

# Morphological Network: How Far Can We Go with Morphological Neurons?

Ranjan Mondal<sup>1</sup>

<https://ranjanz.github.io>

Sanchayan Santra<sup>2</sup>

<https://san-santra.github.io>

Soumendu Sundar Mukherjee<sup>3</sup>

<https://soumendu041.gitlab.io>

Bhabatosh Chanda<sup>3</sup>

[bchanda57@gmail.com](mailto:bchanda57@gmail.com)

<sup>1</sup> Samsung Research

Bangalore

India

<sup>2</sup> Institute for Dataability Science

Osaka University

Osaka, Japan

<sup>3</sup> Indian Statistical Institute

Kolkata

India

---

## Abstract

Morphological neurons, that is morphological operators such as dilation and erosion with learnable structuring elements, have intrigued researchers for quite some time because of the power these operators bring to the table despite their simplicity. These operators are known to be powerful nonlinear tools, but for a given problem coming up with a sequence of operations and their structuring element is a non-trivial task. So, the existing works have mainly focused on this part of the problem without delving deep into their applicability as generic operators. A few works have tried to utilize morphological neurons as a part of classification (and regression) networks when the input is a feature vector. However, these methods mainly focus on a specific problem, without going into generic theoretical analysis. In this work, we have theoretically analyzed morphological neurons and have shown that these are far more powerful than previously anticipated. Our proposed morphological block, containing dilation and erosion followed by their linear combination, represents a sum of hinge functions. Existing works show that hinge functions perform quite well in classification and regression problems. Two morphological blocks can even approximate any continuous function. However, to facilitate the theoretical analysis that we have done in this paper, we have restricted ourselves to the 1D version of the operators, where the structuring element operates on the whole input. Experimental evaluations also indicate the effectiveness of networks built with morphological neurons, over similarly structured neural networks.

## 1 Introduction

Mathematical morphology is a set and lattice theoretic technique for the analysis of geometrical structures. Although it originated from the theoretical study of the geometry of porous materials, it is currently extensively used in the domain of digital image processing. It serves as a non-linear tool for processing digital images. Morphological operators decompose objects or shapes into meaningful parts which helps in understanding them in

terms of the elements. Since the identification of objects and their features are directly correlated with their shapes and arrangement, morphological methods are quite suited for visual tasks [8]. However, coming up with a sequence of transformation and their parameters for a given problem is not straightforward and requires expert knowledge of the problem. To this end, researchers have tried to automatically learn the parameters (structuring element) of mathematical morphology operators (i.e., dilation and erosion). These learnable structures are termed as *morphological neurons*. Although this idea is motivated by the use of these operators in image processing, they are quite generic and can be used for tasks like classification [14, 21] and regression [5]. On applying morphological neurons for specific tasks. Theoretical analysis of this structure and its properties are lacking in the literature. Intending to fill this gap, in this paper, we have theoretically analysed the properties of morphological neurons and shown that a specific arrangement of the neurons can approximate any continuous function. To be more precise, we have defined a structure called a *Morphological Block* and shown that a sequence of two morphological blocks can work as a universal approximator. However, to facilitate the theoretical analysis, in this paper, we have restricted ourselves to the 1D version of the morphological operators, where the operators work over the whole input at once, not locally.

The contributions of this work can be summarized as follows.

1. We have theoretically analyzed the properties of the morphological neurons and have shown that not all sequences of neurons are useful as some of them may be represented using fewer neurons.
2. One such useful sequence is a layer of both dilation and erosion operations, followed by a layer computing a linear combination of the outputs. We call this a *Morphological Block*.
3. We have shown that a morphological block represents a sum of hinge functions. Sum hinge functions work well for tasks like regression, classification and function approximation [9].
4. We have proved that a sequence of two morphological blocks can approximate any continuous function over arbitrary compact sets.
5. We have shown due to the nonlinear nature of these operators, the neurons can learn more complex decision boundaries with a similar number of parameters.

