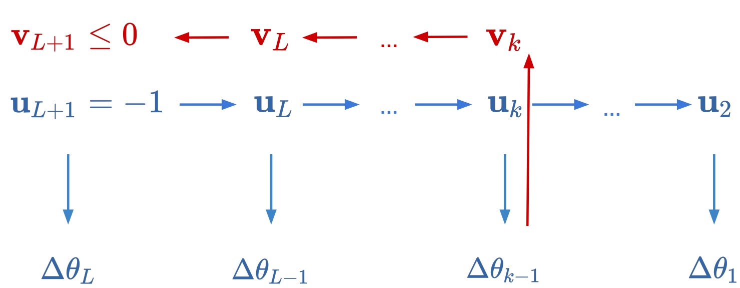

Fig. 1. illustration of the chain rule algorithm.

Initialize the message $\mathbf{u}_{L+1}$: Since we want to decrease f_θ, the first target direction to be passed on to layer L is $\mathbf{u}_{L+1} = -1$.

Update θ_k, given $\mathbf{u}_{k+1}$: Move θ_k in the direction

$$\Delta\theta_k \coloneqq \arg\max_{\|\mathbf{h}\|=1} \langle f'_{k,\mathbf{x}_k}(\theta_k; \mathbf{h}), \mathbf{u}_{k+1}\rangle. \tag{6}$$

Pass on message $\mathbf{u}_k$, given $\mathbf{u}_{k+1}$: If $k \geq 2$ pass to layer $k-1$ the target direction

$$\mathbf{u}_k \coloneqq \arg\max_{\|\mathbf{h}\|=1} \langle f'_{k,\theta_k}(\mathbf{x}_k; \mathbf{h}), \mathbf{u}_{k+1}\rangle. \tag{7}$$

Both problems (6) and (7) are easily solved using $f^*_{k,\mathbf{x}_k}(\theta_k;\ \cdot\ )$ and $f^*_{k,\theta_k}(\mathbf{x}_k;\ \cdot\ )$, the adjoint operators to the derivatives $f'_{k,\mathbf{x}_k}(\theta_k;\ \cdot\ )$ and $f'_{k,\theta_k}(\mathbf{x}_k;\ \cdot\ )$ respectively:

$$\Delta\theta_k = \frac{f^*_{k,\mathbf{x}_k}(\theta_k;\ \mathbf{u}_{k+1})}{\|f^*_{k,\mathbf{x}_k}(\theta_k;\ \mathbf{u}_{k+1})\|} \quad \text{and} \quad \mathbf{u}_k = \frac{f^*_{k,\theta_k}(\mathbf{x}_k;\ \mathbf{u}_{k+1})}{\|f^*_{k,\theta_k}(\mathbf{x}_k;\ \mathbf{u}_{k+1})\|}. \tag{8}$$

These solutions do not ensure a change of the output value of layer k in the direction $\mathbf{u}_{k+1}$, but they do guarantee

$$\langle f'_{k,\mathbf{x}_k}(\theta_k; \Delta\theta_k), \mathbf{u}_{k+1}\rangle \geq 0 \quad \text{and} \quad \langle f'_{k,\theta_k}(\mathbf{x}_k; \mathbf{u}_k), \mathbf{u}_{k+1}\rangle \geq 0. \tag{9}$$

Answer $\mathbf{v}_k$ to message $\mathbf{u}_k$: Then, when layer $k-1$ ($k \geq 2$) updates its parameter in the direction $\Delta\theta_{k-1}$, its output does not move in the target direction $\mathbf{u}_k$, but in the direction $\mathbf{v}_k \coloneqq f'_{k-1,\mathbf{x}_{k-1}}(\theta_{k-1}; \Delta\theta_{k-1})$, which "only" verifies $\langle \mathbf{v}_k, \mathbf{u}_k\rangle \geq 0$, according to (9). Therefore, the output of layer k moves in the direction $\mathbf{v}_{k+1} \coloneqq f'_{k,\theta_k}(\mathbf{x}_k; \mathbf{v}_k)$ instead of $f'_{k,\theta_k}(\mathbf{x}_k; \mathbf{u}_k)$, and so on. The linearity of $f'_{k,\theta_k}(\mathbf{x}_k;\ \cdot\ )$ ensures the crucial following property

