Binary morphological neural network

Théodore Aouad, Hugues Talbot

BINARY MORPHOLOGICAL NEURAL NETWORK

Theodore Aouad[1], *Hugues Talbot*[1]

[1]CentraleSupélec, Université Paris-Saclay, Inria. Gif-sur-Yvette, France

ABSTRACT

In the last ten years, Convolutional Neural Networks (CNNs) have formed the basis of deep-learning architectures for most computer vision tasks. However, they are not necessarily optimal. For example, mathematical morphology is known to be better suited to deal with binary images. In this work, we create a morphological neural network that handles binary inputs and outputs. We propose their construction inspired by CNNs to formulate layers adapted to such images by replacing convolutions with erosions and dilations. We give explainable theoretical results on whether or not the resulting learned networks are indeed morphological operators. We present promising experimental results designed to learn basic binary operators, and we have made our code publicly available online.

***Index Terms*—** Mathematical morphology, binary, deep learning, machine learning, image processing

1. INTRODUCTION

Convolutional Neural Networks (CNNs) constitute the basis of most deep-learning architectures. These can learn complex task-specific processes while requiring only input-output pairs, albeit sometimes in large numbers. Since their inception in the mid-1990s, they have achieved outstanding results in computer vision and have become the go-to technology for many computer vision tasks, provided enough annotated data is available.

However, standard literature on image processing states that some tasks remain for which convolutions are not optimal. Mathematical Morphology (MM) [1] is one of these. For many applications, MM operators are more suitable than convolution-based methods, particularly when dealing with binary or discrete images. However, finding the right sequence of operations and the right structuring elements can be difficult and time-consuming depending on the problem at hand [2]. Our objective is to mimic the way CNNs are built on convolutional filters and create a morphological network that can learn a compact sequence of operators together with their optimal parameters. Morphological networks can also be used conjointly with CNNs to learn the morphological operators that would be otherwise manually engineered, as in [3].

Learning morphological operators is not new, whereas the trend of replacing the convolution of CNNs with morphological operations is recent. Some researchers have investigated the use of the max-plus algebra [4, 5], for example, to perform image filtering (de-raining and de-hazing) [6]. Others have replaced the non-differentiable max / min operators by differentiable approximations, e.g. the adaptative morphological layer [7], the PConv layer [8], or even the $\mathcal{L}$*Morph* and $\mathcal{S}$*Morph* layers [9]. All these methods were studied in the context of grey-scale morphology.

In this work, we seek to learn binary morphological operators from binary image inputs. End-to-end learning of these operators could be helpful in shape analysis. We first introduce the Binary Structuring Element (BiSE) neuron, which can learn erosion and dilation together with a structuring element. The BiSE neuron is built using convolution and benefits from the highly optimized implementations of this operation. By stacking multiple BiSE, we build a Binary Morphological Neural Network (BiMoNN). We give theoretical explainability of the BiSE such that each learned morphological operator can be recovered. Binarizing these networks can lead to faster and cost-efficient deep networks for inference [10, 11, 12].

Our code is publicly available online at `https://github.com/TheodoreAouad/Bimonn_ICIP2022`.

2. METHOD

2.1. Mathematical Morphology

Mathematical morphology [1] was created to study porous materials. It is based on set theory and is well suited to studying binary images. An image of dimension d ($d = 2$ for 2D images or $d = 3$ for 3D images) is seen as a subset of $\mathbb{Z}^d$: $I \subset \mathbb{Z}^d$. Morphological operations transform I based on a small structuring element $S \subset \mathbb{Z}^d$. The two fundamental operations are dilation and erosion.

Definition 1 *The dilation ($\oplus$) and erosion ($\ominus$) of an image $I \subset \mathbb{Z}^d$ by a structuring element $S \subset \mathbb{Z}^d$ are defined as:*

$$\delta_S(I) = I \oplus S = \bigcup_{s \in S} (I + s) \tag{1}$$

$$\varepsilon_S(I) = I \ominus \check{S} = \bigcap_{s \in \check{S}} (I + s), \tag{2}$$

where $\check{S}$ is the symmetric of S with respect to the origin.