The rest of the paper is organized as follows. Existing works that has experimented with morphological neurons are briefly outlined in Section 2. In Section 3, we describe morphological neurons and theoretically analyse their properties. Section 4 provides empirical validation of the proposed structure. Finally, concluding remarks are presented in Section 5.

## 2 Related works

The use of morphological operations in a learning framework is first proposed by Davidson and Hummer [9] in their effort to learn the structuring element of dilation operation on images. A similar effort has been made to learn the structuring elements in more recent work by Masci *et al.* [22]. The use of morphological neurons for problems other than images is first proposed by Ritter and Sussner [16]. They propose to use single-layer network architecture and focused only on the binary classification task. To classify the data, their proposed

network is able to learn two axis-parallel hyperplanes as the decision boundary. This single-layer architecture has later been extended to two-layer architecture by Sussner [24]. This two-layer architecture can learn multiple axis-parallel hyperplanes, and therefore is able to solve arbitrary binary classification tasks. But, in general, the decision boundaries may not be axis-parallel, and so, a large number of hyperplanes may need to be learned by the network. So, Barmpoutis and Ritter [10] proposed to learn an additional rotational matrix that rotates the input before trying to classify data using axis-parallel hyperplanes. In a separate work by Ritter *et al.* [18] the use of  $L^1$  and  $L^\infty$  norm has been proposed as a replacement of the *min/max* operation of dilation and erosion in order to smooth the decision boundaries. Ritter and Urcid [17] introduced the dendritic structure of biological neurons to the morphological networks. This structure creates hyperbox-based decision boundaries instead of hyperplanes. The authors have proved that hyperboxes can estimate any compact region and, thus, any two-class classification problem can be solved. A generalization of this structure to the multiclass case has also been done by Ritter *et al.* [19]. Experimentation with network architecture has also been attempted by Sussner and Esmi [24], where they propose a new structure called morphological neurons with competitive learning. In this setting, the *argmax* operator is utilized at the output of several neurons to implement the winner-take-all strategy. The authors claim with this setup the network is able to learn complex decision boundaries. = Methods mentioned till now, employ special optimization techniques to learn the parameters since the max and min operations employed by dilation and erosion are not differentiable. So, altogether different strategies have been proposed to overcome this issue. Araújo [5] utilized network architecture similar to morphological neurons with competitive learning to forecast stock markets. The *argmax* operator was replaced with a linear activation function so that the network is able to regress forecasts and the gradient descent could be utilized for training. For morphological neurons with dendritic structure, Zamora and Sossa [25] proposed to replace the *argmax* operator with a softmax function, in order to utilize the gradient descent optimizer. However, more recent methods don't consider this a hindrance. The gradient is computed where it is possible and taken to be 0 at other places.

The more recent works employing morphological neurons take altogether different approaches in using morphological operations. Franchi *et al.* [6] proposed to utilize morphological operations as layers within neural networks. They have shown the pooling layer in CNNs can be replaced with a learned morphological pooling, and using CNNs with only morphological layers works well in denoising images. Nogueira *et al.* [15] proposed to utilize morphological operations to be able to learn novel deep features while training the network end-to-end with gradient descent. The authors' have shown experimentally that these features work well for different image classification tasks. Islam *et al.* [9] proposed using morphological hit-or-miss transform to build networks. Mondal *et al.* [13] introduced the opening-closing network for image de-raining and dehazing using morphological opening and closing operations. Limonova *et al.* [11] proposed new morphological neurons called, bipolar morphological neurons. The authors claim to achieve better recognition results compared to neural networks.

Although morphological neurons have been utilized in various ways for specific applications, their generic theoretical justification is scarce to date. It is still an open question how morphological networks should be designed so that they become a generic tool that can solve any learning problem. In the following subsections, we have tried to answer these questions.

### 3 Morphological Neurons

In this section, we first introduce morphological neurons and their properties. Then we define the Morphological Block and show that it computes a sum of hinge functions. As pointed out in [10], hinge functions are a powerful alternative for classification, regression and function approximation. Although the function approximation capability of a morphological block is not known, we have proved that by using two morphological blocks sequentially, we can approximate any function.

