Hybrid Gray-Scale and Fuzzy Morphological/Linear Perceptrons Trained By Extreme Learning Machine

1st Peter Sussner
Dept. of Applied Math, IMECC
University of Campinas
Campinas, SP, Brazil
sussner@ime.unicamp.br

2nd Israel Campiott
NeuralMind
Campinas, SP, Brazil
israelcampiotti@neuralmind.ai

3rd Manuel Alejandro Quispe Torres
Dept. of Applied Math, IMECC
University of Campinas
Campinas, SP, Brazil
manuel.qt@gmail.com

***Abstract*—Morphological perceptrons (MPs) belong to the class of morphological neural networks (MNNs) whose neuronal aggregation functions are drawn from mathematical morphology (MM). Most MNN models including MPs employ operators of gray-scale mathematical morphology as aggregation functions. Recently, a hybrid morphological/linear perceptron (HMLP) appeared in the literature. This neural network model combines the approximation capabilities of the two-layer perceptron having sigmoid activation functions with the capability of the MP to represent non-differentiable functions. For a number of reasons, that include the non-differentiability of morphological operators, it is advantageous to train HMLPs using extreme learning machine (ELM). The fact that gray-scale MM is closely related to fuzzy MM based on Lukasiewicz operators motivated us to introduce hybrid fuzzy morphological/linear perceptrons (FMLPs) and train them using ELM in this paper. We compare the performances of the HMLP and the FMLP with the ones of some related models in a number of well-known classification problems.**

***Index Terms*—Mathematical morphology and complete lattices, erosion and anti-dilation, morphological neural networks, fuzzy sets, hybrid (fuzzy) morphological/linear perceptron, extreme learning machine, classification.**

I. Introduction

Morphological neural networks can be viewed as lattice computing approaches towards computational intelligence since mathematical morphology has its theoretical foundations in lattice theory [1], [2]. The operators of gray-scale MM used in MNN models such as MPs and (gray-scale) morphological associative memories can be expressed in terms of so called additive maximum and additive minimum operations that are defined in a lattice algebra known as minimax algebra, max-plus algebra or tropical linear algebra [3]. Davidson has shown that gray-scale MM following Sternberg's umbra approach [4] can be embedded into minimax algebra [5] and therefore the technical term "morphological neural network" was coined in the 1990s as a name for artificial neural networks that aggregate neuronal inputs using operators of minimax algebra [6]. More generally, an MNN performs an operation of MM at every node before applying an activation function [7]. Although there is no formal definition of a morphological operator [8], MM on complete lattices is equipped with exact definitions of four types of elementary operators, namely erosion, dilation, anti-erosion, and anti-dilation [9].

One of the main factors that led to the development of MNNs are the commonalities of linear and minimax algebra [10] that suggested replacing the linear aggregation functions used in traditional neural network models by certain non-linear operators defined in minimax algebra. Whereas conventional multilayer perceptrons are usually trained using variants of gradient descent [11], a number of constructive training algorithms were proposed for MPs [12] as well as the closely related dendrite or dendritic MNNs [13], [14]. To the best of our knowledge, the first of these constructive training algorithms appeared in [15] and was used for binary classification. Interestingly, the function described by the MP after training via this simple algorithm is given by a supremum of pairwise infima of erosions and anti-dilations from the complete product lattice of extended real numbers to the complete lattice $\{0, 1\}$. This result is reminiscent of the fact that every mapping between complete lattices can be expressed as a supremum of pairwise infima of erosions and anti-dilations [9]. Although the training algorithm of morphological perceptrons with competitive learning (MP/CL) [12] is much more sophisticated than the original MP training algorithm and automatically constructs a modular architecture, it also produces a representation of the aforementioned form. Each module comprises morphological units or components with each one computing a pairwise infimum of an erosion and an anti-dilation. As shown in [16], a morphological component of this form corresponds to a dendrite having an excitatory response in a so called dendritic morphological neural network (DMNN) [13], [17]. Ritter et al. have argued that most of the synapses of a neuron occur in its dendritic tree and it is there where the information is processed [13]. In addition, neurons with dendrites can not only implement a rich repertoire of logical functions but also other lattice-based operations such as the maximum [18], [19], [20]. Recall that this capability of biological neurons served as one of the initial motivations for introducing morphological neural networks [6].

Two of the main advantages of constructive algorithms for MPs [12], [15] and dendrite MNNs [13], [14] are the automatic construction of the network's architecture and the capability of dealing with highly non-differentiable morpho-

This work was supported in part by CNPq under grant nos. 313145/2017-2 as well as FAPESP under grant nos. 2018/13657-1 and 2017/10224-4.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

logical aggregation functions. In Pessoa and Maragos's hybrid morphological/rank/linear neural networks (MRL NNs) [21], problems with the non-differentiability of morphological and rank operations are overcome by smoothing these operations before performing error backpropagation. This idea was also used by Araújo et al. in several of their hybrid models such as the dilation/erosion/linear perceptron (DELP) [22]. Alternatively, morphological and hybrid morphological/linear models were trained using variants of evolutionary computation [23]. Recently, a group of Mexican researcher have trained models of dendritic MNNs and hybrid morphological/linear models without commenting on issues of non-differentiability [17], [24]. It seems reasonable to assume that locations where the partial derivatives do not exist are dealt with by setting the search directions equal to 0.

This year, we introduced another hybrid model called the hybrid morphological/linear perceptron (HMLP), and proposed an ELM approach for training [16]. For convenience, the resulting model was denoted using the acronym HMLP-EL. Undoubtedly, training a network using ELM [25] is computationally inexpensive compared to evolutionary optimization [26] and classical neural network training algorithms and generally leads to a good generalization performance without requiring some form of regularization in order to avoid "overfitting" [27], [28], [29]. In simulations concerning classification, the HMLP-EL exhibited significantly higher classification accuracies than other classifiers including other morphological and hybrid morphological/linear models [16].

The HMLP is a single hidden layer neural network having two layers of weights. The hidden layer comprises conventional semi-linear neurons with sigmoid activation functions as well as morphological components. Recall that, in this model, each morphological component computes an infimum of an erosion and an anti-dilation with values in the complete lattice $\mathbb{R}_{\pm\infty}$, that is commonly used in *gray-scale* MM [2]. This paper introduces hybrid morphological/linear perceptrons whose morphological components compute pairwise infima of erosions and anti-dilations that have values in the complete lattice $[0, 1]$, the usual value set of *fuzzy* MM [30]. Therefore we speak of *fuzzy morphological components*. We refer to the latter model as *fuzzy morphological/linear perceptrons* (FMLPs). The FMLP models can also be trained using ELM. To be more precise, the novelty and the contributions of this paper compared to our previous paper on the HMLP model are as follows:

- We used some identities from general lattice theory in order to derive erosions and anti-dilations that are membership functions of fuzzy sets;
- Each (fuzzy) morphological component of the FMLP model presented in this paper corresponds to the Cartesian product of triangular fuzzy numbers on the universe $\mathbb{R}_{\pm\infty}$;
- In contrast to the morphological components of the HMLP model, the ones of the FMLP model only have a local influence on the computations of the network.

The paper is organized as follows. The next section reviews some relevant concepts of lattice theory and MM. Section 3 recalls the HMLP model and how this model can be trained using ELM. Section 4 proposes a modification of the HMLP-EL approach, namely an FMLP-EL, i.e., an FMLP trained using ELM. Section 5 compares the classifcation accuracies produced by the HMLP-EL, the FMLP-EL, some related models, and some well-known models from the literature in a number of benchmark classification problems.

II. A Few Relevant Concepts of Lattice Theory, Mathematical Morphology and Fuzzy Set Theory

Lattice theory (LT) is a branch of pure and applied mathematics that dates back to the 19th century [31]. In recent years, LT has exerted an ever growing influence on a wide variety of areas of computer science and engineering [32] This influence has provided rigorous mathematical foundations for these areas and has often led to new insights, developments, and applications.