$$\begin{cases} \langle f'_{k,\theta_k}(\mathbf{x}_k; \mathbf{u}_k), \mathbf{u}_{k+1}\rangle \geq 0 \\ \qquad \langle \mathbf{v}_k, \mathbf{u}_k\rangle \geq 0 \end{cases} \Rightarrow \langle \mathbf{v}_{k+1}, \mathbf{u}_{k+1}\rangle \coloneqq \langle f'_{k,\theta_k}(\mathbf{x}_k; \mathbf{v}_k), \mathbf{u}_{k+1}\rangle \geq 0. \tag{10}$$

Hence, as soon as (9) and (10) hold for layer $k-1$ and later layers, the property $\langle \mathbf{v}_k, \mathbf{u}_k\rangle \geq 0$, triggered by the update $\Delta\theta_{k-1}$, propagates and eventually yields $\langle \mathbf{v}_{L+1}, \mathbf{u}_{L+1}\rangle \geq 0$, i.e. $\mathbf{v}_{L+1} \leq 0$, meaning the output of f_θ is decreased.

This quick reminder of the chain rule mechanism highlights that the layer derivatives have two goals: optimal message passing and optimal parameter update based on the message passed by later layers. Therefore, in the case of non Fréchet-differentiable layers, like dilation and erosion layers, we may investigate if these two targets, namely properties (9) and (10), can still be met somehow. In the next sections we will see that morphological layers are differentiable in the more general sense of the Bouligand differentiability, which makes this notion worth analyzing in the perspective of optimization with gradient-descent-like algorithms.

4 The Bouligand derivative

The Bouligand derivative has been introduced in the nonsmooth analysis literature [13,14]. It is a directional derivative that provides a first order approximation of its function in all directions. Formally, given a function $g : \mathbb{R}^n \to \mathbb{R}^m$ and

$x \in \mathbb{R}^n$, if for every $y \in \mathbb{R}^n$ the limit

$$g'(x; y) \coloneqq \lim_{\alpha \to 0, \alpha > 0} \frac{g(x + \alpha y) - g(x)}{\alpha} \tag{11}$$

exists, then g is directionally differentiable at x and $g'(x; .\ )$ is called its directional derivative at x. If additionally for any $h \in \mathbb{R}^n$

$$g(x + h) = g(x) + g'(x; h) + o_0(h) \tag{12}$$

then g is said to be Bouligand differentiable (or B-differentiable) at x, and $g'(x; .\ )$ is its Bouligand derivative, also called B-derivative[3]. If g is B-differentiable at every $x \in \mathbb{R}^n$, then it simply said B-differentiable.

Fréchet differentiablilty implies B-differentiability, but what makes the latter more general than the former is that the B-derivative does not need to be a linear function. If $g'(x; .\ )$ is a linear function, then g is Fréchet differentiable at x, and $g'(x; .\ )$ is its Fréchet derivative at that point.

The B-derivative has nice properties similar to the Fréchet derivative, in particular [14]:

- **Positive homogeneity:** $g'(x; \lambda y) = \lambda g'(x; y)$ for any $\lambda \geq 0$.
- **Chain rule:** if $f : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^p \to \mathbb{R}^n$ are continuous and B-differentiable at $x \in \mathbb{R}^p$ and $g(x)$ respectively, then $f \circ g$ is B-differentiable at x and
$$(f \circ g)'(x; y) = f'\left(g(x); g'(x; y)\right) \tag{13}$$
- **Linearity of $f \mapsto f'(x; .)$:** if $f : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are continuous and B-differentiable at $x \in \mathbb{R}^n$, then so is $\alpha f + \beta g$ for any $\alpha, \beta \in \mathbb{R}$ and
$$(\alpha f + \beta g)'(x; y) = \alpha f'(x; y) + \beta g'(x; y). \tag{14}$$
- **Derivative of components:** $g : \mathbb{R}^n \to \mathbb{R}^m$ is B-differentiable at x if and only if each of its components $g_i : \mathbb{R}^n \to \mathbb{R}$ is, and in this case
$$g'(x; y) = \big(g_1'(x; y), \ldots, g_m'(x; y)\big). \tag{15}$$