#### 3.1 Dilation and Erosion neurons

Dilation and Erosion neurons are the two most basic morphological neurons because all other morphological operations can be represented as a composition of these two. Note that, we are utilizing the 1D version of these operations to facilitate theoretical analysis. Given an input  $\mathbf{x} \in \mathbb{R}^d$  and a structuring element  $\mathbf{s} \in \mathbb{R}^d$ , the operation of **dilation** ( $\oplus$ ) and **erosion** ( $\ominus$ ) neurons are defined, respectively, as

$$\mathbf{x} \oplus \mathbf{s} = \max_k \{x_k + s_k\}, \quad (1)$$

$$\mathbf{x} \ominus \mathbf{s} = \min_k \{x_k - s_k\}, \quad (2)$$

where  $x_k$  denotes  $k^{\text{th}}$  element of input vector  $\mathbf{x}$ . After computing dilation and erosion we may set a *limiter* or *bias*, say,  $s_{d+1}$  to compute final output from dilation and erosion neurons by  $\max\{\mathbf{x} \oplus \mathbf{s}, s_{d+1}\}$  and  $\min\{\mathbf{x} \ominus \mathbf{s}, -s_{d+1}\}$  respectively. Note that this ensures  $s_{d+1}$  to be the lower bound of the output of the dilation neuron, whereas it is the upper bound for the erosion; hence, the term ‘limiter’. Alternatively, we can write it as follows. Let 0 is appended to the input  $\mathbf{x}$ , i.e.  $\mathbf{x}' = [\mathbf{x}, 0]^T$  and  $s_{d+1}$  is appended to  $\mathbf{s}$ , then

$$\max\{\mathbf{x} \oplus \mathbf{s}, s_{d+1}\} = \max\{\max_k \{x_k + s_k\}, s_{d+1}\} = \max_{k=1, \dots, d+1} \{x'_k + s'_k\} = \mathbf{x}' \oplus \mathbf{s}' \quad (3)$$

Where  $s'_k$  is a element of structuring element  $\mathbf{s}'$ . Similarly we can get  $\mathbf{x}' \ominus \mathbf{s}'$ . It may be argued that  $d+1^{\text{th}}$  component is selected if the input has no effect on the output or the function. In these neurons, the structuring element ( $\mathbf{s}'$ ) is learned in the training phase.

The max and min operators used in the dilation and erosion neurons are only piece-wise differentiable. As a result, only a single element of the structuring element is updated at each iteration. To overcome this problem we propose to use the soft version of max and min [11] to define *soft dilation* and *soft erosion* neurons as follows.

$$\mathbf{x}' \hat{\oplus} \mathbf{s}' = \frac{1}{\beta} \log \left( \sum_{k+1} e^{(x'_k + s'_k)\beta} \right), \quad (4)$$

$$\mathbf{x}' \hat{\ominus} \mathbf{s}' = -\frac{1}{\beta} \log \left( \sum_{k+1} e^{(s'_k - x'_k)\beta} \right), \quad (5)$$

where  $\hat{\oplus}$  and  $\hat{\ominus}$  denote the soft dilation and soft erosion, respectively, and  $\beta$  is the ‘‘hardness’’ of the soft operations. The soft version can be made close to its ‘‘hard’’ counterpart by making  $\beta$  large enough [11]. Henceforth, for notational convenience, we use  $\mathbf{x}$  and  $\mathbf{s}$  to denote input and structuring element respectively for dilation (or erosion) with a limiter. In other words, the dilation and erosion neurons include the limiter.

## 3.2 Gradient of Morphological Neurons

Network build using morphological neurons can be trained using the backpropagation algorithm, provided we are able to find their derivative. The max and min operations of dilation and erosion, respectively, are not differentiable. To be precise, they are not differentiable when the arguments to the max or min operation are equal. However, this rarely occurs in practice. So, we may define the derivative of the dilation and erosion operation in the following way.

$$\frac{\partial z^+}{\partial s_i} = \begin{cases} 1 & \text{if } x_i + s_i \text{ is the max,} \\ 0 & \text{otherwise.} \end{cases} \quad \frac{\partial z^-}{\partial s_i} = \begin{cases} 1 & \text{if } x_i - s_i \text{ is the min,} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

So, the computed gradient is non-zero only in the element for which the maximum (or minimum) is attained. For this reason, the computed loss or error affects only one element of the structuring element for a given sample. As a result, only a single neuron is activated in the network and, consequently, only a single weight ( $s_i$ ) is updated at a time. This may result in slow convergence of the training of the network. In practice, training a large morphological network is very slow. Soft morphological neurons mitigate this issue to some extent.