These operators thus defined are *adjunct*, i.e. $\forall I, J, S, I \subseteq \varepsilon_S(J) \iff \delta_S(I) \subseteq J$. All basic morphological operations are obtained from erosions and dilations. The opening is the application of erosion followed by its adjunct dilation, and the closing is the other way around. Here we consider any composition of dilations or erosions, extending to any sequence of openings or closings.

Erosions and dilations are similar to convolutions: the structuring elements can be interpreted as the kernel for the convolutional filter, and the sum is replaced by max or min operator. Therefore, we must establish a model that can learn the structuring element and operation type: erosion or dilation.

2.2. Binary Structuring Element Neuron

We now define the BiSE (Binary Structuring Element) neuron, which can learn both the operation and the structuring element. The BiSE neuron is built upon the convolution operation. First, we notice that the dilation and erosion can be exactly expressed using convolution. Note that in practice, for the erosion, we learn the symmetric $\varepsilon_{\check{S}}$ such that $\varepsilon_{\check{S}}(I) = I \ominus S$.

Proposition 1 (Morphological operators from convolution) *Let $S \subset \mathbb{Z}^d$ be a binary structuring element and $X \subset \mathbb{Z}^d$ be a binary image.*

$$X \oplus S = \left(\mathbb{1}_X \circledast \mathbb{1}_S \geqslant 1\right) \tag{3}$$

$$X \ominus S = \left(\mathbb{1}_X \circledast \mathbb{1}_S \geqslant |S|\right) \tag{4}$$

For the same structuring element S, the difference between dilation and erosion is determined by a scalar. Learning the operation is the same as learning this scalar.

To learn $S \subset \mathbb{Z}^d$, first we suppose that S is bounded: $S \subseteq \Omega$ with $|\Omega| < +\infty$. Let Ω be the grid bounded by some integer n, $\Omega = \mathbb{Z}^d \cap [-n, n]^d$. Similarly to [11], we define a relaxed weight $W \subset \mathbb{R}^\Omega$, we apply a smooth increasing threshold function ξ such that $\xi(W(i)) \approx 1$ if $i \in S$, else $\xi(W(i)) \approx 0$. We use the hyperbolic tangent:

$$\xi(x) = \frac{1}{2}\tanh(x) + \frac{1}{2} \tag{5}$$

We introduce the softplus function $f^+ : x \in \mathbb{R} \mapsto \ln(1 + \exp(x)) + 0.5$ to ensure that $f^+(x) > 0.5$.

Definition 2 (BiSE neuron) *Let $W \in \mathbb{R}^\Omega$ be a weight matrix, $b \in \mathbb{R}$ a bias and $p \in \mathbb{R}$ a scaling number. We define a **BiSE (Binary Structuring Element) neuron** as follow:*

$$\epsilon_{W,b,p} : x \in [0,1]^{\mathbb{Z}^d} \mapsto \xi(p(x \circledast \xi(W) - f^+(b))) \in [0,1]^{\mathbb{Z}^d} \tag{6}$$

First, the weights are thresholded. Then we apply the morphological operation: we perform the convolution and subtract a bias. The bias is forbidden to become negative (in practice, $f^+(b) > 0.5$): we avoid the bias at 0, which leads to constant output and zero-grad zones. Finally, we have to threshold this result. Before thresholding, the result is multiplied by a scaling factor p.

The BiSE neuron can learn erosion, dilation, and the associated structuring element. The weights W learn the structuring element, and the bias b determines the operation. The scaling number p has two purposes. It determines how close to binary the output is, and if $p < 0$, the output is inverted, so in theory, we could learn the complementation as well.

Our thresholding function, the hyperbolic tangent, is smooth to allow back-propagation; we do not deal with binary images. We thus introduce the concept of almost binary images, which are more flexible and are easier to handle than binary images.

Definition 3 (Almost Binary Image) *We say an image $I \in [0,1]^{\mathbb{Z}^d}$ is **almost binary** if there exists $u < v \in [0,1]$ such that $I(\mathbb{Z}^d) \cap]u, v[= \varnothing$. We denote this set $\mathcal{I}(u,v)$.*

A pixel value of an almost binary image is either close to 0 or close to 1.

When is a BiSE neuron equivalent to dilation or erosion by a structuring element S? The following propositions allow us to perform a verification for a given structuring element. Given the weights W and the bias b, we can check if, in its current state, the BiSE neuron is a dilation or erosion by this structuring element.