In the late 1990s some researchers have started to incorporate concepts of mathematical morphology (MM) into artificial neural networks. The resulting models were called MNNs [6]. The reasons why several models of MNNs such as morphological perceptrons [12], morphological associative memories [33], morphological/rank/linear neural networks [21], and dilation/erosion perceptrons [22] employ operations of the mathematical theory of minimax algebra are twofold: First of all, classical grayscale MM (umbra approach) can be embedded into minimax algebra [5]. Secondly, minimax algebra grants an easy access to defining real-valued weights in MNNs [34].

Minimax algebra investigates algebras of matrices and vectors with entries in a so called bounded lattice ordered group [3]. For the purposes of this paper, it suffices to consider the extended reals, i.e., $\mathbb{R}_{\pm\infty} = \mathbb{R} \cup \{+\infty, -\infty\}$. Nevertheless, we need to review a few concepts of LT. First of all, recall that a poset is a set $\mathbb{L} \neq \emptyset$ in which a reflexive, antisymmetric, and transitive binary relation "$\leq$" is defined. Hence, a poset is formally given by a pair $(\mathbb{L}, \leq)$. If $\leq$ arises clearly from the context, then we simply refer to the poset $\mathbb{L}$. For $a, b \in \mathbb{L}$, the *closed interval* $[a, b]$ is defined as the set $\{x \in \mathbb{L} \,|\, a \leq x \leq b\}$. An element $l \in \mathbb{L}$ is said to be a *lower bound* of $\mathcal{X} \subseteq \mathbb{L}$ if $l \leq x$ for all $x \in \mathcal{X}$. Similarly, $u \in \mathbb{L}$ is said to be an *upper bound* of $\mathcal{X} \subseteq \mathbb{L}$ if $x \leq u$ for all $x \in \mathcal{X}$. The *infimum* of $\mathcal{X} \subseteq \mathbb{L}$, denoted $\bigwedge \mathcal{X}$, is defined as the greatest lower bound of X. Similarly, the *supremum* of $\mathcal{X} \subseteq \mathbb{L}$, denoted $\bigvee \mathcal{X}$, is defined as the least upper bound of $\mathcal{X}$. If $\mathcal{X} = \{x, y\}$, then one writes $x \wedge y$ and $x \vee y$ instead of $\bigwedge \mathcal{X}$ and $\bigvee \mathcal{X}$, respectively. If $\mathcal{X} = \{x_j : j \in J\}$, then the symbols $\bigwedge_{j \in J} x_j$ and $\bigvee_{j \in J} x_j$ stand respectively for $\bigwedge \mathcal{X}$ and $\bigvee \mathcal{X}$.

If $\mathbb{L}$ is an arbitrary poset, then not every subset $\mathcal{X}$ of $\mathbb{L}$ has a lower bound or an upper bound, let alone an infimum or a supremum. A poset $\mathbb{L}$ is called a *lattice* if for any $x, y \in \mathbb{L}$ we have that $x \wedge y$ and $x \vee y$ exist in $\mathbb{L}$. In particular, a *chain* or *totally ordered set* is a lattice $\mathbb{L}$ in which $\{x \wedge y, x \vee y\} = \{x, y\}$ for all $x, y \in \mathbb{L}$. Equivalently, a chain can be

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

defined as a lattice $\mathbb{L}$ in which all $x, y \in \mathbb{L}$ satisfy $x \leq y$ or $y \leq x$. A lattice $\mathbb{L}$ is *bounded* if $\bigwedge \mathbb{L}$ and $\bigvee \mathbb{L}$ exist in $\mathbb{L}$. If every subset of $\mathbb{L}$ has an infimum and a supremum in $\mathbb{L}$, then the lattice $\mathbb{L}$ is called *complete*. In *gray-scale and fuzzy MM*, images have pixel values in the *complete chains* $\mathbb{R}_{\pm\infty}$ *and* $[0, 1]$, respectively. A complete lattice $\mathbb{L}$ satisfies the infinite distributive laws if for all $x \in \mathbb{L}$ and all $\{y_i \,|\, i \in I\} \subseteq \mathbb{L}$:

$$x \wedge \bigvee_{i\in I} y_i = \bigvee_{i\in I} (x \wedge y_i), \tag{1}$$

$$x \vee \bigwedge_{i\in I} y_i = \bigwedge_{i\in I} (x \vee y_i). \tag{2}$$

Every complete chain such as $\mathbb{R}_{\pm\infty}$ or $[0, 1]$ satisfies the infinite distributive laws since it represents a completely distributive lattice [35]. As an immediate consequence, we have that the direct product of complete chains is also a complete distributive lattice. Recall that, for any poset $\mathbb{L}$, a partial order on $\mathbb{L}^n$, the direct product of n copies of $\mathbb{L}$, is defined by

$$(x_1, \ldots, x_n) \leq (y_1, \ldots, y_n) \Leftrightarrow x_i \leq y_i,\ i = 1, \ldots, n. \tag{3}$$

The poset $\mathbb{L}^n$ can be identified with $\mathbb{L}^X = \{f : X \to \mathbb{L}\}$, where $X = \{1, \ldots, n\}$. In general, if $\mathbb{L}$ is a poset and $X \neq \emptyset$, then one defines a partial order on $\mathbb{L}^X$ as follows:

$$f \leq g \Leftrightarrow f(x) \leq g(x),\ \forall x \in X. \tag{4}$$

Equation 4 implies that $(f \vee g)(x) = f(x) \vee g(x)$ and $(f \wedge g)(x) = f(x) \wedge g(x)$ for all $x \in X$. We use the same symbol to denote the scalar $a \in \mathbb{L}$ and the corresponding constant function. Thus, $(f \vee a)(x) = f(x) \vee a$ and $(f \wedge a)(x) = f(x) \wedge a$.

If $\mathbb{L}$ is a complete lattice then $\mathbb{L}^n$ is a complete lattice as well and in this case we have for any $\{(x_1^j, \ldots, x_n^j) \,|\, j \in J\} \subseteq \mathbb{L}^n$ (here the superscript j merely denotes an index and not an exponent):

$$\bigvee_{j\in J} (x_1^j, \ldots, x_n^j) = (\bigvee_{j\in J} x_1^j, \ldots, \bigvee_{j\in J} x_n^j), \tag{5}$$

$$\bigwedge_{j\in J} (x_1^j, \ldots, x_n^j) = (\bigwedge_{j\in J} x_1^j, \ldots, \bigwedge_{j\in J} x_n^j). \tag{6}$$

As mentioned before, MM on complete lattices has four types of elementary algebraic operators, namely erosion, dilation, anti-erosion, and anti-dilation. Let $\mathbb{L}$ and $\mathbb{M}$ be complete lattices. An (algebraic) erosion is an operator $\varepsilon : \mathbb{L} \to \mathbb{M}$ that satisfies the left side and an (algebraic) dilation is an operator $\delta : \mathbb{L} \to \mathbb{M}$ that satisfies the right side of Equation 7 below (recall that $\varepsilon(\mathcal{X})$ stands for $\{\varepsilon(x) \,|\, x \in \mathcal{X}\}$) for all $\mathcal{X} \subseteq \mathbb{L}$.

$$\varepsilon(\bigwedge \mathcal{X}) = \bigwedge \varepsilon(\mathcal{X}),\ \delta(\bigvee \mathcal{X}) = \bigvee \delta(\mathcal{X}). \tag{7}$$

Similarly, $\bar{\delta}, \bar{\varepsilon} : \mathbb{L} \to \mathbb{M}$ are respectively called an *anti-dilation* and an *anti-erosion* if

$$\bar{\delta}\left(\bigvee \mathcal{X}\right) = \bigwedge \bar{\delta}(\mathcal{X}) \text{ and } \bar{\varepsilon}\left(\bigwedge \mathcal{X}\right) = \bigvee \bar{\varepsilon}(\mathcal{X}) \quad \forall \mathcal{X} \subseteq \mathbb{L}\,. \tag{8}$$