## 3.3 Equivalence of configurations

The morphological neurons may be arranged in different ways to create a network with the aim of solving a particular task. But not all of these configurations are useful and some of them may be spurious. The following are true for different network architectures.

**Theorem 1.** *If we denote  $D_{m_1}E_{m_2}$  as a layer with  $m_1$  dilation neurons and  $m_2$  erosion neurons and  $L$  as a linear combination layer, the following may be said about their configurations.*

- (i) *The architecture  $D_{m_1}E_0 \rightarrow D_{m_2}E_0 \rightarrow \dots \rightarrow D_{m_t}E_0$  consisting only of dilation layers is equivalent to the architecture  $D_{m_t}E_0$  with a single dilation layer. A similar statement is true if one considers architectures with only purely erosion layers.*
- (ii) *The architecture  $D_1E_1 \rightarrow D_1$  is not equivalent to  $D_1E_0$ . Similarly, it is not equivalent to  $D_0E_1$ , and, consequently, the architectures  $D_1E_1 \rightarrow D_1E_1$  and  $D_1E_1$  are not equivalent.*
- (iii) *The architecture  $D_1E_1 \rightarrow D_1 \rightarrow L$  is not equivalent to  $D_1E_0 \rightarrow L$ .*
- (iv) *The architecture  $D_2E_0 \rightarrow D_0E_2 \rightarrow D_1$  is not equivalent to  $D_2E_0 \rightarrow D_1$ .*

The proof is provided in the supplementary material.

## 3.4 Morphological block

Here we define Morphological Block, which is one configuration that can be utilized as a building block for making more complex networks. A Morphological block consists of a layer with dilation and erosion neurons followed by a linear combination of their outputs (Figure 1). We call the layer of dilation and erosion neurons the **dilation-erosion layer** and the following layer as the **linear combination layer**. Let us consider a morphological block with  $n$  dilation neurons and  $m$  erosion neurons in the dilation-erosion layer followed by  $c$

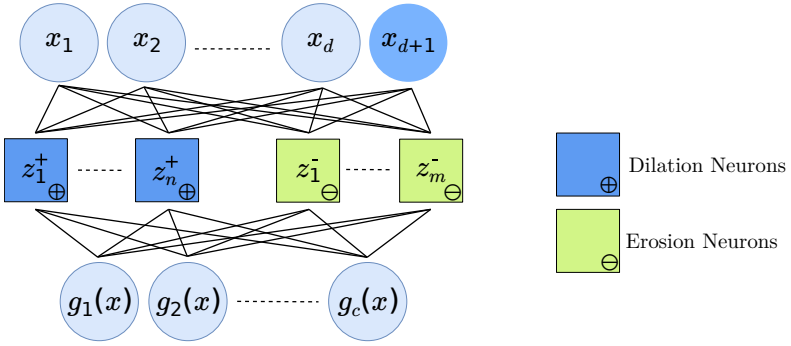


Figure 1: Architecture of single layer morphological block. It contains an input layer, a dilation-erosion layer with  $n$  dilation and  $m$  erosion neuron and a linear combination layer with  $c$  neurons producing the output. The limiter associated with the input  $x_{d+1} = 0$ .

neurons in the linear combination layer. Let  $\mathbf{x} \in \mathbb{R}^d$  be the input to the network and  $z_i^+$  and  $z_j^-$  be the output of the  $i^{\text{th}}$  dilation neuron and the  $j^{\text{th}}$  erosion neuron respectively:

$$z_i^+ = \mathbf{x} \oplus \mathbf{s}_i^+, \quad (7)$$

$$z_j^- = \mathbf{x} \ominus \mathbf{s}_j^-, \quad (8)$$

where  $\mathbf{s}_i^+$  and  $\mathbf{s}_j^-$  are the structuring elements of the respective neurons. Note that  $i \in \{1, 2, \dots, n\}$  and  $j \in \{1, 2, \dots, m\}$ . The final output of a node in the linear combination layer is computed as

$$\mathcal{M}(\mathbf{x}) = \sum_{i=1}^n z_i^+ \omega_i^+ + \sum_{j=1}^m z_j^- \omega_j^-, \quad (9)$$

where  $\omega_i^+$  and  $\omega_j^-$  are the weights of the combination layer. When the network is trained, it learns all  $\mathbf{s}_i^+$ ,  $\mathbf{s}_j^-$ ,  $\omega_i^+$  and  $\omega_j^-$ .

### 3.5 Morphological block as a sum of hinge functions

In this subsection, we show that the simple morphological block can be represented as a sum of hinge functions. Hinge functions provide a powerful representation for problems like classification, and regression tasks [2]. Additionally, we try to develop the intuition behind the morphological block with a toy example.

**Definition 1** ( $k$ -order Hinge Function [23]). A  $k$ -order hinge function  $h^{(k)}(\mathbf{x})$  consists of  $(k+1)$  hyperplanes continuously joined together. It may be defined as

$$h^{(k)}(\mathbf{x}) = \pm \max\{\mathbf{w}_1^T \mathbf{x} + b_1, \mathbf{w}_2^T \mathbf{x} + b_2, \dots, \mathbf{w}_{k+1}^T \mathbf{x} + b_{k+1}\} \quad (10)$$

**Proposition 1.** The function computed by a Morphological Block (denoted by  $\mathcal{M}(\mathbf{x})$ ) with  $n$  dilation and  $m$  erosion neurons followed by their linear combination, is a sum of multi-order hinge functions.

In fact, we can show that

$$\mathcal{M}(\mathbf{x}) = \sum_{i=1}^l \alpha_i h_i^{(d)}(\mathbf{x}), \quad (11)$$

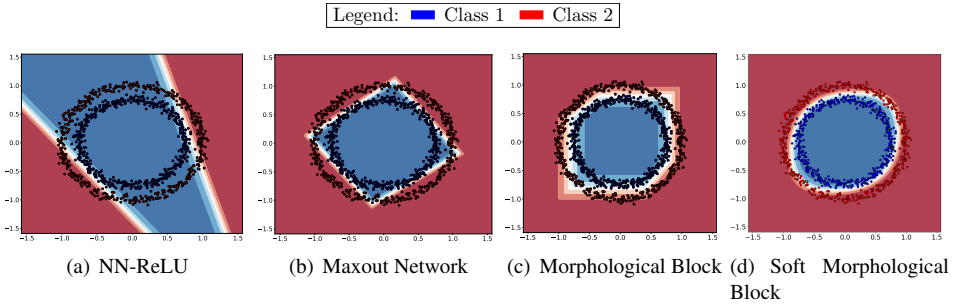


Figure 2: Decision boundaries learned by different networks with two hidden neurons. (a) Baseline neural network is able to learn only two planes. (b) Maxout network is able to learn two more planes with the help of additional parameters. (c) Morphological Block is able to learn more planes with the same number of parameters as NN-ReLU. (d) Using the soft version of the block, smooths the learned decision boundary. This further enhances the discrimination capability of the network while retaining the same number of parameters.

where  $l = m + n$ ,  $\alpha_i \in \{1, -1\}$  and  $h_i^{(d)}(\mathbf{x})$ ,  $1 \leq i \leq l$ , are  $d$ -order hinge functions. The proof is given in the supplementary material.

Since a morphological block computes a sum of hinge functions, it can potentially learn a large number of hyperplanes. It can be seen a morphological block can have maximum  $(d+1)^l - 1$  hyperplanes. Out of all those, there can be almost  $d! \times \binom{l}{d} \times (d+1)^{l-d}$  hyperplanes that are not parallel to any of the axes. (This has been further explained in the supplementary material.) These hyperplanes can act as decision boundaries as has been demonstrated experimentally using a toy dataset representing a two-class problem.

The toy dataset contains samples that are distributed along two concentric circles, one circle for each class. The circles are centered at the origin. We compare the results obtained by various networks with two neurons in the hidden layer. It is observed that the baseline neural network (NN-ReLU) fails to classify this data as with two hidden neurons it learns only two hyperplanes, one for each neuron (Figure 2(a)). The result of maxout network [10] is better, because, in this case, the network learns  $2k = 4$  hyperplanes as shown in figure 2(b). Note that with two morphological neurons in the dilation-erosion layer, our network has learned 6 hyperplanes to form the decision boundary (Figure 2(c)). We should get at most 8 hyperplanes from the morphological block. However, out of these only two decision boundaries are placed in any arbitrary orientation in the 2D space, while others are parallel to either of the axes. Whereas using the soft version of dilation and erosion smooths the boundary, making it aligned with the data (Figure 2(d)).

### 3.6 Two Morphological Blocks: An universal approximator

A single morphological block may be able approximate functions, but we don't know how well is its approximation capability (Supplementary material provides an empirical study on this). However, with two morphological blocks (applied sequentially) any function can be approximated.

**Lemma 1.** Any linear combination of hinge functions  $\sum_{i=1}^m \alpha_i h^{(k_i)}(\mathbf{x})$  can be represented over an arbitrary compact set  $K$  as a two sequential morphological block consisting of dilation neurons only.

Dataset	Test Accuracy		
	Morph-Net	Soft Morph-Net ( $\beta = 8$ )	Similar Network
MNIST	98.39	98.90	<b>99.79</b> [23]
Fashion-MNIST	89.87	<b>89.84</b>	89.70 [24]

Table 1: Accuracy on MNIST and Fashion-MNIST Datasets using a single hidden layer with 400 morphological neurons.

*Proof.* The proof is given in the supplementary material.  $\square$

**Theorem 2** (Universal approximation). *Two morphological blocks applied sequentially, can approximate continuous functions over arbitrary compact sets.*

*Proof.* Continuous functions can be approximated over compact sets by sums of hinge functions (Theorem 3.1 of [2]). Therefore, by Lemma 1, it follows that any continuous function can be approximated over arbitrary compact sets by two-layer Morph-Nets.  $\square$

## 4 Experimental Evaluation

To empirically evaluate the performance of our proposed Morphological block, we have done experiments using various benchmark datasets of several real-world problems. We have compared our results with similarly structured networks because the state-of-the-art methods employ more than just simple neurons to accomplish the results. Also, since our proposed morphological block utilizes 1D morphological operations, the comparison has been done with networks with 1D neurons. So, all the data has been flattened before feeding to the networks. Because of these reasons, we have evaluated our method on MNIST, Fashion-MNIST [24], CIFAR-10, and SVHN dataset only. We have refrained from using large image datasets as a 1D version of the operators won't be able to extract meaningful features from them. Yes, using the 2D version of the morphological operations would indeed be more appropriate for image data, but here our focus is the evaluation of our proposed morphological block, not its 2D version.

### 4.1 MNIST and Fashion-MNIST

For MNIST and Fashion-MNIST [24] dataset, the network we have utilized contains an input layer and a single morphological block. Only a sigmoid activation has been utilised in the last layer, no other activation function has been used. The morphological block contains 200 dilation and 200 erosion neurons. Table 4.1 shows the accuracy of test data after training the network for 300 epochs. We get an average accuracy of 98.43% and 89.84%, respectively, on MNIST and Fashion-MNIST datasets. Note that the reported state-of-the-art techniques make use of different data augmentation and pre-processing techniques to train the data. We have not utilized any of such techniques, but still, we are able to get comparable results.