Proposition 2 (Dilation / Erosion Equivalence) *We assume the weights are thresholded: $W \in [0,1]^\Omega$. Given an almost binary input in $\mathcal{I}(u,v)$*

- *$\epsilon_{W,b,+\infty}$ is a dilation by S if and only if*

$$\sum_{i \in \Omega \setminus S} w_i + u \sum_{i \in S} w_i \leqslant b < v \min_{i \in S} w_i \tag{7}$$

- *$\epsilon_{W,b,+\infty}$ is an erosion by S if and only if*

$$\max_{j \in S}\left(\sum_{i \in S \setminus j} (w_i) + u \cdot w_j\right) \leqslant b < v \sum_{i \in S} W_i \tag{8}$$

*If either of these expressions is fulfilled, we say that the BiSE neuron is **activated**.*

If the weights and bias are correctly learned, the structuring element can be recovered by thresholding the weights for some value. The suitable threshold is given in proposition 3.

Proposition 3 (Linear Check) *Let us assume the BiSE is activated for almost binary images $\mathcal{I}(u,v)$. Let b be the BiSE bias, let W be the normalized weights ($W \in]0,1[^\Omega$). Then there exists $\tau \in \mathbb{R}$ such that $S = \{i \in \Omega \mid W(i) \geqslant \tau\}$.*

- *If the BiSE is a dilation*

$$\tau = \frac{b}{v} \tag{9}$$

- *If the BiSE is an erosion*

$$\tau = \frac{\sum_{k \in \Omega} W_k - b}{1 - u} \tag{10}$$

To find out the learned structuring element and operation, we only need to check both thresholds. This can be done in $\mathcal{O}(|\Omega|)$ operations.

Erosion and dilation are dual operations: applying a dilation is the same as applying an erosion to the background. This property allows us to recover the inequalities of one operation to get its dual. Finally, if the BiSE neuron is activated, then its output is almost binary. The notion of almost binary images is justified: we now deal with almost binary images instead of binary images.

2.3. BiMoNN

We can now define the Binary Morphological Neural Network (BiMoNN) as a composition of multiple BiSEs:

$$\text{BiMoNN} = \epsilon_L \circ ... \circ \epsilon_1 \tag{11}$$

If each BiSE is activated, the BiMoNN's inputs and outputs are almost binary. In theory, this framework can learn any sequence of dilations or erosions with any structuring element, including opening and closing.

We learn the BiMoNN using the classical deep learning framework. Given a loss $\mathcal{L}$, we use a masked version of $\mathcal{L}$. Borders act depending on our interpretation of the value out-of-bounds pixels. To avoid this problem, we mask the borders of size kernel shape divided by two. Parameters are updated with the Adam [13] optimizer. We initialize the biases at $f^+(2) = 0.63$. The weights W of the convolutions have kaiming uniform initialization [14].

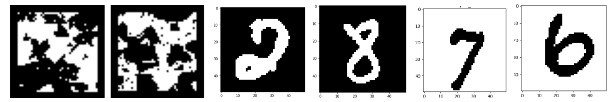
3. EXPERIMENTS

3.1. Datasets

We create a dataset of generated images that we call Diskorect. Each image is made of random-shaped and random-oriented rectangles and disks. Then, we add some random Bernoulli noise. Finally, complementation is applied half the time. The images dimensions are 50×50.

MNIST [15] is a dataset of 70000 handwritten digits of size 28×28. The images are grey-level. To be able to work with structuring elements of size $(7, 7)$, we reshape them to $(50, 50)$, with a cubic interpolation [16]. Then we threshold the image to recover binary images.

In order to analyze the duality of operators, we also test our network on the complementation of MNIST, which we call inverted MNIST.



(a) Diskorect (b) MNIST (c) Inverted MNIST

Fig. 1. Datasets example.

3.2. Experiment description

We check the ability of a BiMoNN to learn basic morphological operators. We attempt to learn the erosion and dilation (table 1), as well as the opening (erosion then dilation) and closing (dilation then erosion) (table 2). The protocol is the same for each operation: the morphological operator is applied to the input image to create the target y_i. To learn the erosion and dilation, we train a single BiSE neuron. To learn the opening and closing, we stack two BiSE neurons. We follow the DICE [17] of the target images vs. predicted images. We also check if each BiSE neuron is activated.