Banon and Barrera's representation theorem for mappings between complete lattices $\mathbb{L}$ and $\mathbb{M}$ [9] implies that for every $\psi : \mathbb{L} \longrightarrow \mathbb{M}$ there exists an index set I and erosions ε^i and anti-dilations $\bar{\delta}^i$ where $i \in I$ such that

$$\psi = \bigvee_{i\in I} (\varepsilon^i \wedge \bar{\delta}^i)\,. \tag{9}$$

In this paper, certain types of erosions and anti-dilations $\mathbb{R}_{\pm\infty}^n \to \mathbb{R}_{\pm\infty}$, where $\mathbb{R}_{\pm\infty}^n$ denotes $(\mathbb{R}_{\pm\infty})^n$, will play an important role. Specifically, we have that for every $\mathbf{w} \in \mathbb{R}_{\pm\infty}^n$, the following operators $\varepsilon_{\mathbf{w}}, \bar{\delta}_{\mathbf{w}} : \mathbb{R}_{\pm\infty}^n \to \mathbb{R}_{\pm\infty}$ represent respectively an erosion and an anti-dilation [12]:

$$\varepsilon_{\mathbf{w}}(\mathbf{x}) = \bigwedge_{i=1}^{n} (x_i + w_i),\ \bar{\delta}_{\mathbf{w}}(\mathbf{x}) = \bigwedge_{i=1}^{n} (x_i^* + w_i). \tag{10}$$

Here, $+\infty + w = +\infty$ and $-\infty + w = -\infty$ for all $w \in \mathbb{R}$ and $+\infty + (-\infty) = -\infty + (-\infty) = +\infty$ (note that the extension of the addition to $\mathbb{R}_{\pm\infty} \times \mathbb{R}_{\pm\infty}$ with these properties was denoted using the symbol "$+'$" in previous publications [3], [12]). Moreover, the symbol x^* denotes the conjugate of x_i given by [3]:

$$x^* = \begin{cases} -x & \text{if } x \in \mathbb{R}, \\ -\infty & \text{if } x = +\infty, \\ +\infty & \text{if } x = -\infty. \end{cases} \tag{11}$$

.

For every $\mathbf{w} \in \mathbb{R}_{\pm\infty}^n$ and every $j \in \{1, \ldots, n\}$, let us employ the notations ε_{w_j} and $\bar{\delta}_{w_j}$ to denote the following special cases of the operators $\varepsilon_{\mathbf{w}}, \bar{\delta}_{\mathbf{w}} : \mathbb{R}_{\pm\infty}^n \to \mathbb{R}_{\pm\infty}$:

$$\varepsilon_{w_j}(\mathbf{x}) = x_j + w_j = x_j + w_j + \bigwedge_{i\neq j} (x_i + \infty), \tag{12}$$

$$\bar{\delta}_{w_j}(\mathbf{x}) = x_j^* + w_j = x_j^* + w_j + \bigwedge_{i\neq j} (x_i^* + \infty). \tag{13}$$

Let us use the following observations in order to deduce erosions and anti-dilations $\mathbb{R}_{\pm\infty}^n \to \mathbb{R}_{\pm\infty}^n$ from $\varepsilon_{\mathbf{w}}$ and $\delta_{\mathbf{w}}$.

Theorem 1: Let c be a positive real number and $a, b \in \mathbb{R}_{\pm\infty}$. If ε and $\bar{\delta}$ are respectively an erosion and an anti-dilation $\mathbb{R}_{\pm\infty}^n \to \mathbb{R}_{\pm\infty}$, then we have

1) $c\varepsilon$, $\varepsilon \vee a$, and $\varepsilon \wedge b$ are also erosions $\mathbb{R}_{\pm\infty}^n \to \mathbb{R}_{\pm\infty}$;
2) $c\bar{\delta}$, $\bar{\delta} \vee a$, and $\bar{\delta} \wedge b$ are also anti-dilations $\mathbb{R}_{\pm\infty}^n \to \mathbb{R}_{\pm\infty}$.

Similar statements hold true for dilations and anti-erosions;

Proof 1: The facts that $c\varepsilon$ and $c\bar{\delta}$ are respectively erosions and anti-dilations for all $c > 0$ follow immediately from the identities $c \cdot (\bigwedge_{i\in I} y_i) = \bigwedge_{i\in I} (cy_i)$ and $c \cdot (\bigvee_{i\in I} y_i) = \bigvee_{i\in I} (cy_i)$. Moreover, for any $\mathcal{X} \subseteq \mathbb{R}_{\pm\infty}^n$, we have

$$(\varepsilon \wedge b)(\bigwedge \mathcal{X}) = \varepsilon(\bigwedge \mathcal{X}) \wedge b = \bigwedge \varepsilon(\mathcal{X}) \wedge b = \bigwedge (\varepsilon \wedge b)(\mathcal{X}), \tag{14}$$

$$(\bar{\delta} \wedge b)(\bigvee \mathcal{X}) = \bar{\delta}(\bigvee \mathcal{X}) \wedge b = \bigwedge \bar{\delta}(\mathcal{X}) \wedge b = \bigwedge (\bar{\delta} \wedge b)(\mathcal{X}). \tag{15}$$

Let us employ the meet infinite distributive law given in Equation 2 to finish the proof of the theorem. For clarity, let us write any $\mathcal{X} \subseteq \mathbb{R}_{\pm\infty}^n$ in the form $\mathcal{X} = \{\mathbf{x}^j \,|\, j \in J\}$. We obtain

$$(\varepsilon \vee a)(\bigwedge \mathcal{X}) = \varepsilon(\bigwedge \mathcal{X}) \vee a = \bigwedge \varepsilon(\mathcal{X}) \vee a \tag{16}$$

$$= \left[\bigwedge_{j\in J} \varepsilon(\mathbf{x}^j)\right] \vee a = \bigwedge_{j\in J} [\varepsilon(\mathbf{x}^j) \vee a] = \bigwedge (\varepsilon \vee a)(\mathcal{X}) \tag{17}$$

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

and

$$(\bar{\delta} \vee a)(\bigvee \mathcal{X}) = \bar{\delta}(\bigvee \mathcal{X}) \vee a = \bigwedge \bar{\delta}(\mathcal{X}) \vee a \quad (18)$$

$$= \left[\bigwedge_{j\in J} \bar{\delta}(\mathbf{x}^j)\right] \vee a = \bigwedge_{j\in J} [\bar{\delta}(\mathbf{x}^j) \vee a] = \bigwedge (\bar{\delta} \vee a)(\mathcal{X}) \quad (19)$$

Corollary 1: If $\varepsilon, \bar{\delta} : \mathbb{R}^n_{\pm\infty} \to \mathbb{R}\pm\infty$ are respectively an erosion and an anti-dilation and $a, b \in \mathbb{R}_{\pm\infty}$ such that $a \leq b$, then the functions $(\varepsilon \vee a) \wedge b$ and $(\bar{\delta} \vee a) \wedge b$ represent also respectively an erosion and an anti-dilation from $\mathbb{R}^n_{\pm\infty}$ to $\mathbb{R}_{\pm\infty}$.

The proof is straightforward.

Corollary 2: Given erosions ε_i and anti-dilations $\bar{\delta}_i : \mathbb{R}^n_{\pm\infty} \to \mathbb{R}_{\pm\infty}$ for $i = 1, \ldots, m$, we have that for every $\mathbf{c} \in \mathbb{R}^n_+$ that the operators $\bigwedge_{i=1}^m c_i\varepsilon_i$ and $\bigwedge_{i=1}^m c_i\bar{\delta}_i$ are respectively also an erosion and an anti-dilation $\mathbb{R}^n_{\pm\infty} \to \mathbb{R}_{\pm\infty}$.