### 4.2 CIFAR-10 and SVHN

CIFAR-10 [10] and SVHN [14] are two popular classification datasets that are far more challenging than the previous two. All the networks we have utilized here to report the



Architecture	l=200		l=400		l=600	
	CIFAR10	SVHN	CIFAR10	SVHN	CIFAR10	SVHN
NN-tanh	46.6 $\pm$ 0.06	73.9 $\pm$ 0.12	46.9 $\pm$ 0.04	73.9 $\pm$ 0.23	48.0 $\pm$ 0.05	75.6 $\pm$ 0.14
NN-ReLU	47.2 $\pm$ 0.11	64.2 $\pm$ 0.88	48.0 $\pm$ 0.05	76.2 $\pm$ 0.32	48.1 $\pm$ 0.02	<b>79.5 <math>\pm</math> 0.11</b>
Maxout-Network ( $k = 2$ ) [14]	46.9 $\pm$ 0.05	69.4 $\pm$ 0.10	48.0 $\pm$ 0.10	74.1 $\pm$ 0.22	46.4 $\pm$ 0.33	37.8 $\pm$ 3.15
Our	52.0 $\pm$ 0.02	73.4 $\pm$ 0.03	53.6 $\pm$ 0.01	76.9 $\pm$ 0.03	54.0 $\pm$ 0.02	78.2 $\pm$ 0.03
Our (Soft: $\beta = 12$ , 20)	<b>53.5 <math>\pm</math> 0.04</b>	<b>74.1 <math>\pm</math> 0.06</b>	<b>55.8 <math>\pm</math> 0.05</b>	<b>77.0 <math>\pm</math> 0.05</b>	<b>56.9 <math>\pm</math> 0.04</b>	78.5 $\pm$ 0.05

Table 2: Test accuracy achieved on CIFAR-10 and SVHN dataset by different networks when the number of neurons ( $l$ ) in the hidden layer is varied. The value of  $\beta$  is 12 and 20 for CIFAR10 and SVHN respectively.

results, follow a 3-layer architecture: input layer, hidden layer and output layer. For the Maxout network [14], we have taken  $k = 2$  which means each hidden neuron has two extra nodes over which the maximum is computed. For the network with (soft) morphological neurons, sigmoid activation has been utilized only in the last layer. Table 4.2 shows the mean and standard deviation of test accuracy obtained over 5 runs of 300 epochs each and by varying the number of neurons in the hidden layer. It is seen from the table that the Morph-Nets achieve better accuracy for the CIFAR-10 dataset in all cases. However, for the SVHN dataset, its results are comparable with that of other networks. But for both datasets, the accuracy obtained by Morph-Net stays almost the same across training runs. That is not true for other networks.

## 5 Conclusion

In this paper, we have theoretically analysed the morphological neurons and have shown that our proposed morphological block is a good way to arrange morphological neurons. We have also shown, that a morphological block represents a sum of hinge functions and two morphological blocks can approximate any continuous function. This provides the theoretical basis that networks built with morphological neurons are equally capable and it is applicable to a variety of problems. The empirical results also show the applicability of morphological neurons. But there is huge scope for further explorations. Firstly, the networks build with morphological neurons (or morphological blocks) are very slow to train since only a small number of parameters are updated at each iteration (Remember that the gradient is 1 only where the max/min occurs). So, it may also require more number iterations to converge. Improvements in this regard will greatly boost the scope for further exploration. Secondly, to facilitate theoretical analysis, we have restricted ourselves to the 1D version of the morphological operations. This can be easily extended to 2D morphological operations to make CNN-like networks, but to compete with the state-of-the-art networks other advanced layers (e.g. batch norm, drop-out) may need to be adapted for morphological neurons.

## References

- [1] A. Barmpoutis and G. X. Ritter. Orthonormal Basis Lattice Neural Networks. In *2006 IEEE International Conference on Fuzzy Systems*, pages 331–336, July 2006. doi:

10.1109/FUZZY.2006.1681733.

- [2] Leo Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory*, 39(3):999–1013, May 1993. doi: 10.1109/18.256506.
- [3] John D Cook. Basic properties of the soft maximum. *UT MD Anderson Cancer Center Department of Biostatistics Working Paper Series; Working Paper 70*, 2011.
- [4] Jennifer L. Davidson and Frank Hummer. Morphology neural networks: An introduction with applications. *Circuits, Systems and Signal Processing*, 12(2):177–210, June 1993. ISSN 1531-5878. doi: 10.1007/BF01189873.
- [5] Ricardo de A. Araújo. A morphological perceptron with gradient-based learning for Brazilian stock market forecasting. *Neural Networks*, 28:61–81, April 2012. ISSN 0893-6080. doi: 10.1016/j.neunet.2011.12.004.
- [6] Gianni Franchi, Amin Fehri, and Angela Yao. Deep morphological networks. *Pattern Recognition*, 102:107246, 2020.
- [7] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout Networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13*, pages III–1319–III–1327, Atlanta, GA, USA, 2013. JMLR.org.
- [8] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1992. ISBN 0201569434.
- [9] Muhammad Aminul Islam, Bryce Murray, Andrew Buck, Derek T. Anderson, Grant J. Scott, Mihail Popescu, and James Keller. Extending the morphological hit-or-miss transform to deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4826–4838, 2021. doi: 10.1109/TNNLS.2020.3025723.
- [10] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- [11] Elena Limonova, Daniil Matveev, Dmitry Nikolaev, and Vladimir V Arlazarov. Bipolar morphological neural networks: convolution without multiplication. In *Twelfth International Conference on Machine Vision (ICMV 2019)*, volume 11433, page 114333J. International Society for Optics and Photonics, 2020.
- [12] 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*, pages 329–340. Springer, 2013.
- [13] Ranjan Mondal, Moni Shankar Dey, and Bhabatosh Chanda. Image restoration by learning morphological opening-closing network. *Mathematical Morphology-Theory and Applications*, 4(1):87–107, 2020.

- [14] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.
- [15] Keiller Nogueira, Jocelyn Chanussot, Mauro Dalla Mura, and Jefersson A. Dos Santos. An introduction to deep morphological networks. *IEEE Access*, 9:114308–114324, 2021. doi: 10.1109/ACCESS.2021.3104405.
- [16] G. X. Ritter and P. Sussner. An introduction to morphological neural networks. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 4, pages 709–717 vol.4, August 1996. doi: 10.1109/ICPR.1996.547657.
- [17] G. X. Ritter and G. Urcid. Lattice algebra approach to single-neuron computation. *IEEE Transactions on Neural Networks*, 14(2):282–295, March 2003. ISSN 1045-9227. doi: 10.1109/TNN.2003.809427.
- [18] G. X. Ritter, G. Urcid, and V. Juan-Carlos. Two lattice metrics dendritic computing for pattern recognition. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 45–52, July 2014. doi: 10.1109/FUZZ-IEEE.2014.6891551.
- [19] Gerhard X. Ritter and Gonzalo Urcid. Learning in Lattice Neural Networks that Employ Dendritic Computing. In Vassilis G. Kaburlasos and Gerhard X. Ritter, editors, *Computational Intelligence Based on Lattice Theory*, Studies in Computational Intelligence, pages 25–44. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-72687-6.
- [20] P. Sussner. Morphological perceptron learning. In *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell*, pages 477–482, September 1998. doi: 10.1109/ISIC.1998.713708.
- [21] Peter Sussner and Estevão Laureano Esmi. Morphological perceptrons with competitive learning: Lattice-theoretical framework and constructive learning algorithm. *Information Sciences*, 181(10):1929–1950, May 2011. ISSN 0020-0255. doi: 10.1016/j.ins.2010.03.016.
- [22] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066, 2013.
- [23] Shuning Wang and Xusheng Sun. Generalization of hinging hyperplanes. *IEEE Transactions on Information Theory*, 51(12):4425–4431, 2005.
- [24] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [25] Erik Zamora and Humberto Sossa. Dendrite morphological neurons trained by stochastic gradient descent. *Neurocomputing*, 260:420–431, October 2017. ISSN 0925-2312. doi: 10.1016/j.neucom.2017.04.044.