For both MNIST and inverted MNIST, the structuring elements are of size 5×5 for the erosion and dilation; the transformations are too significant compared to the size of the digits otherwise.

The parameter p is fixed $p = 4$. For Diskorect, we use the Dice loss [18] with learning rate 0.01 for the erosion / dilation and 0.001 for the opening / closing. For MNIST, we use the MSE loss with learning rate 0.1 for the erosion / dilation and 0.01 for the opening / closing.

Table 1. Results on erosion and dilation. DICE error $(1 - \text{DICE})$ is presented for each case. ✓ indicates if the neuron is activated.

Dataset	Diskorect			MNIST			Inverted MNIST		
Operation	Disk	Stick	Cross	Disk	Stick	Cross	Disk	Stick	Cross
Target									
Dilation ⊕	✓ 0.000	✓ 0.000	× 0.000	× 0.000	✓ 0.000	× 0.000	× 0.000	✓ 0.000	× 0.000
Erosion ⊖	✓ 0.000	✓ 0.000	✓ 0.000	× 0.000	✓ 0.000	× 0.000	× 0.000	✓ 0.000	× 0.000

Table 2. Results on opening and closing. DICE error $(1 - \text{DICE})$ is presented for each case. ✓ indicates if the neuron is activated.

Dataset	Diskorect			MNIST			Inverted MNIST		
Operation	Disk	Stick	Cross	Disk	Stick	Cross	Disk	Stick	Cross
Target									
Opening ○	× × 0.072	✓ ✓ 0.000	✓ ✓ 0.002	× ✓ 0.008	✓ ✓ 0.000	× ✓ 0.001	× × 0.006	✓ × 0.001	× × 0.012
Closing •	× × 0.038	✓ ✓ 0.000	✓ × 0.000	× × 0.009	✓ × 0.001	× × 0.020	× × 0.009	✓ ✓ 0.000	✓ ✓ 0.000

4. DISCUSSION

4.1. Erosion and Dilation

For any optimizer or any initialization, on both MNIST and Diskorect, the dilation is learned perfectly for all structuring elements: after a few hundred iterations, the DICE is 1 for the validation set. The BiSE are all activated except for the cross. On the Diskorect dataset, the BiSE neurons are activated after a few thousand iterations for the disk and stick.

For the erosion, on Diskorect, we obtain similar results but with slower convergence. Sometimes the weights are darker than in the dilation case. This is not a problem: if the BiSE is activated, the weights do not need to be as high as 1. On MNIST, the learned disk is limited to its border, which is not surprising: given S and its border ∂S, the difference between $I \ominus S$ and $I \ominus \partial S$ is small and only visible on a dataset with small holes. While the DICE is not 1, the error is small at 0.003.

The perfect metric is reached before the BiSE neurons are activated. Indeed, it is possible for the inequality not to be fulfilled, while the BiSE is a dilation only for a specific dataset $D \subsetneq \mathbb{Z}^d$. Could a necessary and sufficient condition be established for a specific dataset to know if the BiSE is a dilation on this dataset? This question is left for future work.

4.2. Opening and Closing

On Diskorect, for both opening and closing, we learn perfectly the stick and cross, and almost all the BiSE are activated. However, the disk is more challenging to learn for opening (resp. closing): the DICE is still high at 0.93 (resp. 0.96).

On MNIST, the opening is learned well for all structuring elements. However, the training on closing returns chaotic weights for the disk and cross. To explain this, we notice that closing with S has little effect on the image for this dataset. Therefore many weight combinations yield a similar transformation. See the disk closing: even with totally different structuring elements, the error is small at 0.002. For the inverted MNIST, the situation is almost the same, but for the dual operators: we will dive into that in the next paragraph.

We compare to *$\mathcal{S}$Morph* and *$\mathcal{L}$Morph* [9]. On both datasets, they achieve good results on erosion and dilation, with perfect metric and good structuring elements. However, they fail to converge on the opening and closing and present stability issues (because of an exponential in their formulation). This is not surprising as they are both built on a differentiable softmax function that is not suitable to deal with binary elements.

4.3. A few words on duality

In mathematical morphology, two operators δ and ε are dual if $\forall X$, $\overline{\delta(X)} = \varepsilon(\overline{X})$. This is the case for the couples (dilation, erosion) and (opening, closing). Let D be a dataset and $\overline{D}$ the set of its complementations. We formulate the hypothesis $\mathcal{H}$: are the training of δ on D and the training of ε on $\overline{D}$ similar?