Proof 2: Consider $\mathbf{c} \in \mathbb{R}^m_+$, erosions and anti-dilations $\varepsilon_i, \bar{\delta}_i : \mathbb{R}^n_{\pm\infty} \to \mathbb{R}_{\pm\infty}$, where $i = 1, \ldots, m$. Corollary 1 implies that $c_i\varepsilon_i$ is an erosion for every $i = 1, \ldots, m$ and therefore we obtain the following for any $\mathcal{X} \subseteq \mathbb{R}^n_{\pm\infty}$:

$$\bigwedge_{i=1}^m c_i\varepsilon_i(\bigwedge \mathcal{X}) = \bigwedge_{i=1}^m \bigwedge c_i\varepsilon_i(\mathcal{X}) = \bigwedge \bigwedge_{i=1}^m c_i\varepsilon_i(\mathcal{X}). \quad (20)$$

Thus, $\bigwedge_{i=1}^m c_i\varepsilon_i$ is an erosion. The fact that $\bigwedge_{i=1}^m c_i\bar{\delta}_i$ can be proven in a similar way.

Corollary 2 can be applied in particular to the erosions ε_{w_j} and the anti-dilations $\bar{\delta}_{w_j}$ of Equations 12 and 13, where $\mathbf{w} \in \mathbb{R}^n_{\pm\infty}$ and $j = 1, \ldots, n$. Together with Corollary 1, this observation implies that, for every $\mathbf{c} \in \mathbb{R}^n_+$, $a \leq b \in \mathbb{R}_{\pm\infty}$, the following operators also represent respectively an erosion and an anti-dilation from from $\mathbb{R}^n_{\pm\infty}$ to $\mathbb{R}_{\pm\infty}$:

$$\varepsilon_{\mathbf{w}}^{\mathbf{c},a,b} = [(\bigwedge_{i=1}^n c_i\varepsilon_{w_i}) \vee a] \wedge b, \bar{\delta}_{\mathbf{w}}^{\mathbf{c},a,b} = [(\bigwedge_{i=1}^n c_i\bar{\delta}_{w_i}) \vee a] \wedge b. \quad (21)$$

The meet infinite distributive law given in Equation 2, which is satisfied in a product of complete chains such as $\mathbb{R}^n_{\pm\infty}$, implies that $\varepsilon_{\mathbf{w}}^{\mathbf{c},a,b}$ and $\bar{\delta}_{\mathbf{w}}^{\mathbf{c},a,b}$ can also be written as follows:

$$\varepsilon_{\mathbf{w}}^{\mathbf{c},a,b} = \bigwedge_{i=1}^n [(c_i\varepsilon_{w_i} \vee a) \wedge b], \bar{\delta}_{\mathbf{w}}^{\mathbf{c},a,b} = \bigwedge_{i=1}^n [(c_i\bar{\delta}_{w_i}) \vee a) \wedge b]. \quad (22)$$

Let us focus on the special case where $a = 0$ and $b = 1$. In this case, $\varepsilon_{\mathbf{w}}^{\mathbf{c},a,b}$, $\bar{\delta}_{\mathbf{w}}^{\mathbf{c},a,b}$, and the operators over which the infima are taken have values in the complete chain $[0, 1]$. Hence, we have that

- $\varepsilon_{\mathbf{w}}^{\mathbf{c},0,1}$ and $(c_i\varepsilon_{w_i} \vee 0) \wedge 1$ are erosions $\mathbb{R}^n_{\pm\infty} \to [0, 1]$,
- $\bar{\delta}_{\mathbf{w}}^{\mathbf{c},0,1}$ and $(c_i\bar{\delta}_{w_i} \vee 0) \wedge 1$ are anti-dilations $\mathbb{R}^n_{\pm\infty} \to [0, 1]$.

All of these mappings can also be viewed as *fuzzy sets on the universe* $\mathbb{R}^n_{\pm\infty}$. Recall that a fuzzy set on a universe $U \neq \emptyset$ is given its membership function, which is a function $U \to [0, 1]$. Using this point of view, Equation 22 reveals that one can interpret the fuzzy sets given by $\varepsilon_{\mathbf{w}}^{\mathbf{c},0,1}$ and $\bar{\delta}_{\mathbf{w}}^{\mathbf{c},0,1}$ as the so called *Cartesian products [36] of the fuzzy sets* whose membership functions are given by $(c_i\varepsilon_{w_i} \vee 0) \wedge 1, (c_i\bar{\delta}_{w_i} \vee 0) \wedge 1 : \mathbb{R}_{\pm\infty} \to [0, 1]$, respectively.

III. A Fuzzy Morphological/Linear Perceptron with ELM Based Training: A Modification of the HMLP-EL

Recently, hybrid morphological/linear perceptrons (HMLPs) grew out of MPs [6]. In the late 1990, this model was furnished with a simple learning algorithm that can be applied to binary classification problems [15]. The more recent *morphological perceptron with competitive learning* (MP/CL) training algorithm tackles multi-class classification problems by automatically constructing a modular network architecture [12]. The sth module has weight vectors that are denoted $\mathbf{v}_j^s$ and $\mathbf{w}_j^s$, where $j = 1, \ldots, m_s$, and produces the following output $y_s \in \mathbb{R}$ upon presentation of an input pattern $\mathbf{x} \in \mathbb{R}^n$:

$$y_s = \bigvee_{j=1}^{m_s} (\varepsilon_{\mathbf{v}_j^s}(\mathbf{x}) \wedge \bar{\delta}_{\mathbf{w}_j^s}(\mathbf{x})) \quad (23)$$

Hence, the MP/CL is equipped with $\sum_{s=1,\ldots,S} m_s$ *morphological components* or *units* that compute $\varepsilon_{\mathbf{v}_j^s}(\mathbf{x}) \wedge \bar{\delta}_{\mathbf{w}_j^s}(\mathbf{x})$ upon presentation of an input pattern $\mathbf{x}$. Each of these morphological units determines a closed interval $[\mathbf{a}_j^s, \mathbf{b}_j^s]$ that is given by $[\mathbf{a}_j^s, \mathbf{b}_j^s] = \{\mathbf{x} \in \mathbb{R}^n : 0 \leq \varepsilon_{\mathbf{v}_j^s}(\mathbf{x}) \wedge \bar{\delta}_{\mathbf{w}_j^s}(\mathbf{x})\}$. If the number of classes equals S, then the final output is $\operatorname{argmax}_{s=1,\ldots S} y_s$.

Attractive features of the MP/CL training algorithm include the following:

1) Constructive algorithm that automatically generates the aforementioned morphological components;
2) Fast convergence in a finite number of steps when applied to binary classification problems;
3) Independence of the sequence of the training data.

Moreover, the MP/CL was shown to produce a competitive performance in a number of low-dimensional benchmark classification problems. However, the MP/CL model generalizes poorly in applications to high-dimensional classification problems with sparse training data [16]. In other words, the MP/CL is unable to deal with the curse of dimensionality.

In contrast, training a neural network using extreme learning machine [25] generally yields a good generalization performance, even without the use of some form of regularization to avoid "overfitting" [27]. Recall that ELM can be applied to any network having two layers of computational units and two layers of weights. The computational units in the first layer can be arbitrary and need not be neuron-like [37], while the second layer, i.e., the output layer, comprises linear neurons. The first layer of weights can be randomly chosen while the second layer of weights is adjustable and can be trained using variants of least squares optimization.