As we will see, the dilation and erosion layers are continuous and B-differentiable functions of both their input variables and parameters, as well as all the usual neural layers. Therefore, a neural network $f_\theta(x)$ is a continuous and B-differentiable function of its parameter $\theta \in \mathbb{R}^p$ for a fixed x. Noting $g : \theta \mapsto f_\theta(x)$ we have for $h \in \mathbb{R}^p$ and any $\eta > 0$,

$$g(\theta + \eta h) = g(\theta) + \eta\big(g'(\theta; h) + \epsilon(\eta)\big) \tag{16}$$

where ϵ is a function that goes to zero when η goes to zero. Hence, we are left with finding in which direction h we need to move the parameter θ in order to ensure $g(\theta + \eta h) < g(\theta)$ for η sufficiently small. Whereas this was straightforward when $g'(\theta; h) = \langle \nabla g(\theta),\ h \rangle$ in Equation (5), the problem is open when $g'(\theta; \cdot)$ is not linear. The purpose of the next section is to focus on this problem in the case of morphological neural networks.

[3] Recall that $o_0(h)$ denotes $h \cdot \epsilon(h)$ where ϵ is any function that goes to zero when h goes to zero.

5 Optimization with the Bouligand derivative

5.1 Derivatives of the morphological layers

The Bouligand derivatives of the dilation and erosion layers with respect to their input values and parameters, are well known in the nonsmooth optimization literature [14], since they are easy examples of *piecewise affine functions* for which formulas exist. Here we provide some details of their computation, that will matter in addressing the problem stated in the previous section. We focus on the dilation layers, the case of erosions being analogous.

With the same notations as in Section 2, we denote by $\mathbf{x} \in \mathbb{R}^n$ and $W \in \mathbb{R}^{m\times n}$ the input vector and parameter matrix of a dilation layer. We will note $\delta_W(\mathbf{x})$ to clarify that we are considering a function of $\mathbf{x}$ with fixed parameter W, and $\delta_{\mathbf{x}}(W)$ for a function of W with fixed $\mathbf{x}$.

Derivative with respect to *W* An interesting property of $\delta_{\mathbf{x}}$ is that, if we move away from W in the direction $H \in \mathbb{R}^{m\times n}$, with a sufficiently small step $\eta \geq 0$, $\delta_{\mathbf{x}}(W+\eta H)$ shows an exact affine behaviour. Proposition 1 below provides a sufficient and necessary condition on the step η for this to hold. It will also provide the Bouligand derivative of $\delta_{\mathbf{x}}$.

Given a fixed $\mathbf{x} \in \mathbb{R}^n$ and a variable $W \in \mathbb{R}^{m\times n}$ we note $\delta_{\mathbf{x}}(W) = \big(\varphi_{\mathbf{x},i}(W)\big)_{1\leq i\leq m}$ with

$$\varphi_{\mathbf{x},i}(W) \coloneqq \max_{1\leq j\leq n} w_{ij} + x_j. \tag{17}$$

Additionally, for each index $1 \leq i \leq m$, let us note

$$J_{W_i,\mathbf{x}} \coloneqq \{j \in \{1,\dots,n\},\ \varphi_{\mathbf{x},i}(W) = w_{ij} + x_j\} \tag{18}$$

the set of indices where the maximum is achieved in $\varphi_{\mathbf{x},i}(W)$. When W and $\mathbf{x}$ will be clear from the context, we shall just denote it by J_i.
Let $H \in \mathbb{R}^{m\times n}$. Then for each $1 \leq i \leq m$, we also introduce the set

$$K_i \coloneqq \left\{k \in \{1,\dots,n\},\ h_{ik} > \max_{j\in J_i} h_{ij}\right\}. \tag{19}$$

Then we have the following result:

Proposition 1 *For fixed* $W, H \in \mathbb{R}^{m\times n}$ *and* $\mathbf{x} \in \mathbb{R}^n$*, let* $\varphi_{\mathbf{x},i}$*,* J_i *and* K_i *as defined by* (17), (18) *and* (19) *respectively for* $1 \leq i \leq m$*. Let*

$$\epsilon_i = \min_{k\in K_i} \frac{\varphi_{\mathbf{x},i}(W) - (w_{ik} + x_k)}{h_{ik} - \max_{j\in J_i} h_{ij}}, \quad 1 \leq i \leq m, \tag{20}$$

and $\epsilon = \min_{1\leq i\leq m} \epsilon_i$*. Then, for any* $\eta \in \mathbb{R}^+$ *we have*

$$\eta \in [0,\epsilon] \iff \delta_{\mathbf{x}}(W+\eta H) = \delta_{\mathbf{x}}(W) + \eta\left(\max_{j\in J_i} h_{ij}\right)_{1\leq i\leq m}. \tag{21}$$

Proof (Proposition 1). Let $\eta \in \mathbb{R}^+$, and let us note $\mathbf{b} := (\max_{j\in J_i} h_{ij})_{1\leq i\leq m}$. Then $\delta_{\mathbf{x}}(W+\eta H) = \delta_{\mathbf{x}}(W) + \eta\mathbf{b}$ if and only if $\varphi_{\mathbf{x},i}(W+\eta H) = \varphi_{\mathbf{x},i}(W) + \eta b_i$ for all $1 \leq i \leq m$. Now, the left-hand term writes

$$\varphi_{\mathbf{x},i}(W+\eta H) = \max\left(\max_{j\in J_i} w_{ij} + x_j + \eta h_{ij}, \max_{k\notin J_i} w_{ik} + x_k + \eta h_{ik}\right). \tag{22}$$

Since by definition $\varphi_{\mathbf{x},i}(W) = w_{ij} + x_j$ for any $j \in J_i$, we get

$$\varphi_{\mathbf{x},i}(W+\eta H) = \max\left(\varphi_{\mathbf{x},i}(W) + \eta b_i,\ \max_{k\notin J_i} w_{ik} + x_k + \eta h_{ik}\right). \tag{23}$$

Therefore $\varphi_{\mathbf{x},i}(W+\eta H) = \varphi_{\mathbf{x},i}(W) + \eta b_i$ if and only if for any $k \notin J_i$, $\varphi_{\mathbf{x},i}(W) + \eta b_i \geq w_{ik} + x_k + \eta h_{ik}$ which is equivalent to $\eta \leq \epsilon_i$, and the result follows. □

Given the definitions of Section 4, Proposition 1 readily gives that $\delta_{\mathbf{x}}$ is Bouligand differentiable everywhere and its B-derivative is

$$\delta'_{\mathbf{x}}(W; H) = \left(\max_{j\in J_i} h_{ij}\right)_{1\leq i\leq m}. \tag{24}$$

Furthermore, with the notations of Proposition 1,

$$\eta \in [0, \epsilon] \iff \delta_{\mathbf{x}}(W+\eta H) = \delta_{\mathbf{x}}(W) + \eta\delta'_{\mathbf{x}}(W; H). \tag{25}$$

It appears that for any W such that $J_i = \{j_i\}$ is a singleton for each $1 \leq i \leq m$ (the maximum is achieved only once for each $\varphi_{\mathbf{x},i}$), $\delta'_{\mathbf{x}}(W; H)$ is a linear function since $\max_{j\in J_i} h_{ij} = h_{ij_i} = \langle H_{i,:}, e_{j_i}\rangle$, where e_{j_i} is the vector with a one at index j_i and zeros elsewhere. Hence in that case $\delta_{\mathbf{x}}$ is Fréchet differentiable. One can check[4] that this happens for almost every W.