Let D be the Diskorect dataset. By construction, $\overline{D} = D$ and the training of dual operators should be the same. For the erosion and dilation, the results are not exactly the same. For the erosion, the structuring elements are darker, and the model takes longer to converge: from 10 times slower on Diskorect to a hundred times slower on MNIST. We still manage to learn both operations. On the other hand, the opening and closing do not behave the same on the closing, with dissimilar weights. These results contradict $\mathcal{H}$.

Now, let D be the MNIST dataset. The dilation behaves identically for both D and $\overline{D}$, and the same can be said for the erosion, which goes against $\mathcal{H}$. However, even though the learned weights seemed chaotic, the closing on D works similarly to the opening on $\overline{D}$, which corroborates $\mathcal{H}$. This similarity also happens for the opening on D and closing on $\overline{D}$, being only different for the disk.

The case of MNIST and its invert suggests that there is a strong link between the learning of dual operators. However, they do not learn exactly the same. We leave the study of this link to future work.

5. CONCLUSION

We created a neural network built to operate on binary images. We successfully learn some morphological operations: results on dilations and erosions are perfect. Overall, we achieved good results on the opening and closing, only failing for the disk and more complicated operations. We also provide explainability results to understand the operation of each neuron.

We sometimes reach a perfect metric without the neurons being theoretically morphological operators. This raises the need for more relaxed inequalities depending on the dataset properties. Enforcing that each BiSE is activated (e.g., using a regularization loss) would allow us to binarize the entire network.

To complete our network, we should also learn the complementation and multiple filters (e.g., to learn the top-hat operators). The complementation can be done by learning the parameter P and allowing it to be negative. While the BiSE could theoretically complement its results, it was not yet demonstrated in practice: we leave it to future work. To fully simulate the CNNs' multiple-filter structure, we will incorporate intersection and union of filters.

These results demonstrate that we can learn simple morphological operators. In the near future, we will investigate its integration in more complex pipelines to leverage useful morphological information, with a view to construct high-performing binary deep networks.

6. REFERENCES

[1] J. SERRA, "Image analysis and mathematical morphology," 1982.

[2] P. Soille, *Morphological Image Analysis, principles and applications*, Springer-Verlag, 2nd edition, 2003, ISBN 3-540-42988-3.

[3] Suprosanna Shit, Johannes C. Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien P. W. Pluim, Ulrich Bauer, and Bjoern H. Menze, "clDice - a Novel Topology-Preserving Loss Function for Tubular Structure Segmentation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, June 2021, pp. 16555–16564, IEEE.

[4] Ranjan Mondal, Moni Shankar Dey, and Bhabatosh Chanda, "Image restoration by learning morphological opening-closing network," *Mathematical Morphology-Theory and Applications*, vol. 4, no. 1, pp. 87–107, 2020.

[5] Gianni Franchi, Amin Fehri, and Angela Yao, "Deep morphological networks," *Pattern Recognition*, vol. 102, pp. 107246, 2020.

[6] Ranjan Mondal, Deepayan Chakraborty, and Bhabatosh Chanda, "Learning 2d morphological network for old document image binarization," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 65–70.

[7] Yucong Shen, Xin Zhong, and Frank Y Shih, "Deep morphological neural networks," *arXiv preprint arXiv:1909.01532*, 2019.

[8] Jonathan Masci, Jesús Angulo, and Jürgen Schmidhuber, "A learning framework for morphological operators using counter–harmonic mean," in *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer, 2013, pp. 329–340.

[9] Alexandre Kirszenberg, Guillaume Tochon, Élodie Puybareau, and Jesus Angulo, "Going beyond p-convolutions to learn grayscale morphological operators," in *International Conference on Discrete Geometry and Mathematical Morphology*. Springer, 2021, pp. 470–482.

[10] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio, "Binarized neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[11] Minje Kim and Paris Smaragdis, "Bitwise neural networks," *arXiv preprint arXiv:1601.06071*, 2016.

[12] Taylor Simons and Dah-Jye Lee, "A review of binarized neural networks," *Electronics*, vol. 8, no. 6, pp. 661, 2019.

[13] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2017.