These observations reveal that the aforementioned morphological components can play the role of first layer computational units. In an effort to combine the approximation capabilities of the two-layer perceptron having sigmoid activation functions with the capability of the MP to represent non-differentiable functions, we equipped our HMLP model with both morphological components and traditional semi-linear neurons. The resulting architecture is shown in Figure 1. Note

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

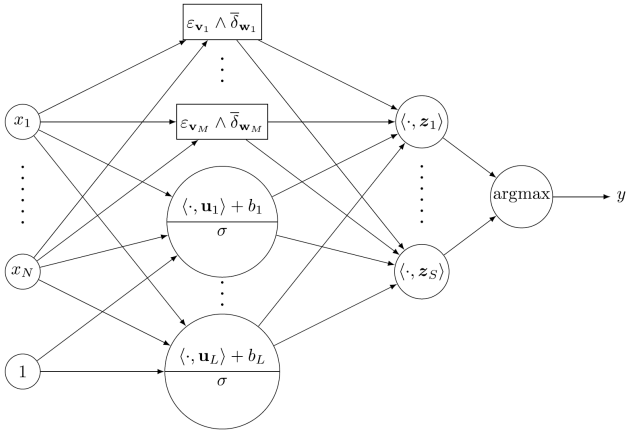

Fig. 1. Topology of an HMLP

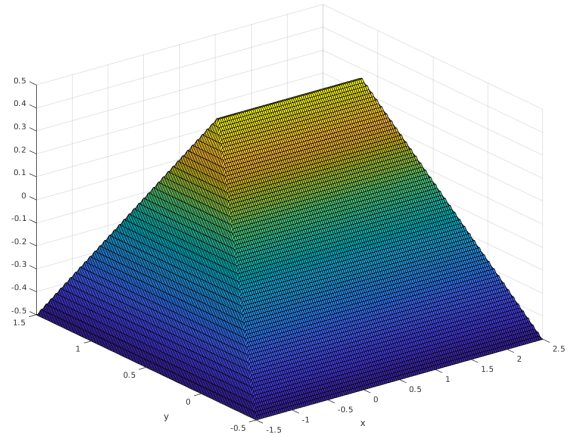

Fig. 2. The morphological unit $\varepsilon_{\mathbf{v}} \wedge \bar{\delta}_{\mathbf{w}}$, where $\mathbf{v} = [1, 0]^T$ and $\mathbf{w} = [2, 1]^T$.

that the hidden layer comprises M morphological components and L linear nodes with a sigmoid activation function σ. There are S output nodes for some $S \in \mathbb{N}$.

Let us briefly review how the HMLP can be trained using ELM. Given a training set $\mathcal{X} = \{(\mathbf{x}_p, \mathbf{t}_p) : p = 1, \ldots, P\} \subset \mathbb{R}^N \times \mathbb{R}^S$ and randomly (with a certain restriction that is explained at the end of this section) chosen first layer weight vectors $\mathbf{v}_m, \mathbf{w}_m, \mathbf{u}_l \in \mathbb{R}^N$ and biases b_l, where $m = 1, \ldots, M$ and $l = 1, \ldots, L$, the goal is to determine the second layer weight vectors $\mathbf{z}_1, \ldots, \mathbf{z}_S \in \mathbb{R}^K$, where $K = M + L$. To this end, consider the following matrices $\mathbf{H} \in \mathbb{R}^{P \times K}$, $\mathbf{Z} \in \mathbb{R}^{K \times S}$, and $\mathbf{T} \in \mathbb{R}^{P \times S}$:

$$\mathbf{H} = \begin{bmatrix} h(\mathbf{x}_1)^{\mathbf{T}} \\ \vdots \\ h(\mathbf{x}_P)^{\mathbf{T}} \end{bmatrix}, \mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 \ldots \mathbf{z}_S \end{bmatrix}, \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_P^T \end{bmatrix}. \tag{24}$$

Here, the vector $h(\mathbf{x}_p)^{\mathbf{T}}$ is given by $[(\varepsilon_{\mathbf{v}_1} \wedge \bar{\delta}_{\mathbf{w}_1})(\mathbf{x}_p), \ldots, (\varepsilon_{\mathbf{v}_M} \wedge \bar{\delta}_{\mathbf{w}_M})(\mathbf{x}_p), \sigma(\langle \mathbf{x}_p, \mathbf{u}_1 \rangle + b_l), \ldots, \sigma(\langle \mathbf{x}_p, \mathbf{u}_L \rangle + b_L)]$, for $p = 1, \ldots, P$. When adopting the original ELM approach, one determines the solution $\mathbf{Z}^* = \mathbf{H}^{\dagger}\mathbf{T}$ to the problem $\mathbf{Z}^*$ of minimizing both $\|\mathbf{HZ} - \mathbf{T}\|_F$ and $\|\mathbf{Z}\|_F$, where $\|\cdot\|_F$ stands for the Frobenius norm. Here, $\mathbf{H}^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{H}$ [38].

In our HMLP-ELM model, we followed the strategy that was proposed in more recent publications on the ELM model [37], [39]. In other words, we improved the stability of the solution and the generalization capability of the resulting model by adding a positive constant $\frac{1}{C}$ to the diagonal of $\mathbf{H}^T\mathbf{H} \in \mathbb{R}^{K \times K}$ or $\mathbf{H}\mathbf{H}^T \in \mathbb{R}^{P \times P}$ [40], [41]. Specifically, if $\mathbf{I}_{K \times K}$ and $\mathbf{I}_{P \times P}$ denote respectively the $K \times K$ and $P \times P$ identity matrices, then the solution $\mathbf{Z}^*$ to the problem of minimizing $\|\mathbf{Z}\|_F + C\|\mathbf{HZ} - \mathbf{T}\|_F$ is given as follows:

1) If $C > 0$ is such that $\frac{1}{C}\ \mathbf{I}_{K \times K} + \mathbf{H}^T\mathbf{H}$ is invertible, then

$$\mathbf{Z}^* = \left(\frac{1}{C} \mathbf{I}_{K \times K} + \mathbf{H}^T\mathbf{H} \right)^{-1} \mathbf{H}^T\mathbf{T}. \tag{25}$$

2) If $C > 0$ is such that $\frac{1}{C}\ \mathbf{I}_{P \times P} + \mathbf{H}^T\mathbf{H}$ is invertible, then

$$\mathbf{Z}^* = \mathbf{H}^T \left(\frac{1}{C} \mathbf{I}_{P \times P} + \mathbf{H}^T\mathbf{H} \right)^{-1} \mathbf{T}. \tag{26}$$

For computational reasons, we chose to determine $\mathbf{Z}^*$ using:

1) Equation 25 if $K < P$;
2) Equation 26 if $P < K$.

Recall that for every $\mathbf{v}, \mathbf{w} \in \mathbb{R}^N$, the set $\{\mathbf{x} \in \mathbb{R}^N \,|\, \varepsilon_{\mathbf{v}}(\mathbf{x}) \wedge \bar{\delta}_{\mathbf{w}}(\mathbf{x})\}$ equals a closed interval of the form $[\mathbf{a}, \mathbf{b}] \subset \mathbb{R}^N$ for some $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$. If $-\mathbf{v} \leq \mathbf{w}$, then we have $\mathbf{a} = -\mathbf{v}$ and $\mathbf{b} = \mathbf{w}$. In ELM-based training for HMLPs, we ensured that $\mathbf{a}_m = -\mathbf{v}_m \leq \mathbf{w}_m = \mathbf{b}_m$ for $m = 1, \ldots, M$ as follows: After selecting arbitrary vectors $\mathbf{c}_m, \mathbf{d}_m \in \mathbb{R}^N$, we generated non-empty closed intervals $[\mathbf{a}_m, \mathbf{b}_m] = [\mathbf{c}_m \wedge \mathbf{d}_m, \mathbf{c}_m \vee \mathbf{d}_m] \subset \mathbb{R}^N$ giving rise to morphological components $\varepsilon_{\mathbf{v}_m} \wedge \bar{\delta}_{\mathbf{w}_m}$ that satisfy $-\mathbf{v}_m \leq \mathbf{w}_m$. For example, Figure 2 provides a visual interpretation of $\varepsilon_{\mathbf{v}} \wedge \bar{\delta}_{\mathbf{w}}$, where $\mathbf{v} = [1, 0]^T$ and $\mathbf{w} = [2, 1]^T$.

