Some Theoretical Aspects and Experimental Results on Feedfoward Morphological Neural Networks

Alexandre Monteiro da Silva and
Peter Sussner

State University of Campinas (UNICAMP), Brazil
{ra007954,sussner}@ime.unicamp.br

1. Introduction

The theory of morphological neural networks and its applications have experienced a steady and consistent growth in the last few years. In this setting, computing the next state of a neuron or performing the next layer computation involves one of the elementary operations of mathematical morpholgy.

In this paper, we present the lattice-theoretic framework for morphological neural networks and we compare the performance of several feedfoward morphological neural networks in a classification problem that was drawn from B. Ripley's database [1].

2. Lattice Background for Morphological Neural Networks

Morphological neurons calculate either an erosion, a dilation, an anti-erosion, or an anti-dilation [2] from a complete lattice $\mathbb{L}$ to a complete lattice $\mathbb{M}$. The extended real numbers $\bar{\mathbb{R}}$ and the unit interval $[0,1]$ represent specific examples of complete lattices. Recall that an operator $\varepsilon : \mathbb{L} \to \mathbb{M}$ represents an erosion if and only if $\varepsilon(\bigwedge Y) = \bigwedge_{y \in Y} \varepsilon(y), \quad \forall Y \subseteq \mathbb{L}$. Dilation, anti-erosion and anti-dilation are defined in a similar way.

Banon and Barrera [3] showed that every mapping $\psi : \mathbb{L} \longrightarrow \mathbb{M}$ can be written as a supremum of infimums of pairs of *erosions* and *anti-dilations*. In other words, there exist erosions ε^i, anti-dilations $\bar{\delta}^i$, and an index set I such that

$$\psi = \bigvee_{i \in I} (\varepsilon^i \wedge \bar{\delta}^i)\,. \tag{1}$$

Similarly, the mapping ψ can be written as an infimum of supremums of pairs of *dilations* and *anti-erosions*. In the special case that ψ is increasing, ψ can be represented as a supremum of erosions or as an infimum of dilations.

3. Some Types of Feedfoward Morphological Neural Networks

- **Morphological Perceptrons (MP)**

  Given a vector of inputs $\mathbf{x} \in \bar{\mathbb{R}}^n$, a vector of synaptic weights $\mathbf{w} \in \bar{\mathbb{R}}^{1\times n}$ and an activation function f, a neuron of the morphological perceptron calculates a output y according to one of the following rules:

  $$\begin{aligned}
  &y = f(\varepsilon_{\mathbf{w}}(\mathbf{x})), \quad \text{where} \quad \varepsilon_{\mathbf{w}}(\mathbf{x}) = \wedge_{i=1}^{n}(x_i + w_i);\\
  &y = f(\delta_{\mathbf{w}}(\mathbf{x})), \quad \text{where} \quad \delta_{\mathbf{w}}(\mathbf{x}) = \vee_{i=1}^{n}(x_i + w_i);\\
  &y = f(\bar{\varepsilon}_{\mathbf{w}}(\mathbf{x})), \quad \text{where} \quad \bar{\varepsilon}_{\mathbf{w}}(\mathbf{x}) = \vee_{i=1}^{n}(-x_i + w_i);\\
  &y = f(\bar{\delta}_{\mathbf{w}}(\mathbf{x})), \quad \text{where} \quad \bar{\delta}_{\mathbf{w}}(\mathbf{x}) = \wedge_{i=1}^{n}(-x_i + w_i).
  \end{aligned}$$

- **Morphological Perceptrons with dendrites (MPD)**

  Ritter and Urcid [5] developped a new paradigm for computing with morphological neurons, where the process is peformed in the dendrite.

  The output of a MPD is defined by the following equation:

  $$y(\mathbf{x}) = f(p_k \bigwedge_{i=1}^{n} \bigwedge_{l \in L} (-1)^{l+1}(x_i + w_{ki}^l)), \tag{2}$$

  where f is a hard limiter function, $L = \{0,1\}$ and $p_k = \{-1,1\}$ denotes the excitatory or inhibitory response of the kth dentrite.

- **Fuzzy Lattice Neural Network (FLNN)**

  Let $\mathbb{L}$ be a complete lattice. Given a vector of inputs $\mathbf{x} \in \mathbb{L}$ and a vector of synaptic weights, a neuron of an FLNN [6] computes $p(\mathbf{x}, \mathbf{w})$, where the function $p : \mathbb{L} \times \mathbb{L} \to [0,1]$ is such that $p(\mathbf{x}, \mathbf{y}) = 1 \Leftrightarrow \mathbf{x} \leq \mathbf{y}$. Many times, $\mathbb{L}$ is given by the complete lattice of the so called *generalized intervals*. This complete lattice is generated by the family of the hyperboxes with vertices in $\bar{\mathbb{R}}$ or in $[0,1]$. For a fixed $\mathbf{w} \in \mathbb{L}$, the function $p(., \mathbf{w}) : \mathbb{L} \to [0,1]$ represents both an anti-dilatation and an anti-erosion.

- **Hybrid morphological/rank/ linear neural network (MRL-NNs)**

  MRL-NNs [7] employ a linear combination of a linear and a rank/morphological agregation function in each node followed by the

application of an activation function. The rank/morphological agregation function generalizes the morphological operators $\delta_{\mathbf{w}}$ and $\varepsilon_{\mathbf{w}}$. Given the list $\mathbf{t} = (x_1 + w_1, \ldots, x_n + w_n)$, we sort its components in decreasing order which yields $t_{(1)} \geqslant t_{(2)} \geqslant \ldots \geqslant t_{(n)}$ and pick the rth element of the sorted list. In this way, we define the rth rank function of $\mathbf{t}$ by

$$\mathcal{R}_r(\mathbf{t}) \equiv t_{(r)}, \quad r = 1, \ldots, n \tag{3}$$

During the training phase, the derivatives of rank functions are estimated according to Pessoa and Maragos's methodology [7] that use rank indicator vectors [7] which marks the locations in $\mathbf{t}$ where $z = \mathcal{R}_r(\mathbf{t})$ occurs.

- **Morphological Modular Neural Networks (MMNNs)**

  MMNNss were introduced by R. Sousa et al. [8]. The arquitecture of an MMNN is based on one of the decompostions of Barrera and Banon that we mentioned in Section 2. MMNN training [8] can be achieved by a simple genetic algorithm (SGA) or by a modified backpropagation (BP) algorithm [7].

4. Experiments Results

We applied our feedfoward morphological models to the problem based on diabetes diagnosis in a population of Indians Pima [1]. The training set contains a randomly selected set of $n = 200$ patients each with 8 numeric variables, representing biological characteristics, and the validation and test sets contain a random selected set of 166 patients each.

Training an MP using Sussner's supervised algorithm automatically generated 52 and 26 neurons on the first and the second hidden layer respectively [4]. In a similar vein, training an MPD using the Ritter's constructive algorithm [5] yielded 26 dendrites. We also considered an MRL-NN with one hidden layer consisting of 5 neurons and applied a modified backpropagation (BP) algorithm [7], using a step size $\mu = 0.01$ and a smoothing parameter $\sigma = 0.05$. Apart from training the MMNN in a similar way, we also used a simple genetic algorithm (SGA), considering a initial population of 50 elements (*synaptic weights*), maximum of 100 generations, crossover weight $w = 0.9$ and mutation probability of 0.1. The FLNN generated 72 neurons during the training phase. Finally, we compared the morphological models with an MLP with ten hidden nodes that was trained using the *gradient descent with momentum and adaptive step backpropagation* rule (learning rate $\eta = 10^{-4}$, increase and decrease parameters 1.05 and 0.5 respectively, momentum factor $\alpha = 0.9$).

The following table reveals that the FLNN outperformed all the other models in this experiment. The MLP and the MP exhibit similar percentages of misclassified patterns, whereas the MRL and the MMNN exhibit the worst performances among the models we tested.

Table 1. Percentage of misclassified patterns for training, validation, and testing.

Model	Train. (%)	Valid. (%)	Test. (%)
MP	0	23.8	21.2
MPD	0	27.7	25.4
MRL	31.2	35.5	34.7
MMNN(BP)	31.9	39.7	37.5
MMNN(SGA)	23.1	37.4	30.7
FLNN	0	16.2	14.4
MLP	22.4	23	20.18

We applied these models in conjunction with early stopping except for the MP and FLNN models which we trained until a error rate of 0% for the training set was achieved.

References

[1] C. L. Blake, S. Hettich, D. J. Newman, and C. J. Merz, *UCI Repository of machine learning databases (http://www.ics.uci.edu/∼mlearn/MLRepository.html).* University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

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

[3] G. J. F. Banon and J. Barrera, *Decompositions of mappings between complete lattices by mathematical morphology*, Signal Processing **30** (1993), no. 3, 229–327.

[4] P. Sussner, *Morphological Perceptron Learning*, Proceedings of IEE ISIC/CIRA/ISAS Joint Conference, 1998 Sept., pp. 447–482.

[5] G. X. Ritter and G. Urcid, *Lattice Algebra Approach to Single-Neuron Computation*, IEE Transactions on Neural Networks **14** (2003), no. 2, 282–295.

[6] V. G. Kaburlasos and V. Petridis, *Fuzzy lattice neurocomputing (FLN) models*, Neural Networks **13** (2003 Mar.), 1145–1170.

[7] 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 **33** (2000), 945–960.

[8] R. A. Araújo, F. Madeiro, R. P. Sousa, and L. F. C. Pessoa, *Modular morphological neural network training via adaptive genetic algorithm for design translation invariant operators*, In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (2006).