Derivative with respect to x With the same approach as previously, one can show that δ_W is B-differentiable with respect to $\mathbf{x}$, and its B-derivative is, for all $\mathbf{h} \in \mathbb{R}^n$,

$$\delta'_W(\mathbf{x}; \mathbf{h}) = \left(\max_{j\in J_i} h_j\right)_{1\leq i\leq m}, \tag{26}$$

Furthermore, for a fixed $\mathbf{h} \in \mathbb{R}^n$, changing only h_{ij} and h_{ik} for h_j and h_k in (19) and (20), it comes that for any $\eta \in \mathbb{R}^+$

$$\eta \in [0, \epsilon] \iff \delta_W(\mathbf{x}+\eta\mathbf{h}) = \delta_W(\mathbf{x}) + \eta\delta'_W(\mathbf{x}; \mathbf{h}). \tag{27}$$

Again, $\delta'_W(\mathbf{x}; \mathbf{h})$ is a linear function of $\mathbf{h}$ as soon as the maximum is achieved only once in $\{w_{ij} + x_j, 1 \leq j \leq n\}$, i.e. $J_i = \{j_i\}$, for each $1 \leq i \leq m$. In that case $\max_{j\in J_i} h_j = h_{j_i} = \langle \mathbf{h}, e_{j_i}\rangle$, hence $\delta'_W(\mathbf{x}; \mathbf{h}) = E\mathbf{h}$, where E is the matrix whose rows are the e_{j_i}s. Again, this case holds for almost every $\mathbf{x}$.

[4] Indeed the set of matrices for which the maximum in $\varphi_{\mathbf{x},i}(W)$ is achieved more than once, for a given i, is of zero Lebesgue measure.

5.2 Updating the parameters

Problem setting. Let us focus on the update of the parameter W of the dilation layer $\delta_W : \mathbb{R}^n \to \mathbb{R}^m$. In the context of the chain rule, we assume that later layers (those closer to the output) have transmitted an instruction direction $\mathbf{u} \in \mathbb{R}^m$, and δ_W is supposed to modify its parameter $W \leftarrow W + \Delta W$ so that $\delta_{\mathbf{x}}(W + \Delta W) - \delta_{\mathbf{x}}(W)$ maximizes the inner product with $\mathbf{u}$. More formally, just as in Eq. (6), we want to solve

$$\Delta W = \arg\max_{\|H\|=1} \langle \delta'_{\mathbf{x}}(W; H), \mathbf{u} \rangle \,, \tag{28}$$

where $\|\cdot\|$ denotes the Frobenius norm and, this time, we consider the Bouligand derivative (24) computed earlier. The reason for which we are faced with the same problem as with Fréchet derivative, is that the Bouligand one also provides the first order approximation (12). Without loss of generality, we assume $\|\mathbf{u}\| = 1$.

Solving (28) does not seem straightforward but an attempt could start by noticing that

$$\|\delta'_{\mathbf{x}}(W; H)\|^2 = \sum_{i=1}^{m} \left(\max_{j \in J_i} h_{ij} \right)^2 \le \sum_{i=1}^{m} \sum_{j=1}^{n} h_{ij}^2 = \|H\|^2 \tag{29}$$

hence $\|\delta'_{\mathbf{x}}(W; H)\| \le 1$ for $\|H\| = 1$, therefore $\langle \delta'_{\mathbf{x}}(W; H), \mathbf{u} \rangle \le \|u\| = 1$. This upper-bound is obviously achieved when $\delta'_{\mathbf{x}}(W;\ \cdot\ )$ is linear (i.e. $\delta_{\mathbf{x}}$ is F-differentiable at W), and for H such that $h_{ij_i} = u_i$, $1 \le i \le m$, and zero elsewhere, where we recall that in the F-differentiable case, j_i is the only index achieving the maximum in $\varphi_{\mathbf{x},i}(W)$, i.e. $J_i = \{j_i\}$. Indeed in that case, $\|H\| = 1$ and $\delta'_{\mathbf{x}}(W; H) = \mathbf{u}$.

Proposition of candidates $\boldsymbol{\Delta W}$. In the non-F-differentiable case (i.e. when at least one J_i has more than one element), without analytically solving (28), we can at least propose decent candidates, inspired by the F-differentiable case. Let $I^+ \coloneqq \{1 \le i \le m, u_i \ge 0\}$, $I^- \coloneqq \{1 \le i \le m, u_i < 0\}$ and for $1 \le i \le m$ let us note $p_i \coloneqq |J_i|$ the number of indices achieving the maximum in $\varphi_{\mathbf{x},i}(W)$. To make $\delta'_{\mathbf{x}}(W; H)$ similar to $\mathbf{u}$ while keeping $\|H\| = 1$, we propose:

- For $i \in I^+$, set $h_{ij_0} = u_i$ for any *one* $j_0 \in J_i$, and zero for $j \ne j_0$
- For $i \in I^-$, set $h_{ij} = \frac{u_i}{\sqrt{p_i}}$ for all $j \in J_i$, and zero for $j \notin J_i$.

Any such H verifies $\|H\| = 1$ and

$$\langle \delta'_{\mathbf{x}}(W; H), \mathbf{u} \rangle = \sum_{i \in I^+} u_i^2 + \sum_{i \in I^-} \frac{u_i^2}{\sqrt{p_i}} = 1 - \sum_{i \in I^-} \left(1 - \frac{1}{\sqrt{p_i}} \right) u_i^2. \tag{30}$$

We see that this quantity gets closer to one as the p_i get closer to one, and we recover the optimal bound in the F-differentiable case, which corresponds to $p_i = 1$ for all $1 \le i \le m$, or when all $u_i \ge 0$. Furthermore, we have the lower bound $\langle \delta'_{\mathbf{x}}(W; H), \mathbf{u} \rangle \ge \frac{1}{\sqrt{n}} > 0$, which is the left hand part of property (9). Note that numerical experiments show that better H can be found (for example in the neighbourhood of the proposed ones).

Choosing the learning rate. Recall that solving problem (28) is relevant as long as a good first order approximation $\delta_{\mathbf{x}}(W + \eta H) \approx \delta_{\mathbf{x}}(W) + \eta \delta'_{\mathbf{x}}(W; H)$ holds, since only in this case does the parameter update ensure a change in the output value towards a descent direction. Proposition 1 provides the exact range of learning rates for which this approximation is an equality. For our proposed H, it holds if and only if $\eta \in [0, \epsilon] \cap \mathbb{R}^+$, with

$$\epsilon = \min_{i \in I^-} \frac{\eta_i \sqrt{p_i}}{|u_i|}. \tag{31}$$

5.3 Message passing

Problem setting For the message passing, we are first faced with the same problem as (7) for F-differentiable functions, but with the B-derivative. Namely, given the received target direction $\mathbf{u}$, we want to find the best update direction $\Delta\mathbf{x}$ for $\mathbf{x}$,

$$\Delta\mathbf{x} = \arg \max_{\|\mathbf{h}\|=1} \left\langle \delta'_W(\mathbf{x}; \mathbf{h}), \mathbf{u} \right\rangle . \tag{32}$$

Assuming we can find a good enough $\mathbf{h}$, which would ensure $\langle \delta'_W(\mathbf{x}; \mathbf{h}), \mathbf{u} \rangle \geq 0$, i.e. the right hand part of property (9), then we have another problem, which is to guarantee property (10): that if $\langle \mathbf{v}, \mathbf{h} \rangle \geq 0$ for some $\mathbf{v}$, then $\langle \delta'_W(\mathbf{x}; \mathbf{v}), \mathbf{u} \rangle \geq 0$. To make sure the chain rule works, we could therefore focus on the problem

$$\text{Find } \mathbf{h} \in \mathbb{R}^n \text{ such that } \begin{cases} \|\mathbf{h}\| = 1 \\ \langle \delta'_W(\mathbf{x}; \mathbf{h}), \mathbf{u} \rangle \geq 0 \\ \forall \mathbf{v} \in \mathbb{R}^n, \;\; \langle \mathbf{v}, \mathbf{h} \rangle \geq 0 \Rightarrow \langle \delta'_W(\mathbf{x}; \mathbf{v}), \mathbf{u} \rangle \geq 0. \end{cases} \tag{33}$$

Proposition of candidates $\boldsymbol{\Delta}\mathbf{x}$ Recall that $\delta'_W(\mathbf{x}; \mathbf{h}) = (\max_{j \in J_i} h_j)_{1 \leq i \leq m}$, hence contrary to the case of parameter update (Section 5.2), the same h_j can contribute to different J_i, which makes a heuristic construction of $\mathbf{h}$ much more complicated. One exception is the case where the sets $J_{W_i, \mathbf{x}}$ are pairwise disjoint, as with the max-pooling layer with strides, for which the same kind of construction as in Section 5.2 can be done. However, this guarantees only the first two conditions of (33), but we cannot say much about the last one.

At this stage we have no provable solution for (33) except, obviously, in the F-differentiable case, where each J_i is a singleton $\{j_i\}$. In that case, as presented in Section 5.1, $\delta'_W(\mathbf{x}; \mathbf{h}) = E\mathbf{h}$, where E is the $m \times n$ matrix whose rows are the e_{j_i}s, each e_{j_i} being the vector with a one at index j_i and zeros elsewhere. Hence the solution of (32), and a solution of (33), is $\mathbf{h} = \frac{E^T\mathbf{u}}{\|E^T\mathbf{u}\|}$ if $E^T\mathbf{u} \neq 0$, and any unit vector $\mathbf{h}$ otherwise.

Therefore we propose as update candidate, one that generalizes the F-differentiable case, namely $\mathbf{h} = \frac{E^T\mathbf{u}}{\|E^T\mathbf{u}\|}$ but with E the matrix whose i-th row is $E_{i,:} = \sum_{j \in J_i} e_j$. Numerical experiments show that this proposition can sometimes violate the last two conditions of (33), but often behaves well.

Choosing the learning rate Hoping that the chosen $\mathbf{h}$ fulfills (33), we make the best of it by choosing a learning rate ensuring the first order equality (27). Hence once again we follow the construction inspired by Proposition 1. The choice of $\mathbf{h} = \frac{E^T\mathbf{u}}{\|E^T\mathbf{u}\|}$ yields no simplification of the expression of ϵ.

5.4 The convolutional case

The definitions (1) and (2) cover translation invariant dilations and erosions, as soon as $W \in \mathbb{R}^{n\times n}$ is a Toeplitz matrix. However, in Section 5.2, we assumed no "shared weights", i.e. each row of W was considered independent from the others, which allowed an easy choice for the parameter update.

To model the constraint on W due to translation invariance, we assume δ_W is represented by a vector $\mathbf{w} \in \mathbb{R}^p$, $p \leq n$, and the input variable $\mathbf{x} \in \mathbb{R}^n$ is now seen as a matrix $X \in \mathbb{R}^{n\times p}$ containing n blocks of length p. The dilation now writes

$$\delta_{\mathbf{w}}(X) = \delta_X(\mathbf{w}) \coloneqq \left(\max_{1\leq j\leq p} x_{ij} + w_j\right)_{1\leq i\leq n}. \tag{34}$$

Unfortunately, we see that even for the parameter update, which was rather favorable in the "dense" layer case of Section 5.2, we are in the same situation as in the message passing of Section 5.3, in the sense that finding good candidate for $\Delta\mathbf{w}$ is as difficult as solving (32). We would therefore apply the same heuristics, i.e. $\mathbf{h} = \frac{E^T\mathbf{u}}{\|E^T\mathbf{u}\|}$, where $E_{ij} = 1$ if j achieves the maximum in $\max_{1\leq j\leq p} x_{ij} + w_j$ and zero elsewhere. Concerning the learning rate, (27) holds.

5.5 Practical consequences

Position in the network, dense or convolutional layer. We saw that the chain rule mechanism is not guaranteed with morphological layers because of uncertainties in the message passing in general, and even in the parameter update for convolutional operators. Therefore, we expect better performance as a morphological layer is closer to the input of the network, and even more so if it is a dense layer. Typically, starting a neural pipeline with a dense dilation or erosion is the most favorable case with the update and learning rate propositions of Section 5.2. Furthermore, if each morphological layer is seen as a noisy message transmitter, then it is expected that many such layers in the same network may be hard to train with the chain rule paradigm.

Initialization In both the dense and convolutional cases, according to our propositions or even in the F-differentiable case, a parameter coefficient is not updated if it does not contribute to a maximum. In the dense case, w_{ij} is not modified if $j \notin J_i$, and in the convolutional one, w_j remains unchanged if $j \notin J_i$ for all i. In particular, if such coefficient is moved to $-\infty$, it will never be updated anymore. Now, consider for example that if the input variable $\mathbf{x}$ has values in $[0, 1]$ and at least one weight $w_{ij_1} \geq 0$, then the closer another weight w_{ij_2}, on the same line, will be to -1 the less likely it will be to achieve the maximum, and $w_{ij_2} \leq -1$ is equivalent to $w_{ij_2} = -\infty$. Therefore it seems preferable to initialize the parameters with non-negative values (typically, zero if input values in $[0, 1]$). Then, the proposed adaptive learning rates should avoid a divergence of weights to values from where they cannot come back.

6 Conclusion

In this paper we investigated the optimization of morphological layers based on the Bouligand derivative and the chain rule. We showed that despite the first order approximation of the B-derivative, its non-linearity makes morphological layers noisy message transmitter in the chain rule, where they are not F-differentiable. We clearly stated the problems to overcome in order to make this framework compatible with the chain rule. We also provided insights regarding the choice of the learning-rate for these layers, which seems much clearer than with classic layers. Future work will deal with addressing the stated problems and show the experimental consequences of the theoretical results presented here.

Acknowledgments

I would like to thank François Pacaud and Santiago Velasco-Forero for fruitful discussions on this topic.

References

1. Aouad, T., Talbot, H.: Binary morphological neural network. In: 2022 IEEE International Conference on Image Processing (ICIP). pp. 3276–3280 (2022)
2. 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)
3. Blusseau, S., Velasco-Forero, S., Angulo, J., Bloch, I.: Morphological adjunctions represented by matrices in max-plus algebra for signal and image processing. In: DGMM. pp. 206–218. Springer (2022)
4. Charisopoulos, V., Maragos, P.: Morphological perceptrons: Geometry and training algorithms. In: ISMM. pp. 3–15. Springer (2017)
5. Davidson, J.L., Ritter, G.X.: Theory of morphological neural networks. In: Digital Optical Computing II. vol. 1215, pp. 378–389 (1990)
6. Franchi, G., Fehri, A., Yao, A.: Deep morphological networks. Pattern Recognition **102**, 107246 (2020)
7. Hermary, R., Tochon, G., Puybareau, É., Kirszenberg, A., Angulo, J.: Learning grayscale mathematical morphology with smooth morphological layers. Journal of Mathematical Imaging and Vision **64**(7), 736–753 (2022)
8. Kirszenberg, A., Tochon, G., Puybareau, É., Angulo, J.: Going beyond p-convolutions to learn grayscale morphological operators. In: DGMM. pp. 470–482. Springer (2021)
9. Maragos, P.: A representation theory for morphological image and signal processing. IEEE Transactions on Pattern Analysis and Machine Intelligence **11**(6), 586–599 (1989)

10. Maragos, P.: Chapter Two - Representations for morphological image operators and analogies with linear operators. Advances in Imaging and Electron Physics, vol. 177, pp. 45 – 187. Elsevier (2013)
11. Matheron, G.: Random sets and integral geometry. Wiley, New York (1975)
12. Mondal, R., Dey, M.S., Chanda, B.: Image restoration by learning morphological opening-closing network. Mathematical Morphology-Theory and Applications **4**(1), 87–107 (2020)
13. Robinson, S.M.: Local structure of feasible sets in nonlinear programming, Part III: Stability and sensitivity. Springer (1987)
14. Scholtes, S.: Introduction to piecewise differentiable equations. Springer (2012)
15. Velasco-Forero, S.: Morpholayers (2020)
16. Velasco-Forero, S., Rhim, A., Angulo, J.: Fixed Point Layers for Geodesic Morphological Operations. In: BMVC. London, United Kingdom (Nov 2022)
17. Wilson, S.S.: Morphological networks. In: Visual Communications and Image Processing IV. vol. 1199, pp. 483–496 (1989)
18. Zhang, Y., Blusseau, S., Velasco-Forero, S., Bloch, I., Angulo, J.: Max-plus operators applied to filter selection and model pruning in neural networks. In: ISMM. pp. 310–322. Springer (2019)