The results of Section II reveal that a morphological component consisting of an infimum of an erosion and an anti-dilation is also given by $\varepsilon_{\mathbf{v}}^{\mathbf{c},0,1} \wedge \bar{\delta}_{\mathbf{w}}^{\mathbf{c},0,1} : \mathbb{R}_{\pm\infty}^N \to [0, 1]$ for any $\mathbf{v}, \mathbf{w} \in \mathbb{R}^N$ and any $\mathbf{c} \in \mathbb{R}_+^n$. Since $\varepsilon_{\mathbf{v}}^{\mathbf{c},0,1} \wedge \bar{\delta}_{\mathbf{w}}^{\mathbf{c},0,1}$ can be seen as a fuzzy set on the universe $\mathbb{R}_{\pm\infty}^N$, we speak of a *fuzzy morphological component*. This observation suggests using the FMLP architecture shown in Figure 3 in which the $\varepsilon_{\mathbf{v}_m} \wedge \bar{\delta}_{\mathbf{w}_m}$ hidden units are replaced by $\varepsilon_{\mathbf{v}_m}^{\mathbf{c}_m,0,1} \wedge \bar{\delta}_{\mathbf{w}_m}^{\mathbf{c}_m,0,1}$ hidden units.

When using ELM to train this model, vectors $\mathbf{v}_m$ and $\mathbf{w}_m$ satisfying $-\mathbf{v}_m \leq \mathbf{w}_m$ are generated as above. Then we set

$$(\mathbf{c}_m)_i = \begin{cases} \frac{2}{v_i^m + w_i^m}, & \text{if } -v_i^m < w_i^m, \\ 0 & \text{if } -v_i^m = w_i^m. \end{cases} \tag{27}$$

where the symbols v_i^m and w_i^m denote respectively $(\mathbf{v}_m)_i$ and $(\mathbf{w}_m)_i$. Figure 4 shows the fuzzy morphological unit $\varepsilon_{\mathbf{v}}^{\mathbf{c},0,1} \wedge \bar{\delta}_{\mathbf{w}}^{\mathbf{c},0,1}$, where $\mathbf{v} = [1, 0]^T$, $\mathbf{w} = [2, 1]^T$, and $\mathbf{c} = (\frac{2}{1+2}, \frac{2}{0+1})^T = (\frac{2}{3}, 2)^T$. Note that $\varepsilon_{\mathbf{v}}^{\mathbf{c},0,1} \wedge \bar{\delta}_{\mathbf{w}}^{\mathbf{c},0,1}$ corresponds to the Cartesian product of triangular fuzzy numbers [36].

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

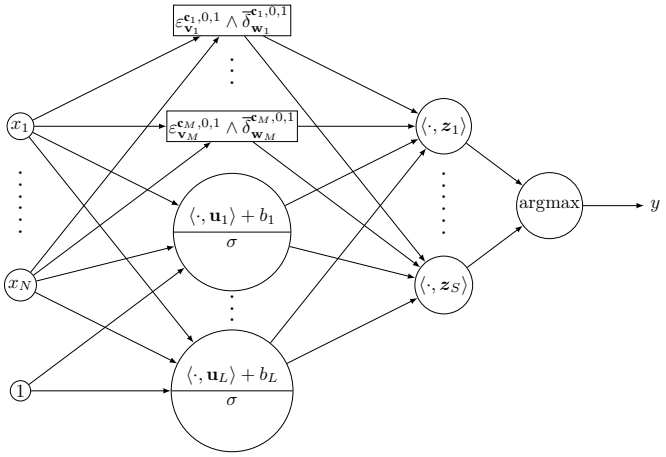

Fig. 3. Topology of an FMLP

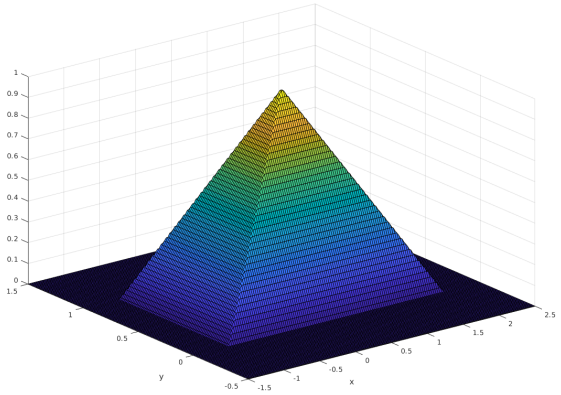

Fig. 4. The fuzzy morphological unit $\varepsilon_{\mathbf{v}}^{\mathbf{c},0,1} \wedge \overline{\delta}_{\mathbf{w}}^{\mathbf{c},0,1}$, where $\mathbf{v} = [1,0]^T$, $\mathbf{w} = [2,1]^T$, and $\mathbf{c} = (\frac{2}{3}, 2)^T$.

IV. Some Experimental Results in Classification

In a recent journal paper [16], we applied the (gray-scale) HMLP-EL to some classification problems given by ten low-dimensional datasets and one high-dimensional dataset, namely the Arrythmia dataset from the UCI repository [42]. We compared the classification accuracy of the HMLP-EL with the ones produced by several morphological and hybrid morphological/linear models as well as two multilayer perceptron (MLP) models with a single hidden layer, one trained using ELM (MLP-EL) [37] and the other one using backpropagation (MLP-BP), and a support vector machine (SVM) [43]. A paired Wilcoxon signed ranks test revealed that the HMLP-EL outperformed all other models except the MP/CL which is unable to deal with sparse, high-dimensional data. Although the MP/CL training algorithm for binary classification problems is guaranteed to converge in less than $2P$ iterations, where P is the number of training patterns, the computational effort of its multi-class version can be prohibitively high due to its one-against-one strategy [12].

Apart from the FMLP-EL. HMLP-EL, and MLP-EL, this section's experimental comparison takes the following recent morphological and hybrid morphological/linear models into account:

1) Dendritic morphological neural network trained by stochastic gradient descent (DMNN-SGD) [17];
2) Dilation/erosion linear perceptron (DELP) [22];
3) Morphological/linear neural network (MLNN) [24].

Instead of the random search strategy employed in [16], we used a grid search in conjunction with 10-fold cross-validation to determine the hyperparameters of the HMLP-EL, FMLP-EL, DMNN-SGD, DELP, MLNN, and MLP-EL. Specifically, we considered the following hyperparameters that are contained in $\mathcal{C} = \{10^e \,|\, e = -5, -4, \ldots, 0\}$, $\mathcal{D} = \{10, 20, \ldots, 100\}$ or $\mathcal{E} = \{100, 200, \ldots, 500, 1000, 2000, \ldots, 5000\}$:

- *HMLP-EL* [16] and *FMLP-EL*: Numbers of classical hidden neurons and hidden morphological units in $\mathcal{E}$ as well as the regularization constant in $\mathcal{C}$;
- *DMNN-SGD [17]:* Initialization method in {HpC, dHpC, D&C, k-means}, and learning rate in $\mathcal{C}$;
- *MLNN [24], [44]:* Number of morphological modules in $\mathcal{D}$ and learning rate in $\mathcal{C}$
- *DELP [22] :* Number of dilation/erosion/linear modules in $\mathcal{D}$;
- *MLP-EL [37]:* Number of classical hidden neurons in $\mathcal{E}$ as well as the regularization constant C in $\mathcal{C}$;

We performed simulations using the following datasets that were drawn from the UCI dataset repository [42]. Note that we generated a balanced subset of the high-dimensional MNIST dataset and considered only red wine in the Wine Quality dataset [45].