[14] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[15] Li Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[16] Robert Keys, "Cubic convolution interpolation for digital image processing," *IEEE transactions on acoustics, speech, and signal processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[17] Lee R Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[18] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," 2016.

Appendices

A. USAGE OF DUALITY FOR EROSION AND DILATION EQUIVALENCE

Erosion and dilation are dual operations, meaning $\forall X, S \in \{0,1\}^\Omega, X \oplus S = \overline{\bar{X} \ominus S}$: applying a dilation is the same as applying an erosion to the background. This property allows us to recover the inequalities of the dilation using those of the erosions, and the other around. Let us show this in the binary case.

Let us consider two BiSE layers ϵ^1_{W,b_1} and ϵ^2_{W,b_2} sharing the same weights, but the first one being a dilation and the second one being an erosion. Let us denote u_e, v_e the bounds for the erosion and u_d, v_d the bounds for the dilation. We have:

$$u_d = \sup_{X \in \{0,1\}^\Omega, i \in \Omega} \{\mathbb{1}_X \circledast W(i) \mid i \notin X \oplus S\} \quad (12)$$

$$v_d = \inf_{X \in \{0,1\}^\Omega, i \in \Omega} \{\mathbb{1}_X \circledast W(i) \mid i \in X \oplus S\} \quad (13)$$

Given that $\mathbb{1}_X \circledast W + \mathbb{1}_{\bar{X}} \circledast W = \sum_{i \in \Omega} w_i$, we can write:

$$\begin{aligned}
v_e &= \inf_{X \in \{0,1\}, i \in \Omega} \{\mathbb{1}_X \circledast W(i) \mid i \in X \ominus S\} && (14)\\
&= \inf_{X \in \{0,1\}, i \in \Omega} \{\mathbb{1}_{\bar{X}} \circledast W(i) \mid i \in \bar{X} \ominus S\} && (15)\\
&= \inf_{X \in \{0,1\}, i \in \Omega} \{\mathbb{1}_{\bar{X}} \circledast W(i) \mid i \in \overline{X \oplus S}\} && (16)\\
&= \inf_{X \in \{0,1\}, i \in \Omega} \{\mathbb{1}_{\bar{X}} \circledast W(i) \mid i \in \overline{X \oplus S}\} && (17)\\
&= \inf_{X \in \{0,1\}, i \in \Omega} \{\sum_{i \in \Omega} w_i - \mathbb{1}_X \circledast W(i) \mid i \notin X \oplus S\} && (18)\\
&= \sum_{i \in \Omega} w_i - \sup_{X \in \{0,1\}, i \in \Omega} \{\mathbb{1}_X \circledast W(i) \mid i \notin X \oplus S\} && (19)\\
v_e &= \sum_{i \in \Omega} w_i - u_d && (20)
\end{aligned}$$

The same can be done for the other bound: $u_e = \sum_{i \in \Omega} w_i - v_d$.

B. BINARY STRUCTURING ELEMENT LAYER

One of CNNs' strengths is the ability to learn multiple filters per layer. We also want to be able to learn multiple filters. In CNNs, each final channel is a sum of one filter per input channel. In our case, the final result is either a union or an intersection of the morphological operators (dilation or erosion). Therefore, we want a layer that can learn the union or intersection of any combination of inputs. Let us consider n binary images $x_1, ..., x_n \subset \Omega$. Let $\mathcal{C} \subset [\![1, n]\!]$. Then the intersection and union are given as:

$$\mathbb{1}_{\bigcap_{i \in \mathcal{C}} x_i} = \Big(\sum_{i \in \mathcal{C}} \mathbb{1}_{x_i} \geqslant |\mathcal{C}|\Big) \quad (21)$$

$$\mathbb{1}_{\bigcap_{i \in \mathcal{C}} x_i} = \Big(\sum_{i \in \mathcal{C}} \mathbb{1}_{x_i} \geqslant 1\Big) \quad (22)$$

As with the BiSE, we can use a single scalar to discriminate between the union or the intersection. To learn the set $\mathcal{C}$, we can use a parameter β_i for each image. This gives the following definition:

Definition 4 **(**LUI**)** *Let $\beta = (\beta_1, ..., \beta_c) \in \mathbb{R}^c$. Let ξ be a smooth increasing threshold. Let $b \in \mathbb{R}_+$ be a bias and $p \in \mathbb{R}$ a scaling factor. We define the LUI (Layer Intersection Union) as a thresholded linear combination:*

$$LUI^\beta : x \in (\mathbb{Z}^d)^c \mapsto \xi\bigg(p\Big(\sum_{i=1}^{c} \beta_i x_i - f^+(b)\Big)\bigg) \in \mathbb{Z}^d \quad (23)$$

A LUI layer can learn any intersection or union of any number of almost binary inputs. We denote $\mathcal{I} = \times_{k=1}^{n} \mathcal{I}(u_k, v_k)$ the set of images with n almost binary channels.

Proposition 4 **(**LUI **intersection / union equivalence)** *Let $n \in \mathbb{N}^*$ and $\mathcal{C} \subset [\![1, n]\!]$. Let $b \in \mathbb{R}$. Let $u_1 < v_1, ..., u_n < v_n \in [0, 1]$. Let $\beta \in \mathbb{R}^n_+$.*

- *LUI is an intersection by $\mathcal{C}$ if and only if*

$$\sum_{k=1}^{n} \beta_k - \min_{k \in \mathcal{C}} \Big[(1 - u_k)\beta_k\Big] \leqslant b < \sum_{k \in \mathcal{C}} \beta_k v_k \quad (24)$$

- *LUI is a union by $\mathcal{C}$ if and only if*

$$\sum_{k \in \mathcal{C}} \beta_k u_k + \sum_{k \in [\![1,n]\!] \setminus \mathcal{C}} \beta_k \leqslant b < \min_{k \in \mathcal{C}} (\beta_k v_k) \quad (25)$$

Like for the BiSE, if the LUI is properly learned (*i.e. inequalities are respected*), the set $\mathcal{C}$ can be found by thresholding the $(\beta_i)_i$ for a certain value. Moreover, if the inequalities are strict, if all the channels of an input image I are almost binary in $\mathcal{I}(u, v)$, then the output $LUI^{\beta,b}(I)$ is almost binary: $LUI^{\beta,b}(I) \in \mathcal{I}(\xi(u - b), \xi(v - b))$.

We combine the BiSE neurons and the LUI to be able to learn the morphological operators, and aggregate them as unions or intersections.

Definition 5 (BiSEL) *A BiSEL (BiSE Layer) is the combination of multiple BiSE and multiple LUI. Let $(\epsilon_{n,k})$ be $N * K$ BiSE and $(LUI_k)_k$ be K LUI. Then we define a BiSEL as:*

$$\phi : x \in (\mathbb{Z}^d)^n \mapsto \bigg(LUI_k\Big[\big(\epsilon_{n,k}(x_n)\big)_n\Big]\bigg)_k \in (\mathbb{Z}^d)^K \quad (26)$$

Given almost binary inputs, the outputs of the BiSEL are also almost binary.

This follows the same logic as CNN. In CNN, at each layer, we have k filters. Each filter applies one convolution to each channel, then we apply a linear combination to all convoluted channels. In BiSEL, we have K filters (the number of LUI). For each filter, we apply a morphological operation to a channel, then we aggregate by taking the intersection or union of any channels. See figure 2.

The parameters are the BiSE weights $\{W_{n,k}\}_{n,k} \subset \mathbb{R}^{\Omega}$, the BiSE biases $\{b_{n,k}\}_{n,k} \subset \mathbb{R}$, the LUI parameters $\{\beta_{n,k}\}_{n,k} \subset \mathbb{R}$ and the LUI biases $\{b^{l}_{n,k}\}_{n,k} \subset \mathbb{R}$. There are $NK(|\Omega| + 3)$ parameters.

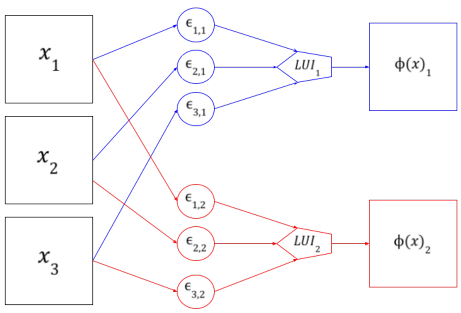


Fig. 2. Schema of BiSEL. Input x with 3 input channels. Output $\phi(x)$ with 2 channels.