Dataset	Train. Size	Test Size	Dimension	Classes
Iris	120	30	4	3
Pages	4450	1023	10	5
MNIST	1000	200	784	10
Wine Quality	1199	400	11	11
Dermatology	292	74	34	6
Ionosphere	280	71	34	2
Cancer	397	172	30	2
Pima	615	153	8	2

TABLE I
Overview of Datasets

For each model and each dataset under consideration, we selected the set of hyperparameters that yielded the best mean validation performance in 10 partitions of the training set into 10 folds. Then we retrained the model using this set of hyperparameters and recorded the classification accuracy for the test set. In the case of the models that were trained using ELM, we also fixed the first layer weights that yielded the highest mean classification rate in the validation phase before determining the second layer weights using the entire training set. If the network has a non-deterministic training algorithm, then we recorded the mean classification accuracy and the standard deviation for the training and testing data in 10 trials.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

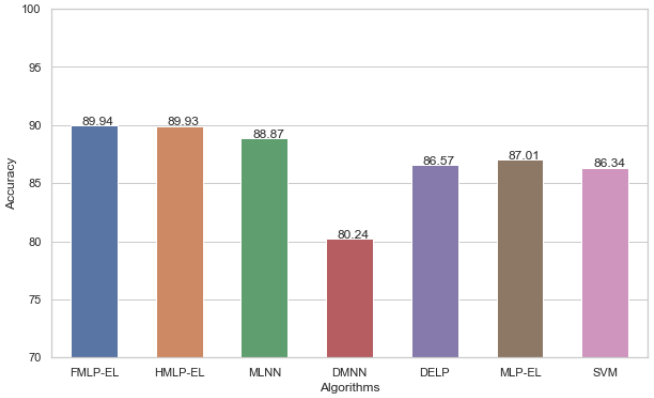

Fig. 5. Mean testing accuracies.

Tables II, III, and IV below list respectively the hyperparameters found in the validation phase and the classification rates in the training and testing phases. For convenience, we included Figure 5 to provide a visual interpretation of Table IV.

Model	Parameters	Pages	MNIST	Wine Quality	Dermatology	Ionosphere	Cancer	Pima
FMLP	Hid. neurons	500	4000	400	200	1000	500	500
	Morph. Units	200	1000	1000	2000	500	1000	200
	C	0.00001	0.1	1	1	0.1	0.001	0.1
HMLP	Hid. neurons	300	5000	500	200	1000	100	200
	Morph. Units	400	5000	5000	300	1000	200	100
	C	0.0001	0.1	0.1	0.001	0.1	0.1	0.01
MLNN	Morph. units	60	100	10	70	40	30	60
	LR	0.02	0.001	0.02	0.02	0.08	0.02	0.01
DMNN-SGD	Method	K-means	HpC	D&C	dHpC	dHpC	dHpC	dHpC
	LR	0.0001	0.001	0.001	0.0001	0.00001	0.00001	0.001
DELP	DEL modules	50	60	100	60	60	80	10
MLP-EL	Hid. neurons	828	830	107	305	368	339	100
	C	0.001	0.001	0.01	0.1	0.001	1	0.1

TABLE II
HYPERPARAMETERS THAT WERE DETERMINED IN THE VALIDATION PHASE.

Dataset	FMLP-EL	HMLP-EL	MLNN	DMNN-SGD	DELP	MLP-EL	SVM
Pages	97.26	98.11	96.8 ±0.04	91,94 ±2.09	97.21 ±0.4	97.30	95.98
MNIST	100	100	100	80.08 ±0.14	95.55 ±0.92	100	97.3
Wine Quality	62.97	95.56	59.99 ±0.6	95.28 ±0.43	67.56 ±2.48	92.83	59.4
Dermatology	99.66	100	99.96 ±0.1	98.66 ±0.08	98.77 ±2.76	100	98.6
Ionosphere	99.64	100	98.78 ±0.17	95.71 ±1.35	99.0 ±0.3	98.14	93.2
Cancer	100	99.24	97.61 ±0.23	95.21 ±0.72	97.48 ±1.96	97.98	98.26
Pima	81.30	82.68	79.82 ±0.36	72.02 ±1.74	80.29 ±0.34	80.62	77.7
Mean	91.57	96.51	90.42 ±0.21	89.84 ±0.87	90.84 ±1.31	95.41	88.63

TABLE III
ACCURACIES ON THE TRAINING SETS

Dataset	FMLP-EL	HMLP-EL	MLNN	DMNN-SGD	DELP	MLP-EL	SVM
Pages	95.80	**97.26**	96.47 ±0.16	89.95	94.9 ±0.8	95.8	94.16
MNIST	93.5	**94.0**	88.29 ±0.9	65.55 ±2.3	84.75 ±2.47	93	88.5
Wine Quality	61.25	**63.0**	61.95 ±0.58	51.62 ±0.58	58.5 ±0.35	57 ±0.4	57.0
Dermatology	**98.65**	**98.65**	97.83 ±0.66	90.95	93.51 ±1.13	97.3	98.6
Ionosphere	**100**	99.59	99.85 ±0.42	94.37	98.31 ±1.1	90.14	93.2
Cancer	**100**	99.42	97.79 ±0.26	95.93	97.56 ±1.56	98.25	97.73
Pima	**80.39**	77.58	79.9 ±0.4	73.33 ±0.57	78.47 ±0.8	77.58	75.2
Mean	**89.94**	89.93	88.87 ±0.48	80.24 ±0.49	86.57 ±1.17	87.01	86.34

TABLE IV
ACCURACIES ON THE TESTING SETS

Table IV and Figure 5 reveal that the FMLP outperformed all other models in terms of mean classification accuracy in the testing phase. The HMLP followed closely behind. Note that for each dataset under consideration, the highest classification rate on the test data was either achieved by the FMLP or the HMLP. The MLNN model exhibited the third best mean classification performance – even better than the one of the MLP-EL in contrast to the experimental results in [16]. Also note that there is a great discrepancy between the best and the worst classification results for the testing data of the high-dimensional MNIST dataset. On the one hand, the three highest classification accuracies of $94, 93.5$, and 93% were achieved by the three ELM-based models we tested, that is, the HMLP-EL, the FMLP-EL, and the MLP-EL. On the other hand, the three gradient-based models MLNN, DELP, and MLNN yielded the lowest classification accuracies. Recall that the strategy of randomly selecting the first layer weights makes ELM-based models less susceptible to overfitting than models with gradient-based learning unless some additional strategies to avoid overfitting are used in conjunction with the latter models. Note that the DMNN-SGD model produced a particularly low classification accuracy of only $65.55 \pm 2.3\%$ for the MNIST test data. Note that in this case the DMNN-SGD was initialized using the first step of the MP/CL training algorithm [12] which is unable to adequately deal with sparsely distibuted high-dimensional data [16].

When performing an unparametric statistical comparison of the FMLP-EL with the other models using the paired Wilcoxon signed-rank test, one notices that there is no significant statistical difference between the results of the FMLP and the MLNN at the default $5\ \%$ significance level, because one obtains a p-value of 0.2969 in this comparison. All other p-values do not exceed 0.05. Specifically, the Wilcoxon signed-rank test yields $p = 0.0313 < 0.05$ when comparing the FMLP and the MLP-EL. The p-value for the pairwise comparison of the FMLP with each of the other models, i.e., the DMNN-SGD, DELP, and SVM, is 0.0156.

Note that the Wilcoxon signed-rank test yields no statistical difference at all between the FMLP and the HMLP. In fact, the corresponding p-value is 1. However, the combined results of Tables III and IV seem to indicate that the HMLP-EL is more susceptible to overfitting than the FMLP-EL.

V. CONCLUDING REMARKS

In this paper, we introduced the fuzzy morphological/linear perceptron as a modification of the hybrid morphological/linear perceptron that recently appeared in the literature. The FMLP is a single hidden layer neural network having both conventional semi-linear nodes and morphological units each of which computes an infimum of an erosion and an anti-dilation from the complete lattice $\mathbb{R}^n_{\pm\infty}$ to the complete chain $[0, 1]$. Each morphological unit of our FMLP model represents a membership function of a Cartesian product of triangular fuzzy numbers [36]. In contrast to a morphological unit of the HMLP model, such a fuzzy morphological unit only has non-zero values in a closed subinterval of $\mathbb{R}^n_{\pm\infty}$ (in practice, $\mathbb{R}^n$) and has therefore only a local effect on the function computed by the network. This may be the reason why the FMLP produced a slightly lower mean classification error in the testing phase than the HMLP although its classification errors in the training phase were generally higher.

Let us also remark that the MLNN and not the MLP-EL came closest to the FMLP-EL and HMLP-EL models in terms

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

of classification performance. We suspect that the grid search employed in this paper is less effective than a random search [46] for determining suitable parameters of the ELM-based models.

REFERENCES

[1] J. Serra, *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. New York: Academic Press, 1988.

[2] C. Ronse, "Why mathematical morphology needs complete lattices," *Signal Processing*, vol. 21, no. 2, pp. 129–154, Oct. 1990.

[3] R. Cuninghame-Green, *Minimax Algebra: Lecture Notes in Economics and Mathematical Systems 166*. New York: Springer-Verlag, 1979.

[4] S. R. Sternberg, "Grayscale morphology," *Computer Vision, Graphics and Image Processing*, vol. 35, pp. 333–355, 1986.

[5] J. L. Davidson, "Foundation and applications of lattice transforms in image processing," in *Advances in Electronics and Electron Physics*, P. Hawkes, Ed. New York, NY: Academic Press, 1992, vol. 84, pp. 61–130.

[6] G. X. Ritter and P. Sussner, "An introduction to morphological neural networks," in *Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austria, 1996, pp. 709–717.

[7] P. Sussner and M. E. Valle, "Fuzzy associative memories and their relationship to mathematical morphology," in *Handbook of Granular Computing*, W. Pedrycz, A. Skowron, and V. Kreinovich, Eds. New York: John Wiley and Sons, Inc., 2008, ch. 33.

[8] H. J. A. M. Heijmans, *Morphological Image Operators*. New York, NY: Academic Press, 1994.

[9] G. J. F. Banon and J. Barrera, "Decomposition of mappings between complete lattices by mathematical morphology, part 1. general lattices," *Signal Processing*, vol. 30, no. 3, pp. 299–327, Feb. 1993.

[10] G. X. Ritter and J. N. Wilson, *Handbook of Computer Vision Algorithms in Image Algebra*, 2nd ed. Boca Raton: CRC Press, 2001.

[11] P. Sussner, "Perceptrons," in *Wiley Encyclopedia of Electrical and Electronics Engineering*, J. G. Webster, Ed. New York: John Wiley and Sons, Inc., 1999, pp. 44–59.

[12] P. Sussner and E. Esmi, "Morphological perceptrons with competitive learning: Lattice-theoretical framework and constructive learning algorithm," *Information Sciences*, vol. 181, no. 10, pp. 1929–1950, 2011.

[13] G. X. Ritter and G. Urcid, "Lattice algebra approach to single-neuron computation," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 282–295, March 2003.

[14] H. Sossa and E. Guevara, "Efficient training for dendrite morphological neural networks," *Neurocomput.*, vol. 131, pp. 132–142, May 2014.

[15] P. Sussner, "Morphological perceptron learning," in *Proceedings of the IEEE International Symposium on Intelligent Control*, Gaithersburg, MD, 1998, pp. 477–482.

[16] P. Sussner and I. Campiotti, "Extreme learning machine for a new hybrid morphological/linear perceptron," *Neural Networks*, vol. 123, pp. 288 – 298, 2020.

[17] E. Zamora and H. Sossa, "Dendrite morphological neurons trained by stochastic gradient descent," *Neurocomputing*, vol. 260, pp. 420 – 431, 2017.

[18] I. Lampl, D. Ferster, T. Poggio, and M. Riesenhuber, "Intracellular measurements of spatial integration and the max operation in complex cells of the cat primary visual cortex," *J. Neurophys*, vol. 92, pp. 2704–2713, 2004.

[19] M. London and M. Häusser, "Dendritic computation," *Annual Review of Neuroscience*, vol. 28, pp. 503–532, 2005.

[20] A. J. Yu, T. A. Poggio, and M. A. Giese, "Biophysiologically plausible implementations of maximum operation," *Neural Computation*, vol. 14, no. 12, pp. 2857–2881, 01 2002.

[21] L. F. C. Pessoa and P. Maragos, "Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition," *Pattern Recognition*, vol. 33, pp. 945–960, 2000.

[22] R. A. Araújo, A. Oliveira, and S. Meira, "A dilation-erosion-linear perceptron for Bovespa index prediction," in *Intelligent Data Engineering and Automated Learning - IDEAL 2012*, ser. Lecture Notes in Computer Science, H. Yin, J. Costa, and G. Barreto, Eds. Springer Berlin Heidelberg, 2012, vol. 7435, pp. 407–415.

[23] R. A. Araújo and T. A. E. Ferreira, "A morphological-rank-linear evolutionary method for stock market prediction," *Information Sciences*, vol. 237, pp. 3 – 17, 2013.

[24] G. Hernández, E. Zamora, and H. Sossa, "Morphological-linear neural network," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2018, pp. 1–6.

[25] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489 – 501, 2006, neural Networks Selected Papers from the 7th Brazilian Symposium on Neural Networks (SBRN '04)7th Brazilian Symposium on Neural Networks.

[26] D. Simon, *Evolutionary Optimization Algorithms*. Hoboken, NJ: John Wiley & Sons, 2013.

[27] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, UK: Oxford University Press, 1995.

[28] ——, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, 2006.

[29] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press, 1996.

[30] P. Sussner and M. E. Valle, "Classification of fuzzy mathematical morphologies based on concepts of inclusion measure and duality," *Journal of Mathematical Imaging and Vision*, vol. 32, no. 2, pp. 139–159, 2008.

[31] G. Birkhoff, *Lattice Theory*, 3rd ed. Providence: American Mathematical Society, 1993.

[32] V. G. Kaburlasos and G. X. Ritter, *Computational Intelligence Based on Lattice Theory*, 1st ed. Springer Publishing Company, Incorporated, 2007.

[33] P. Sussner and M. E. Valle, "Grayscale morphological associative memories," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 559–570, May 2006.

[34] G. X. Ritter and G. Urcid, "A lattice matrix method for hyperspectral image unmixing," *Inf. Sci.*, vol. 181, pp. 1787–1803, 2011.

[35] A. Hopenwasser, "Complete distributivity," *Proceedings of Symposia in Pure Mathematics*, vol. 51, no. 1, pp. 285–305, 1990.

[36] W. Pedrycz and F. Gomide, *Fuzzy Systems Engineering: Towards Human-Centric Computing*. New York: Wiley, IEEE Press, 2007.

[37] G. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, April 2012.

[38] K. Murakami and T. Aibara, "An improvement on the Moore-Penrose generalized inverse associative memory," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC–17, no. 4, pp. 699–707, July/August 1987.

[39] J. Tang, C. Deng, and G. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 809–821, April 2016.

[40] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[41] A. Tikhonov, A. Goncharsky, V. Stepanov, and A. Yagola, *Numerical Methods for the Solution of Ill-Posed Problems*, ser. Mathematics and Its Applications. Springer Netherlands, 2013.

[42] A. Asuncion and D. J. Newman, "UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences," 2007, *http://www. ics. uci. edu/∼mlearn/MLRepository. html*.

[43] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy, "SVM and Kernel Methods Matlab Toolbox," Perception Systèmes et Information, INSA de Rouen, Rouen, France, 2005. [Online]. Available: http://asi. insa-rouen. fr/∼arakotom/toolbox/

[44] G. Hernández, E. Zamora, H. Sossa, G. Téllez, and F. Furlán, "Hybrid neural networks for big data classification," *Neurocomputing*, 2019, available online.

[45] I. Campiotti, "HMLP-EL," https://github.com/israelcamp/hmlp, 2019.

[46] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE