A brief review of morphological neural networks

Samy Blusseau, CMM

12 décembre 2018

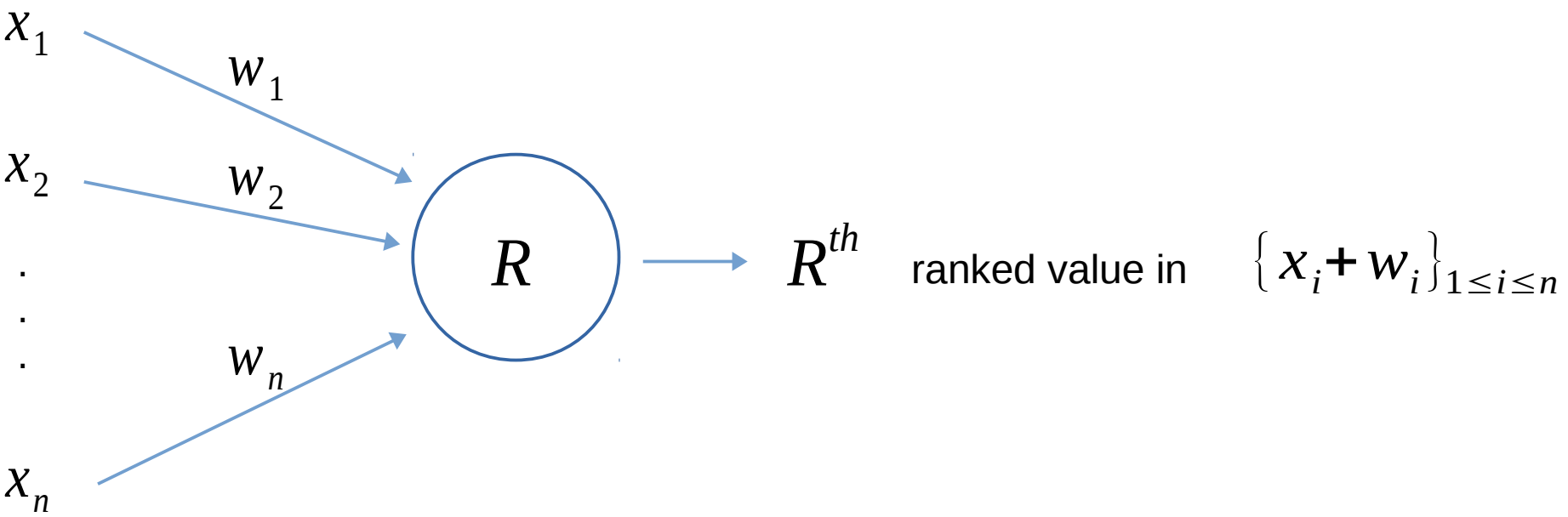
Stephen S. Wilson, *Morphological networks*, 1989

Probably the first paper on the topic

Stephen S. Wilson, *Morphological networks*, 1989

Weighted rank order filters

The weighted rank order filter will be defined informally as follows. The reference point of the structuring element is placed at an image pixel. That pixel value is changed by (a) adding the weights of the structuring elements to the corresponding pixels that the structuring element contacts, (b) ordering the resulting sums, and (c) choosing the Rth element in the ordered list as the output. The same notation as election will be used where the symbol Ξ implies a three dimensional structuring element.

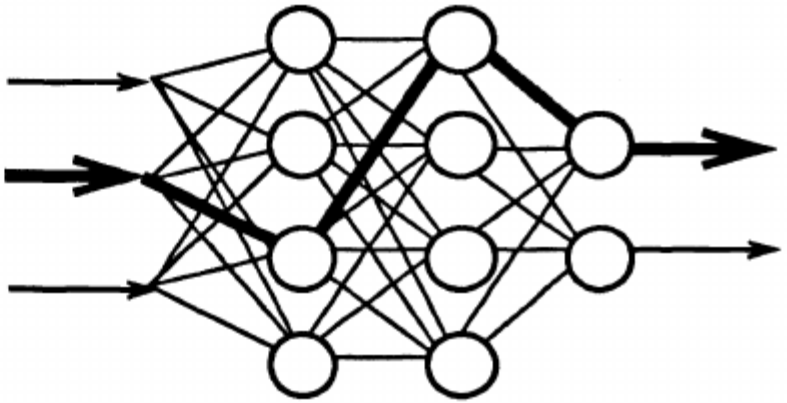


Stephen S. Wilson, *Morphological networks*, 1989

Learning in single layer weighted rank networks.

Figure 11.

Example of a rank trace
in a multi-layer network.



J.L. Davidson & G. X. Ritter,
A theory of morphological neural networks, 1990

Neural networks without learning

J.L. Davidson & G. X. Ritter, *A theory of morphological neural networks*, 1990

1. Introduction. The recent resurgence of interest in artificial neural networks has brought a deluge of publications to the field [4,9,15]. Applications of neural networks are appearing in a wider variety of fields each year. Recently, neural networks have been tied to a mathematical structure known as *image algebra* [14]. Image algebra was developed specifically for the concise expression and clear representation of image processing techniques and to provide a mathematical environment for image processing algorithm development, comparison, and optimization [5,12,13]. It has been shown that a subalgebra of the image algebra includes the mathematical formulations of currently popular neural network models [14], and image algebra expressions have been derived that represent some well-known algorithms designed for neural network computations. The neural network algorithms represented by these expressions look like their textbook formulations. These image algebra expressions are extremely simple and translucent, and the number of expressions representing each algorithm is very small. Furthermore, the image algebra has suggested a more general concept of neural computation than those that are currently used. In this paper we present a theory for a neural network that uses morphological operations. Several specific applications are given. In particular, we discuss networks that compute the morphological operations of opening and closing. We also give an example of a net that can perform the morphological operation of a sieve. A sieve is a morphological filter which filters out objects which are larger than a specified size [16].

We remark that an entirely different approach to a morphological network was presented in [17]. However, this particular model uses the usual operation of multiplication and summation at each node, a fundamental difference from the model presented here, which uses the operation of addition and maximum at each node.

J.L. Davidson & G. X. Ritter, *A theory of morphological neural networks*, 1990

1. Introduction. The recent resurgence of interest in artificial neural networks has brought a deluge of publications to the field [4,9,15]. Applications of neural networks are appearing in a wider variety of fields each year. Recently, neural networks have been tied to a mathematical structure known as *image algebra* [14]. Image algebra was developed specifically for the concise expression and clear representation of image processing techniques and to provide a mathematical environment for image processing algorithm development, comparison, and optimization [5,12,13]. It has been shown that a subalgebra of the image algebra includes the mathematical formulations of currently popular neural network models [14], and image algebra expressions have been derived that represent some well-known algorithms designed for neural network computations. The neural network algorithms represented by these expressions look like their textbook formulations. These image algebra expressions are extremely simple and translucent, and the number of expressions representing each algorithm is very small. Furthermore, the image algebra has suggested a more general concept of neural computation than those that are currently used. In this paper we present a theory for a neural network that uses morphological operations. Several specific applications are given. In particular, we discuss networks that compute the morphological operations of opening and closing. We also give an example of a net that can perform the morphological operation of a sieve. A sieve is a morphological filter which filters out objects which are larger than a specified size [16].

We remark that an entirely different approach to a morphological network was presented in [17]. However, this particular model uses the usual operation of multiplication and summation at each node, a fundamental difference from the model presented here, which uses the operation of addition and maximum at each node.

J.L. Davidson & G. X. Ritter, *A theory of morphological neural networks*, 1990

1. Introduction. The recent resurgence of interest in artificial neural networks has brought a deluge of publications to the field [4,9,15]. Applications of neural networks are appearing in a wider variety of fields each year. Recently, neural networks have been tied to a mathematical structure known as *image algebra* [14]. Image algebra was developed specifically for the concise expression and clear representation of image processing techniques and to provide a mathematical environment for image processing algorithm development, comparison, and optimization [5,12,13]. It has been shown that a subalgebra of the image algebra includes the mathematical formulations of currently popular neural network models [14], and image algebra expressions have been derived that represent some well-known algorithms designed for neural network computations. The neural network algorithms represented by these expressions look like their textbook formulations. These image algebra expressions are extremely simple and translucent, and the number of expressions representing each algorithm is very small. Furthermore, the image algebra has suggested a more general concept of neural computation than those that are currently used. In this paper we present a theory for a neural network that uses morphological operations. Several specific applications are given. In particular, we discuss networks that compute the morphological operations of opening and closing. We also give an example of a net that can perform the morphological operation of a sieve. A sieve is a morphological filter which filters out objects which are larger than a specified size [16].

We remark that an entirely different approach to a morphological network was presented in [17]. However, this particular model uses the usual operation of multiplication and summation at each node, a fundamental difference from the model presented here, which uses the operation of addition and maximum at each node.

17. S.S. Wilson, "Morphological Networks," in *Proc. of the 1989 SPIE Visual Comm. and Image Proc. IV*, Phila., PA (Nov., 1989).

J.L. Davidson & G. X. Ritter, *A theory of morphological neural networks*, 1990

$$\tau = \mathbf{a} \oplus \mathbf{t}$$

$$\mathbf{b} = f(\tau - \theta)$$

$$\mathbf{a} \oplus \mathbf{t} \equiv \{(\mathbf{y}, \mathbf{c}(\mathbf{y})) \colon \mathbf{c}(\mathbf{y}) = \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) \cdot \mathbf{t}_{\mathbf{y}}(\mathbf{x}),\ \mathbf{y} \in \mathbf{Y}\}$$

$$\tau = \mathbf{a} \boxtimes \mathbf{t}$$

$$\mathbf{b} = f(\tau - \theta)$$

$$\mathbf{a} \boxtimes \mathbf{t} \equiv \{(\mathbf{y}, \mathbf{c}(\mathbf{y})) \colon \mathbf{c}(\mathbf{y}) = \bigvee_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) + \mathbf{t}_{\mathbf{y}}(\mathbf{x}),\ \mathbf{y} \in \mathbf{Y}\}$$

While in classical network models, the initial computation of τ (Equation II) is a linear process, the computation of τ in the model described by Equations VI and VIII is nonlinear. If the neural network model can be expressed using Equations VI and VII as the underlying equations of computation, we call it a *morphological neural network*. The remainder of this paper presents several specific morphological neural networks that can calculate openings, closings, and the boolean sieving filters as described by Serra [16].

J.L. Davidson & G. X. Ritter, *A theory of morphological neural networks*, 1990

$$\tau = \mathbf{a} \oplus \mathbf{t}$$

$$\mathbf{b} = f(\tau - \theta)$$

$$\mathbf{a} \oplus \mathbf{t} \equiv \{(\mathbf{y}, \mathbf{c}(\mathbf{y})) \colon \mathbf{c}(\mathbf{y}) = \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) \cdot \mathbf{t}_{\mathbf{y}}(\mathbf{x}),\ \mathbf{y} \in \mathbf{Y}\}$$

$$\tau = \mathbf{a} \boxtimes \mathbf{t}$$

$$\mathbf{b} = f(\tau - \theta)$$

$$\mathbf{a} \boxtimes \mathbf{t} \equiv \{(\mathbf{y}, \mathbf{c}(\mathbf{y})) \colon \mathbf{c}(\mathbf{y}) = \bigvee_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) + \mathbf{t}_{\mathbf{y}}(\mathbf{x}),\ \mathbf{y} \in \mathbf{Y}\}$$

While in classical network models, the initial computation of τ (Equation II) is a linear process, the computation of τ in the model described by Equations VI and VIII is nonlinear. If the neural network model can be expressed using Equations VI and VII as the underlying equations of computation, we call it a *morphological neural network*. The remainder of this paper presents several specific morphological neural networks that can calculate openings, closings, and the boolean sieving filters as described by Serra [16].

J.L. Davidson & G. X. Ritter, *A theory of morphological neural networks*, 1990

$$\tau = \mathbf{a} \oplus \mathbf{t}$$

$$\mathbf{b} = f(\tau - \theta)$$

$$\mathbf{a} \oplus \mathbf{t} \equiv \{(\mathbf{y}, \mathbf{c}(\mathbf{y})) : \mathbf{c}(\mathbf{y}) = \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) \cdot \mathbf{t}_{\mathbf{y}}(\mathbf{x}),\ \mathbf{y} \in \mathbf{Y}\}$$

$$\tau = \mathbf{a} \boxtimes \mathbf{t}$$

$$\mathbf{b} = f(\tau - \theta)$$

$$\mathbf{a} \boxtimes \mathbf{t} \equiv \{(\mathbf{y}, \mathbf{c}(\mathbf{y})) : \mathbf{c}(\mathbf{y}) = \bigvee_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) + \mathbf{t}_{\mathbf{y}}(\mathbf{x}),\ \mathbf{y} \in \mathbf{Y}\}$$

While in classical network models, the initial computation of τ (Equation II) is a linear process, the computation of τ in the model described by Equations VI and VIII is nonlinear. If the neural network model can be expressed using Equations VI and VII as the underlying equations of computation, we call it a *morphological neural network*. The remainder of this paper presents several specific morphological neural networks that can calculate openings, closings, and the boolean sieving filters as described by Serra [16].

J.L. Davidson & F. Hummer,
Morphology neural networks: An introduction with applications, 1993

Learning dilations on images

J.L. Davidson & F. Hummer, *Morphology neural networks: An introduction with applications*, 1993

Example 4. The Boolean Dilation Net with Learning.

The set of training data for this network consists of $\mathbf{P}$ pairs of images, $(\mathbf{a}^k, \mathbf{d}^k)$, $k = 0, \ldots, \mathbf{P} - 1$, where $\mathbf{a}^k$ represents the input image and $\mathbf{d}^k = \mathbf{a}^k \boxtimes \mathbf{t}$ is the input image dilated with an ideal (invariant) template $\mathbf{t}$, which is the same for all images. In practical cases, the template $\mathbf{t}$ is assumed to be unknown. The

Example 6. Grayscale Dilation with Learning for Variant Templates. This network is a generalization of nets in Examples 4 and 5, and is a more general form of the network in [5]. The additive maximum transform that this net attempts to learn can be any arbitrary grayscale one, including a variant transform. For a given set of P training data pairs, the learning rule needs about three passes through the data set to guarantee perfect recall of the P training images.

J.L. Davidson & F. Hummer, *Morphology neural networks: An introduction with applications*, 1993

Example 4. The Boolean Dilation Net with Learning.

case	a_i^k	d_j^k	c_j^k	new value for w_{ji} is
1	0	0	1	current value
2	0	1	0	current value
3	1 or 0	1	1	current value
4	1 or 0	0	0	current value
5	1	0	1	$-\infty$
6	1	1	0	0

Figure 5. Table of change-of-weight rules for boolean dilation network.

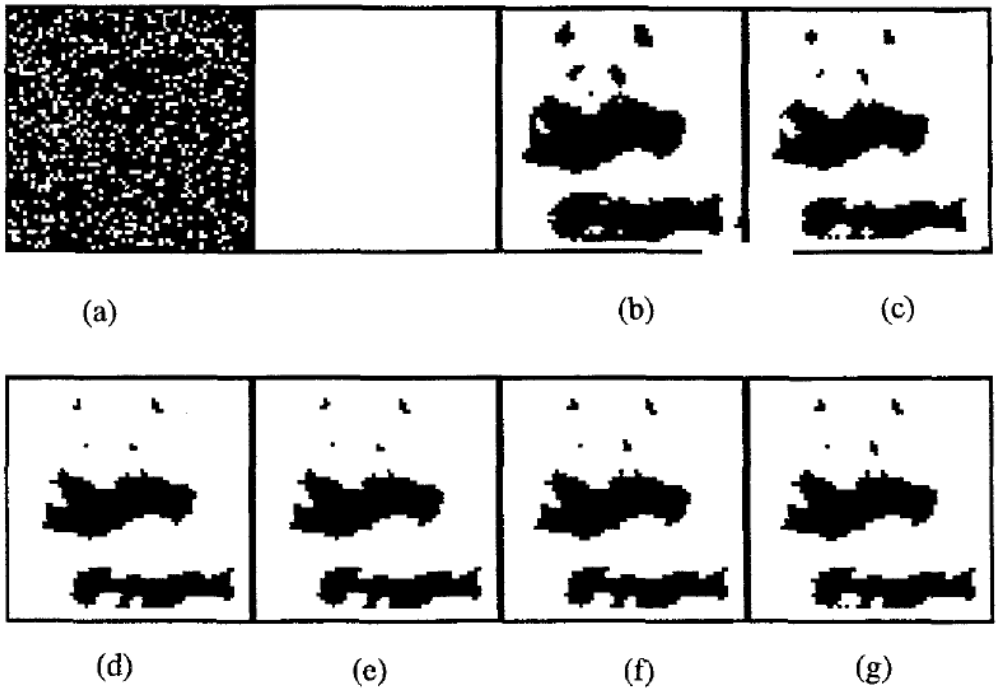


Figure 6. (a) Sample input training data; (b) sample natural scene (panda); (c) image in (b) dilated with von Neumann template; (d) output of net after X = 5 training images applied, for input of (b); (e) output of net after X = 10, for input of (b); (f) output of net after X = 15, for input of (b); (g) output of net after X = 20, for input of (b).

J.L. Davidson & F. Hummer, *Morphology neural networks: An introduction with applications*, 1993

Example 6. Grayscale Dilation with Learning for Variant Templates.

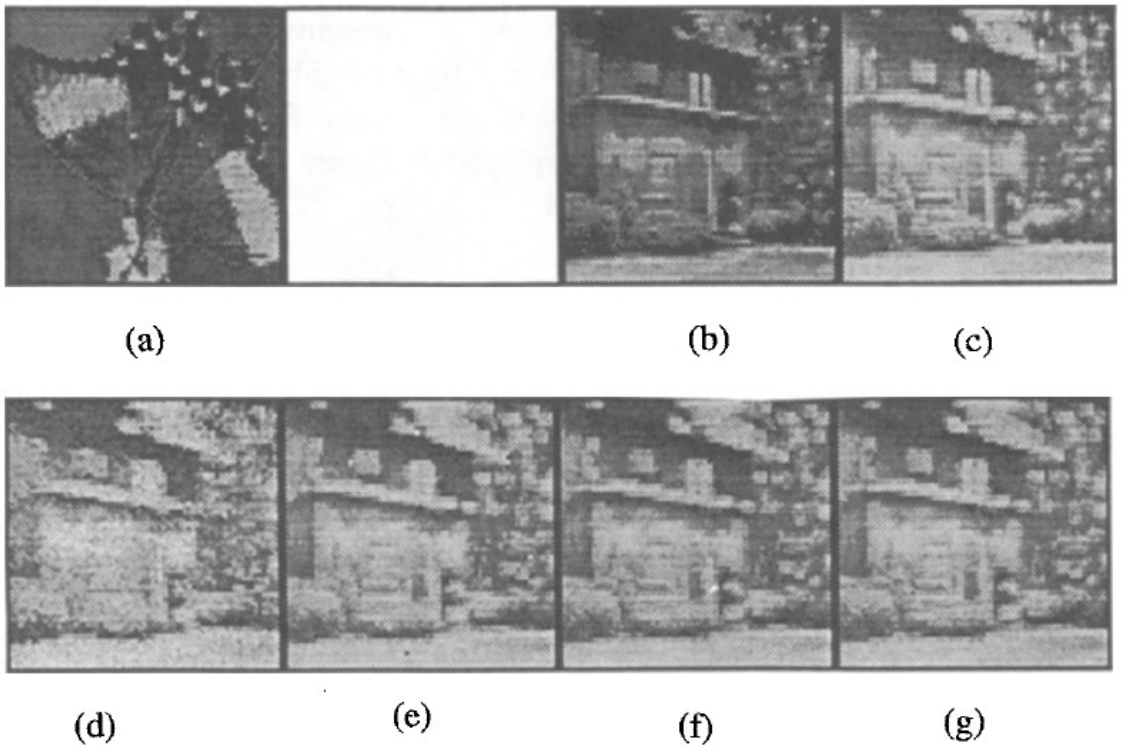


Figure 12. (a) One of the 13 training input images. (b) Test image. (c) Dilation of the test image by **t**. (d) Dilation of the test image by **w**(0). (e) Dilation of the test image by **w**(13). (f) Dilation of the test image by **w**(26). (g) Dilation of the test image by **w**(37) = **w**($3P - 2$).

G. X. Ritter & P. Sussner,
An introduction to morphological neural networks, 1996

Learning *some* decision surfaces (two classes)

G. X. Ritter & P. Sussner, *An introduction to morphological neural networks*, 1996

- Computing capabilities of Morphological Neural networks (can represent any boolean function)
- Morphological Associative Memories
- Single Layer Morphological Perceptron (SLMP): learning in finite steps but limited class of decision surfaces

$$\mathbf{p} = (p_1, p_2, \ldots, p_n) \in \mathbb{Z}^n \qquad W = \{w_1, w_2, \ldots, w_n\} \in \mathbb{Z}^n_{\pm\infty}$$

$$g(\mathbf{p}) = \bigvee_{i=1}^{n} [p_i + w_i] \qquad f(x) = \begin{cases} 1 & if \ \ x > 0 \\ 0 & else \ . \end{cases}$$

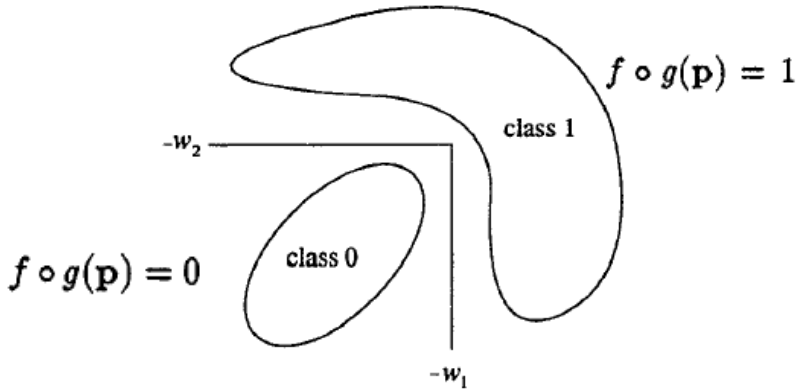


Fig. 3. Decision boundary for a single layer morphological perceptron.

P. Sussner, *Morphological perceptron learning*, 1998

Learning *any* decision surfaces (two classes)

P. Sussner, *Morphological perceptron learning*, 1998

- Generalized Single Layer Morphological Perceptron (SLMP)

$$f\left(\bigvee_{i=1}^{n} a_i\,(p_i + w_i)\right)$$

parameters $a_i \in \{1, -1\}$ are responsible for changing the signs of the sums $p_i + w_i$.

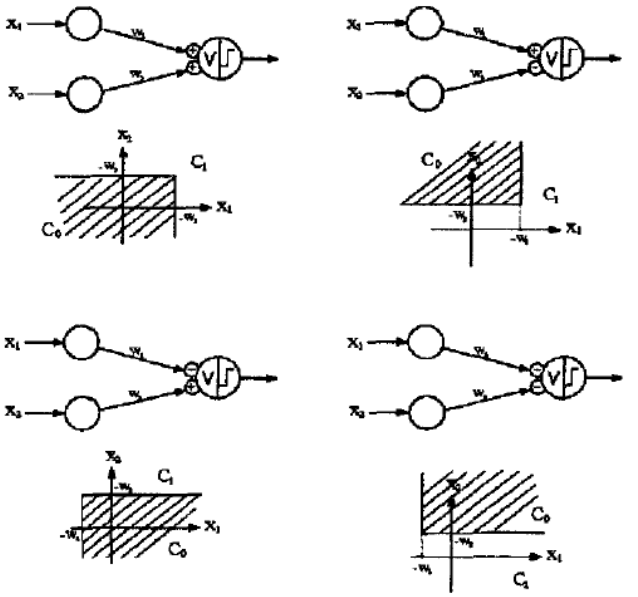


Figure 2:
Decision boundaries for different types of generalized single layer morphological perceptrons in the two-dimensional case.

- Multilayer Morphological perceptrons (MLMP): learning in finite steps and arbitrary decision surfaces (for 2 classes)

Finally, we present an algorithm for solving arbitrary instances of the 2-class problems by means of a two-layer morphological perceptron. Given a set of k training patterns in $\mathbb{R}^n$, this algorithm finds appropriate weights for a two-layer morphological perceptron such that each of the training patterns is correctly classified by the two-layer morphological perceptron. This algorithm also determines how many nodes in the hidden layer are necessary in order to solve the given problem.

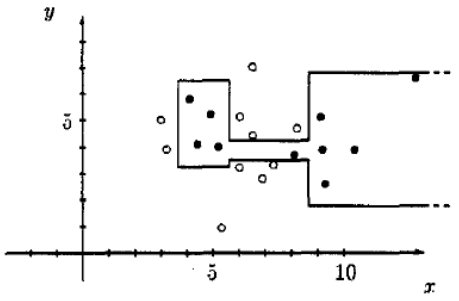


Figure 6:
Decision surface found by learning algorithm.

P. Sussner, *Morphological perceptron learning*, 1998

- Generalized Single Layer Morphological Perceptron (SLMP)

$$f\left(\bigvee_{i=1}^{n} a_i\,(p_i + w_i)\right)$$

parameters $a_i \in \{1, -1\}$ are responsible for changing the signs of the sums $p_i + w_i$.

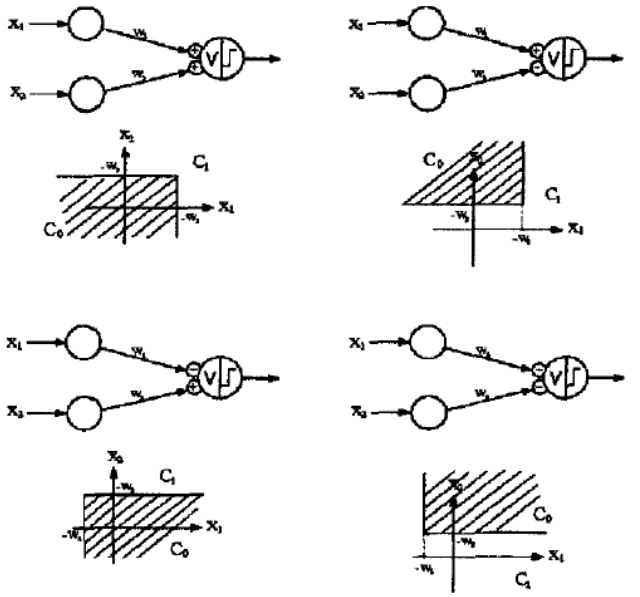


Figure 2:
Decision boundaries for different types of generalized single layer morphological perceptrons in the two-dimensional case.

- Multilayer Morphological perceptrons (MLMP): learning in finite steps and arbitrary decision surfaces (for 2 classes)

Learning in conventional multilayer perceptrons can be achieved by minimizing a certain error function which depends on the weights. Finding the global minimum of this error function is a very difficult task, and is not always possible. Weight training in multilayer morphological perceptrons does not encounter such limitations.

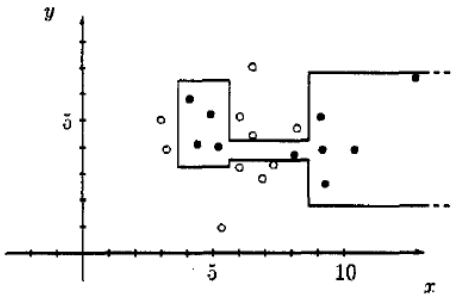
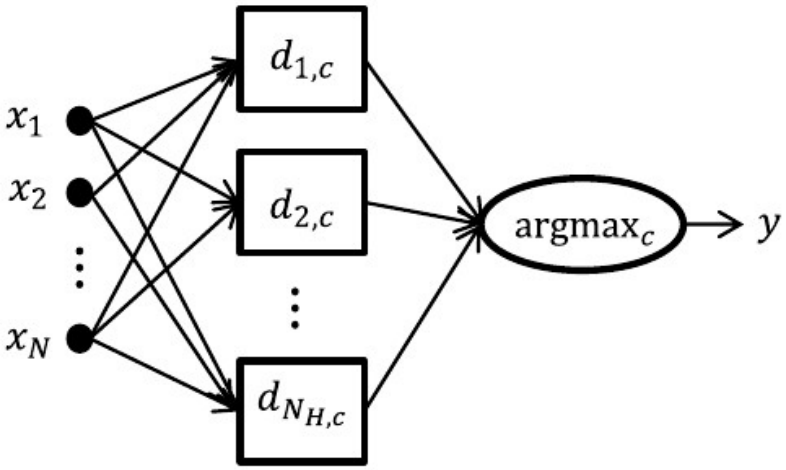


Figure 6:
Decision surface found by learning algorithm.

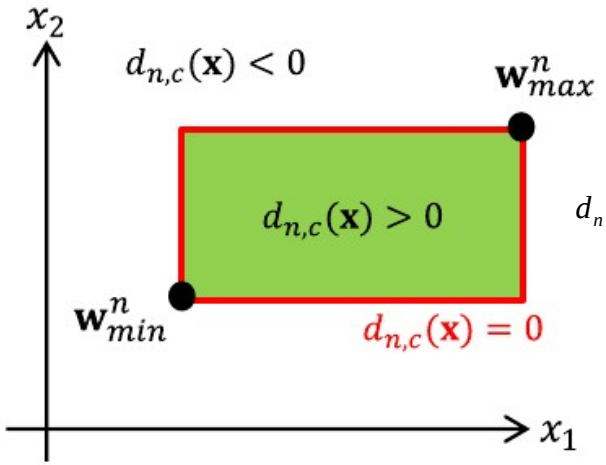
E. Zamora, H. Sossa,
Dendrite morphological neurons trained by stochastic gradient descent,
2017

Improving the learning of hyperboxes with SGD

E. Zamora, H. Sossa, *Dendrite morphological neurons trained by stochastic gradient descent*, 2017

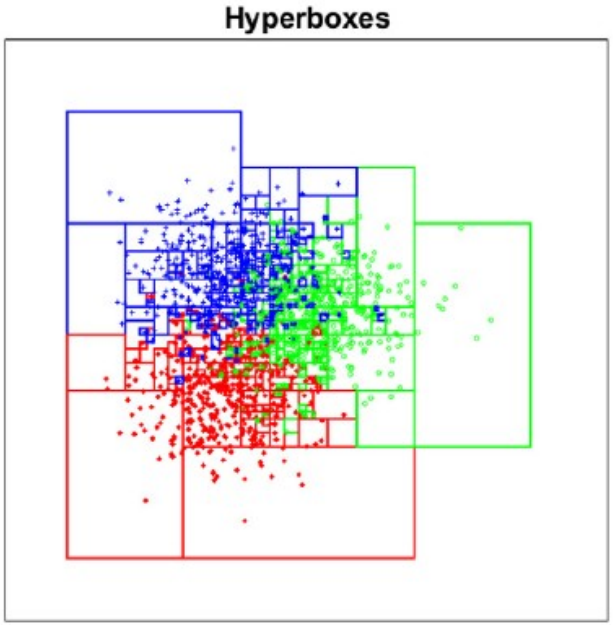


(a)

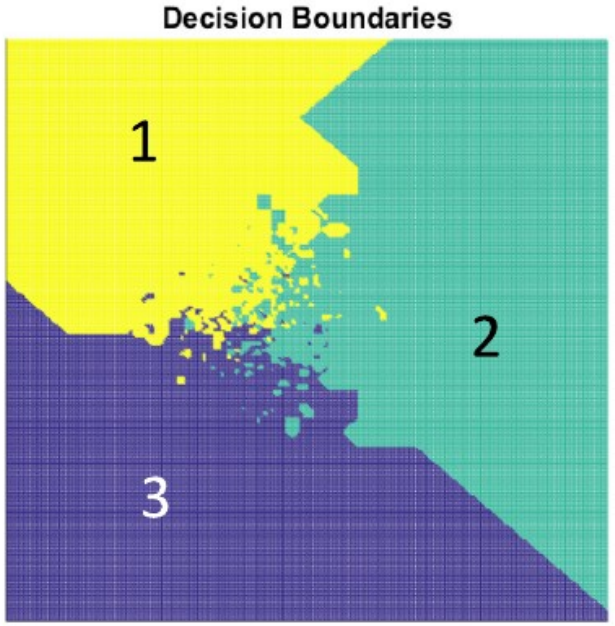


(b)

$$d_{n,c}(x) = min_{1 \le i \le N}\left(min\left(x_i - w^n_{min,i}, w^n_{max,i} - x_i\right)\right)$$



(c)



(d)

E. Zamora, H. Sossa, *Dendrite morphological neurons trained by stochastic gradient descent*, 2017

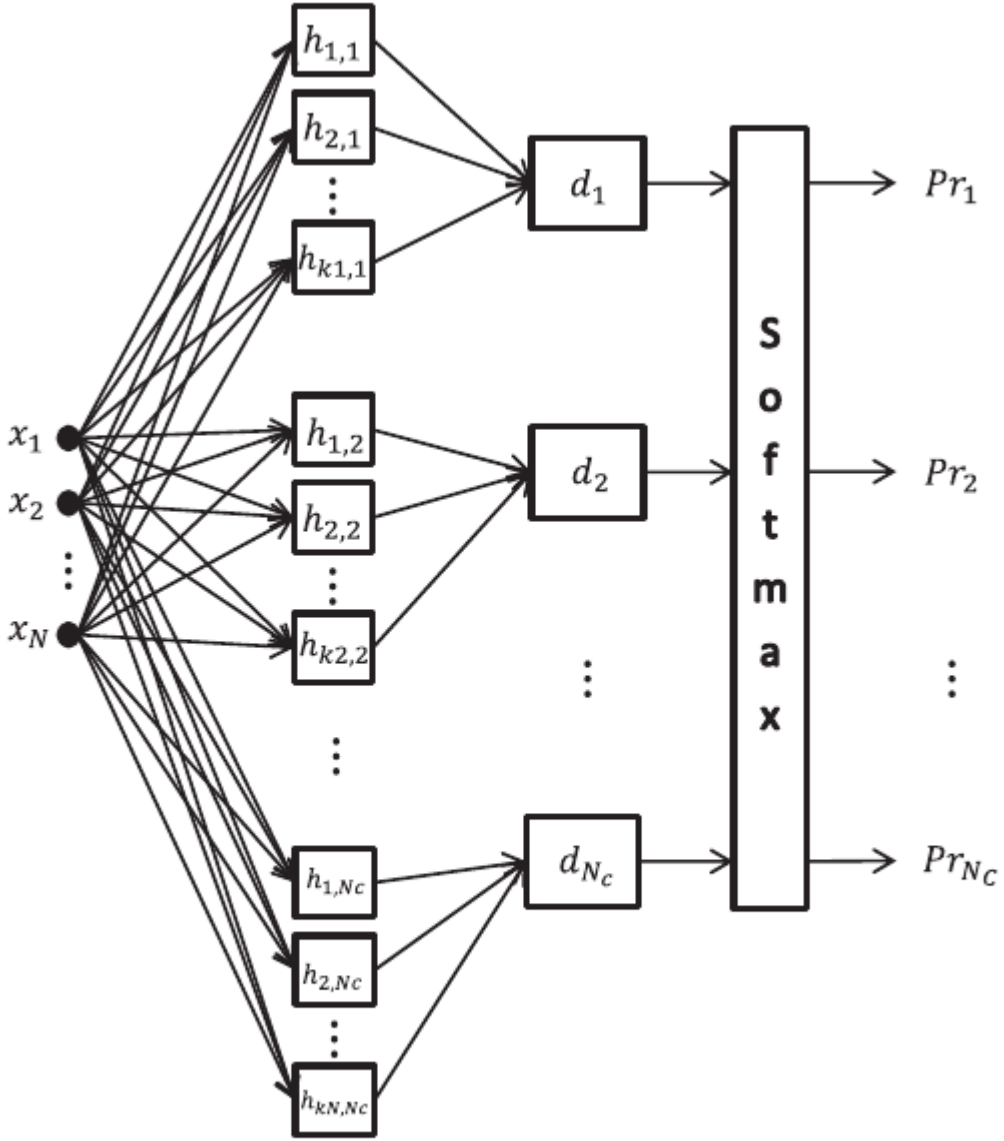


Fig. 3. Neural architecture for a DMN with a softmax layer. Each class corresponds to one dendrite cluster d_c.

$$h_{k,c}(\mathbf{x}) = \min\left(\min\left(\mathbf{x} - \mathbf{w}_{k,c}, \mathbf{w}_{k,c} + \mathbf{b}_{k,c} - \mathbf{x}\right)\right)$$

$$d_c(\mathbf{x}) = \max_k \left(h_{k,c}(\mathbf{x})\right)$$

$$Pr_c(\mathbf{x}) = \frac{\exp\left(d_c(\mathbf{x})\right)}{\sum_{k=1}^{N_c} \exp\left(d_k(\mathbf{x})\right)}$$

$$y = \arg\max_c (Pr_c(\mathbf{x}))$$

L. Pessoa, P. Maragos,
Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition, 2000

Between linear and morphological networks

L. Pessoa, P. Maragos, *Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition*, 2000

$$z_n^{(l)} \equiv \lambda_n^{(l)} \alpha_n^{(l)} + (1 - \lambda_n^{(l)}) \beta_n^{(l)},$$

$$\alpha_n^{(l)} = \mathscr{R}_{r_n^{(l)}}(\mathbf{y}^{(l-1)} + \mathbf{a}_n^{(l)}),$$

$$\beta_n^{(l)} = \mathbf{y}^{(l-1)} \cdot (\mathbf{b}_n^{(l)})' + \tau_n^{(l)},$$

where $\lambda_n^{(l)}, \tau_n^{(l)} \in \mathbb{R}$; $\mathbf{a}_n^{(l)}, \mathbf{b}_n^{(l)} \in \mathbb{R}^{N_{l-1}}$.

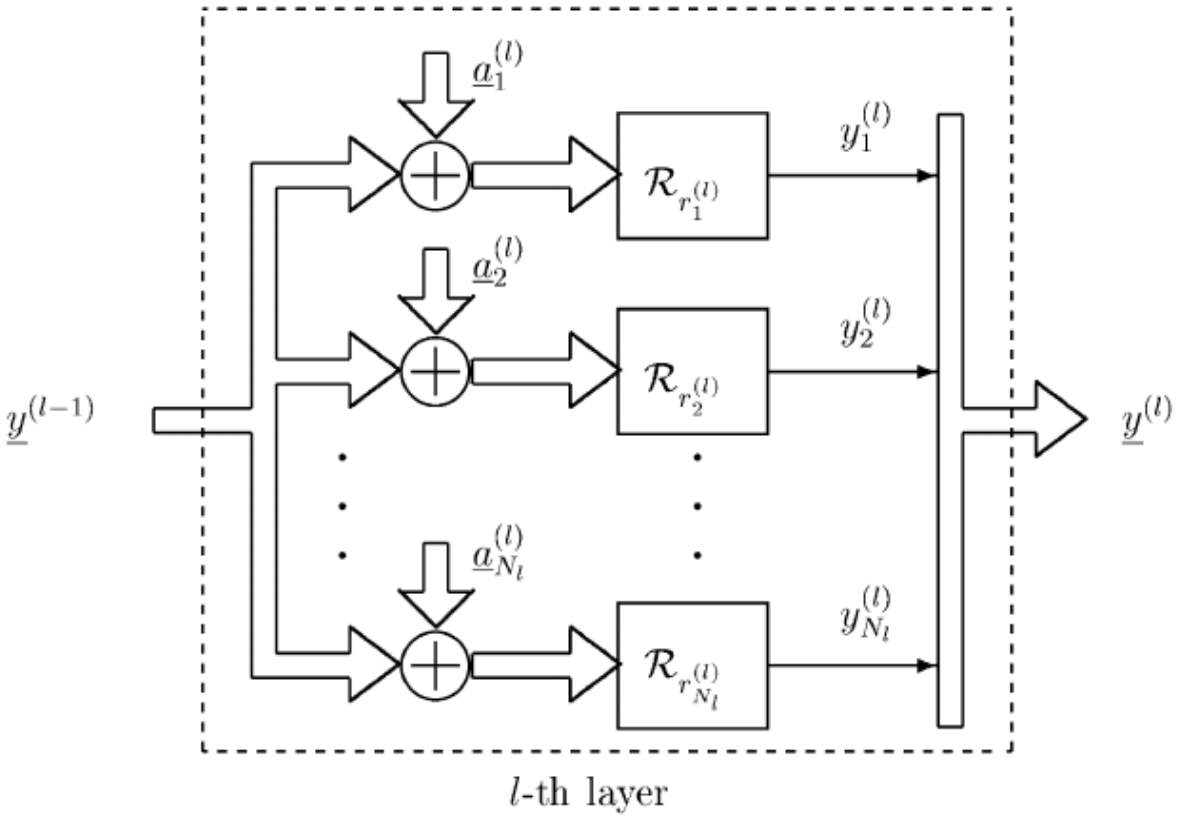
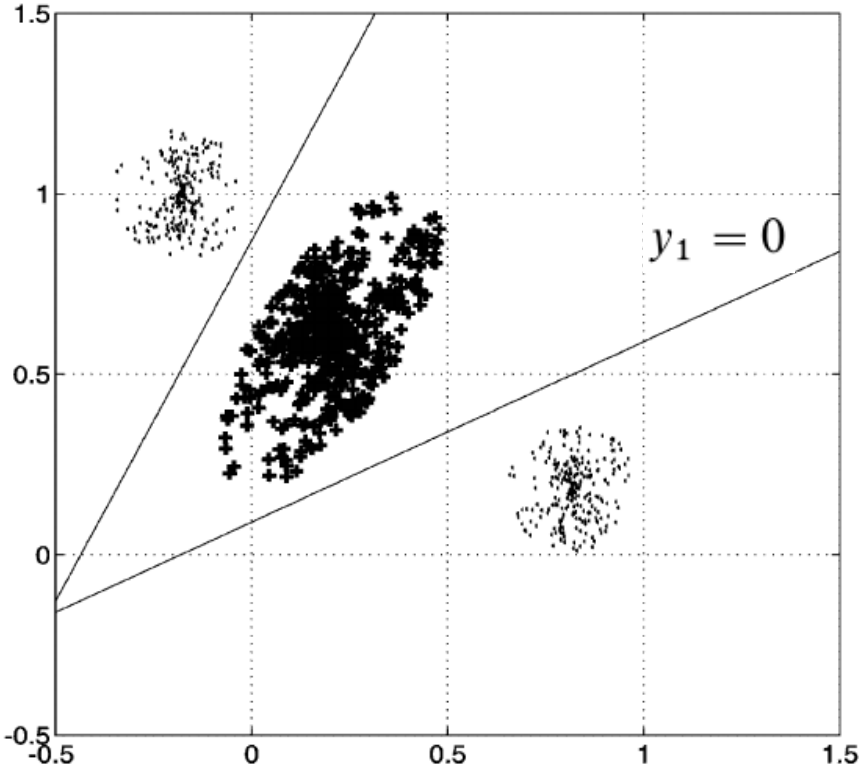


Fig. 2. Structure of the lth layer in an MRNN.

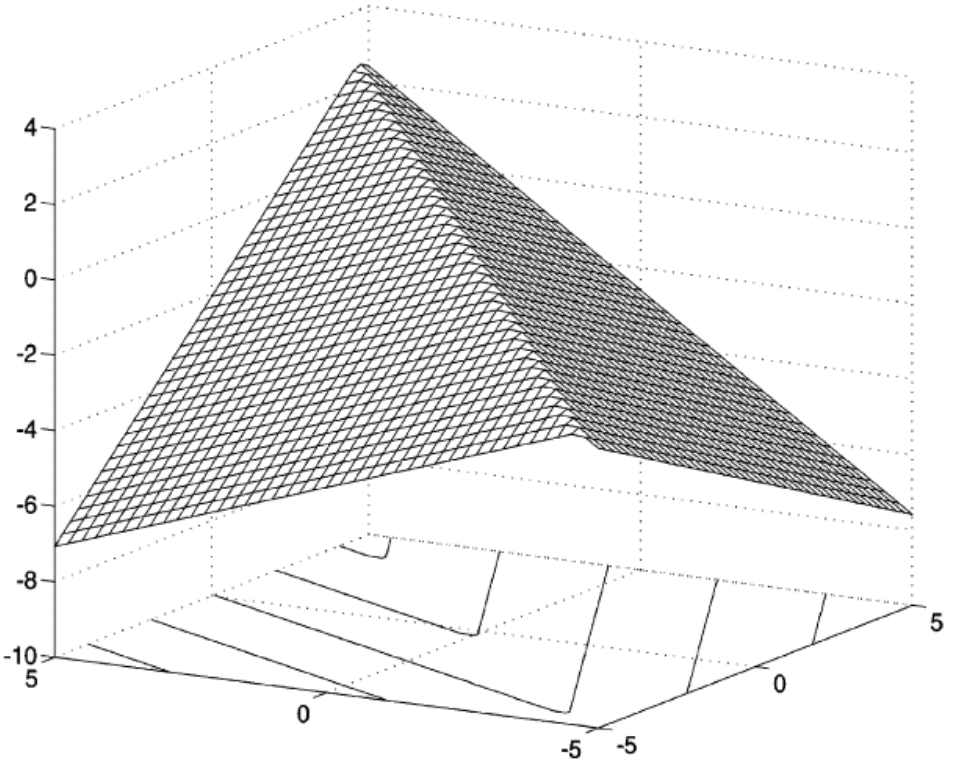
L. Pessoa, P. Maragos, *Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition*, 2000

Example:

$$y_1 = \lambda \min\{x_1 + a_1, x_2 + a_2\}$$
$$+ (1 - \lambda)(x_1 b_1 + x_2 b_2 + \tau_1).$$



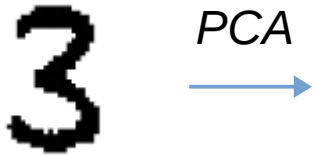
(a)



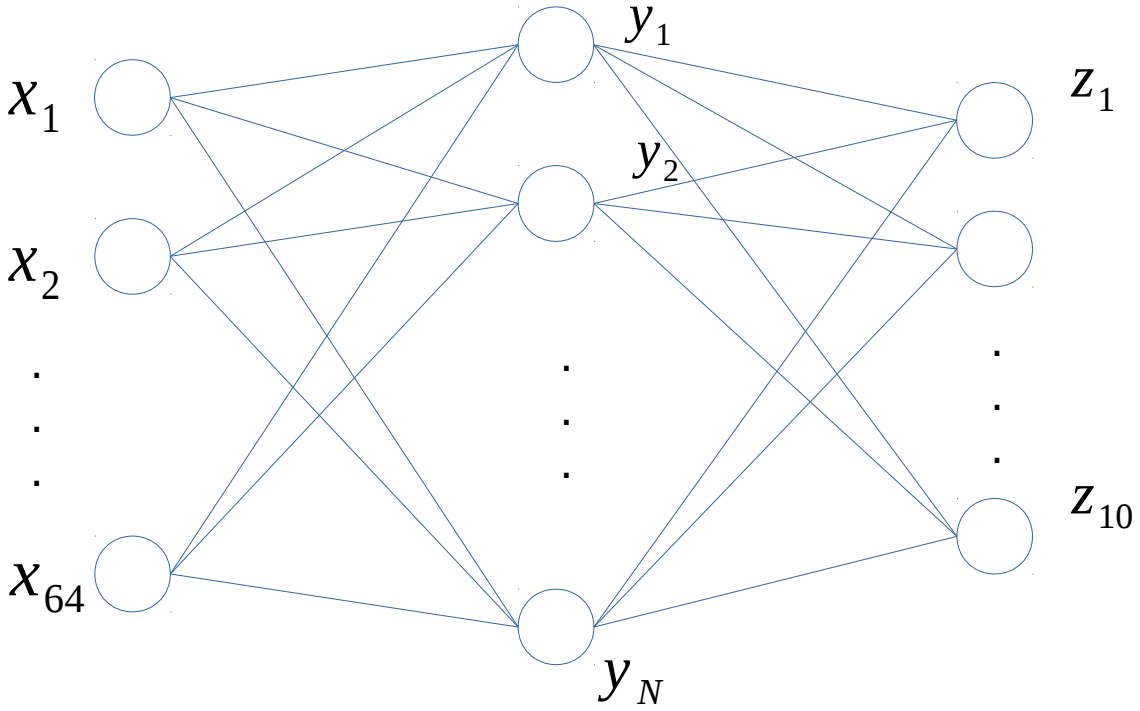
(b)

L. Pessoa, P. Maragos, *Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition*, 2000

Application to OCR (MNIST)



PCA



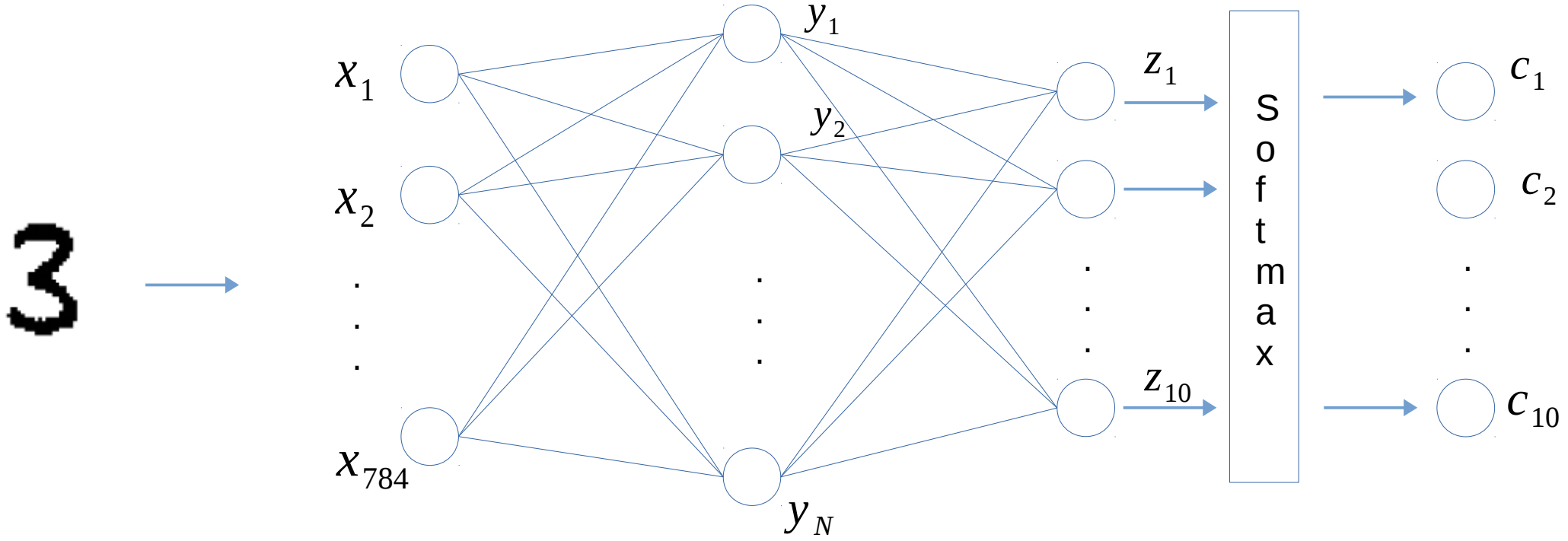
$N = 5, 10, 20$

MLP and MRL nets compared: similar performance with faster convergence for MRLs.

V. Charisopoulos, P. Maragos,
Morphological perceptrons: Geometry and Training Algorithms, 2017

A new method to prune networks?

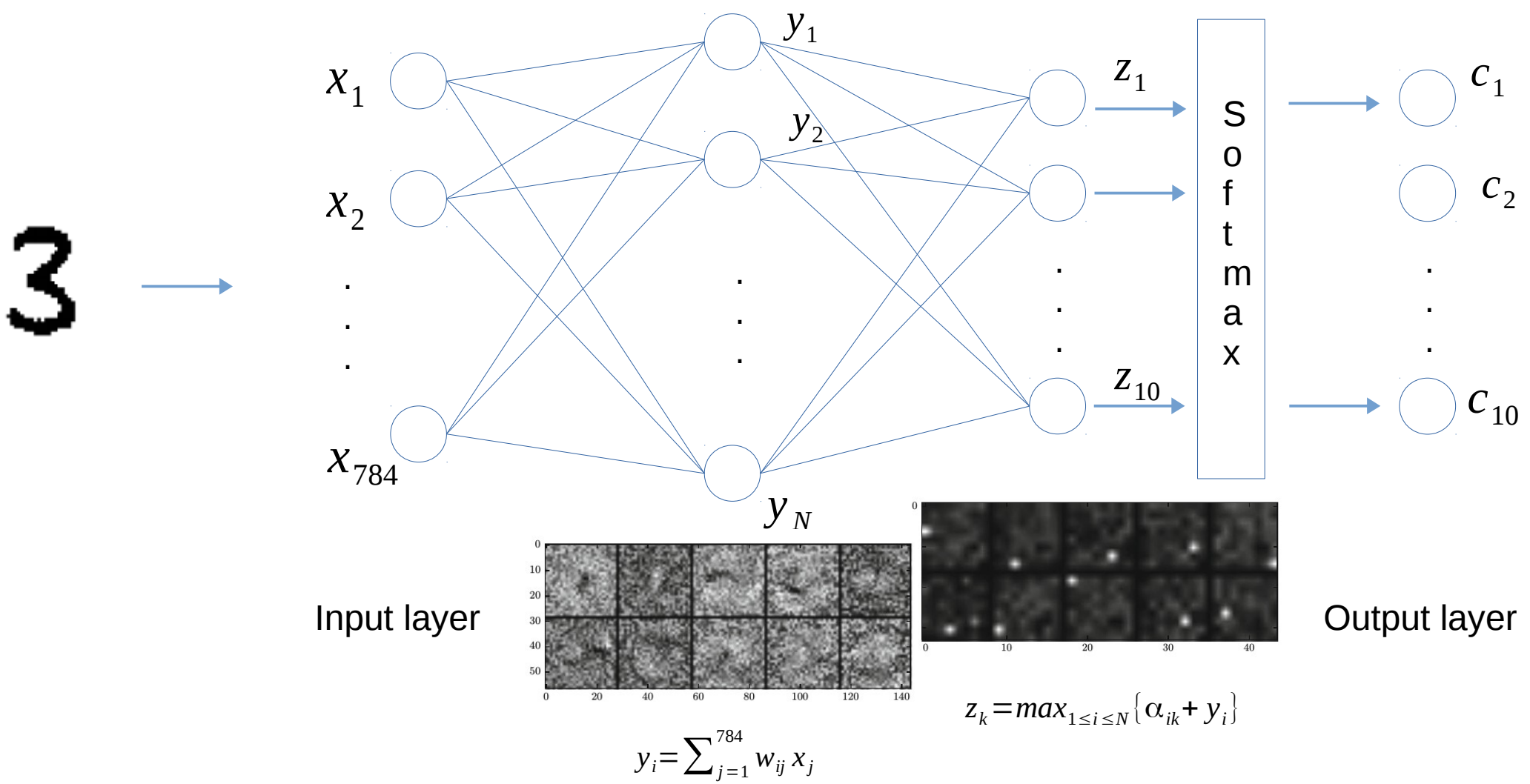
V. Charisopoulos, P. Maragos, *Morphological perceptrons: Geometry and Training Algorithms*, 2017



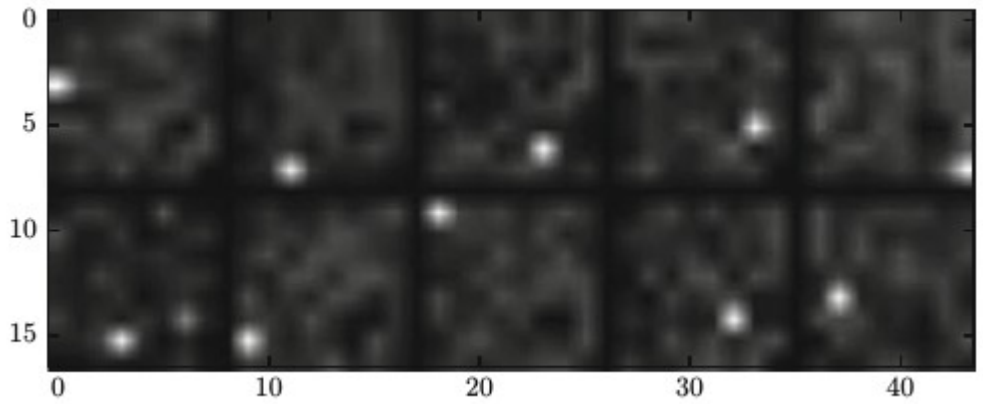
Input layer | Linear layer | Dilation layer | Output layer

$$y_i = \sum_{j=1}^{784} w_{ij}\, x_j \qquad z_k = max_{1 \le i \le N} \{ \alpha_{ik} + y_i \}$$

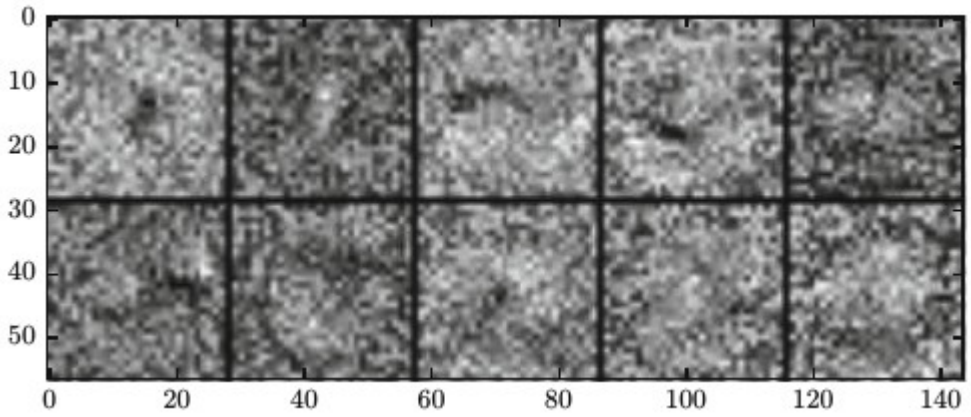
V. Charisopoulos, P. Maragos, *Morphological perceptrons: Geometry and Training Algorithms*, 2017



V. Charisopoulos, P. Maragos, *Morphological perceptrons: Geometry and Training Algorithms*, 2017



Weights α_{jk} *for* $N = 64$

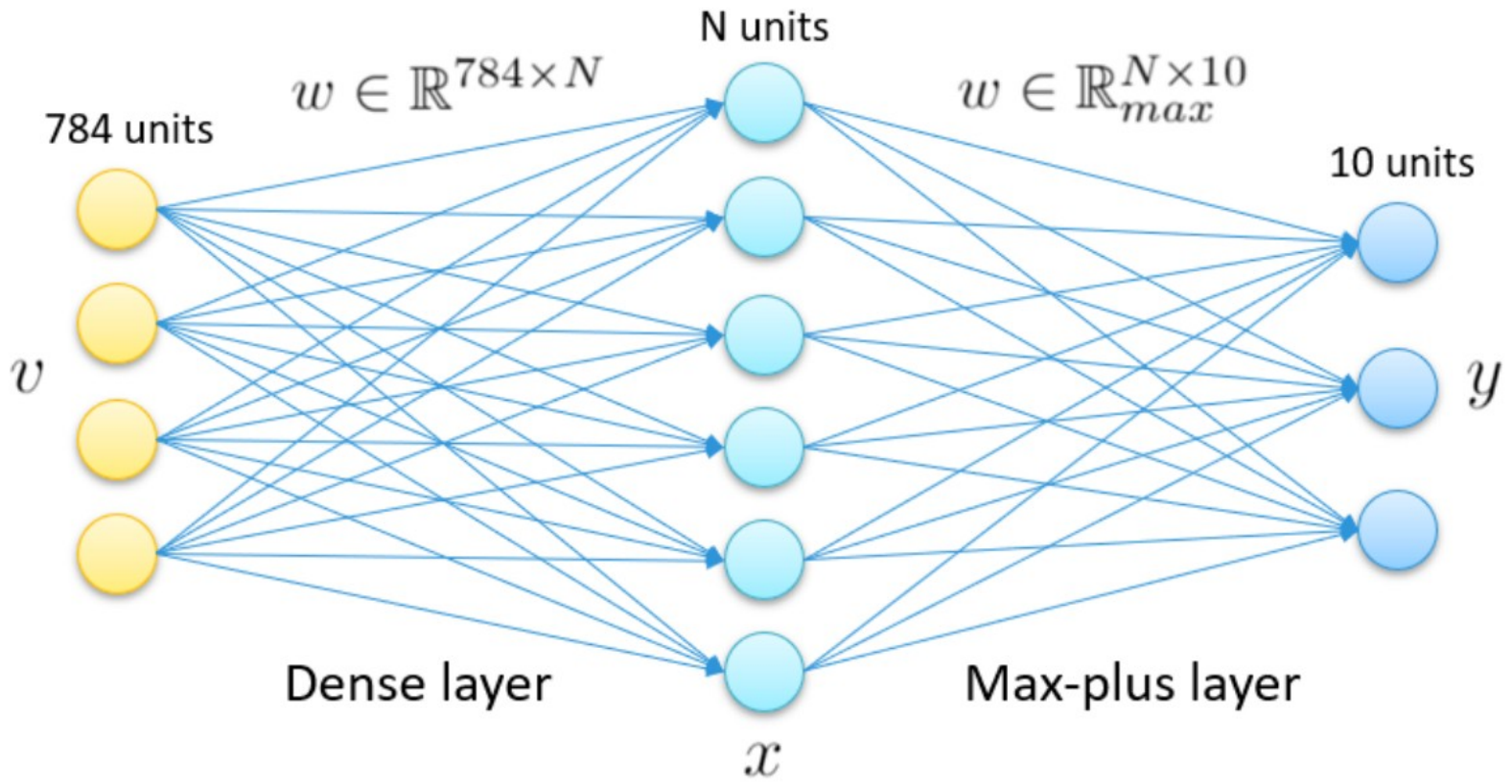


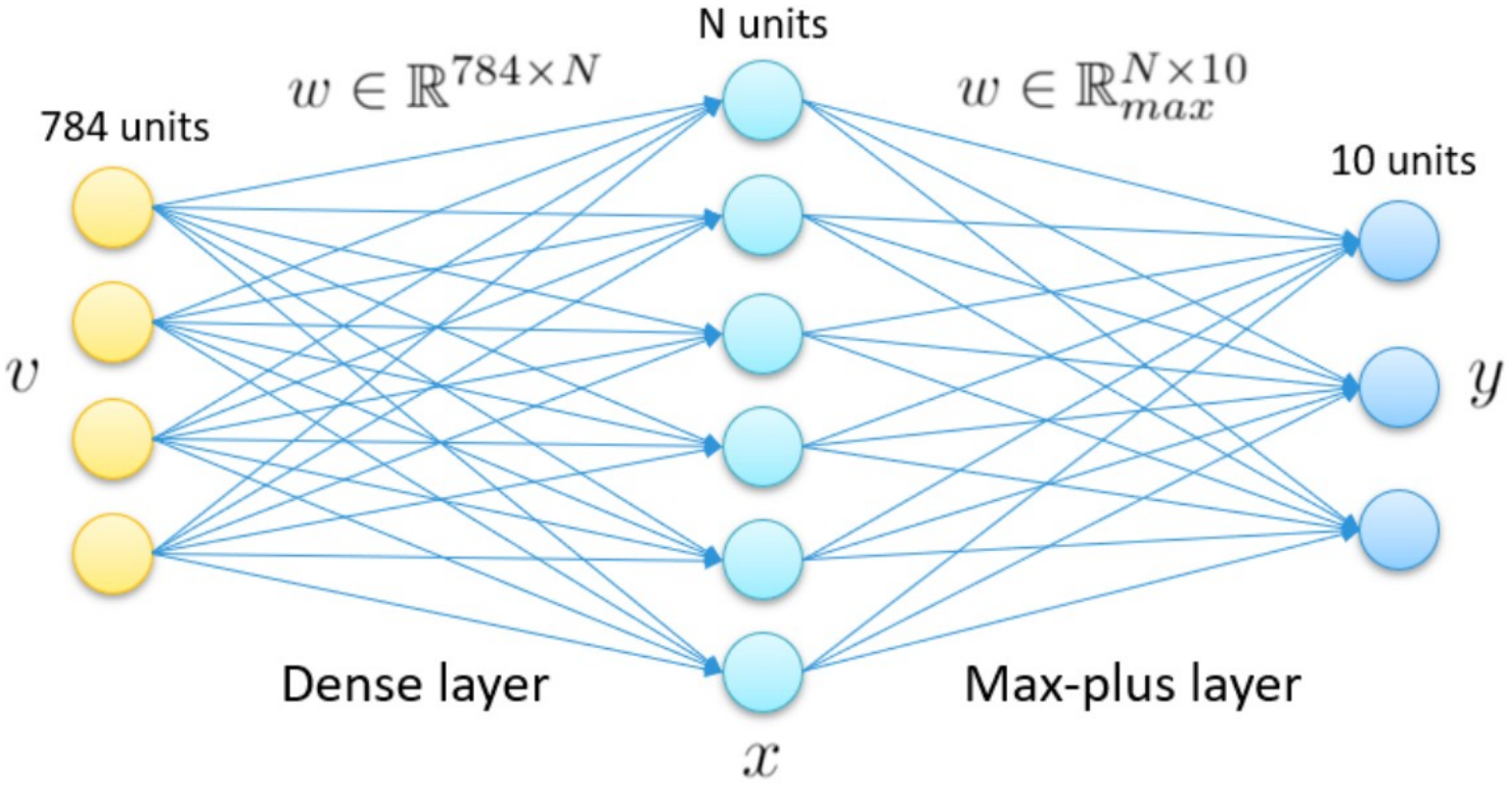
Weights w_{ij} *for the* 10 *active filters*

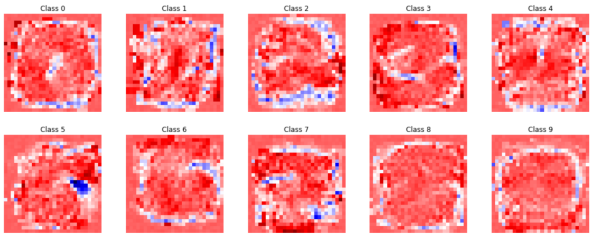
Table 2. MNIST results

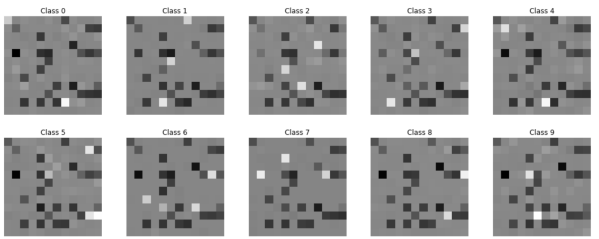
Layer n_1	Accuracy	Accuracy without "dead" units	# Active filters
24	84.29%	84.28%	17
32	84.84%	84.85%	15
48	84.63%	84.61%	18
64	92.1%	92.07%	10

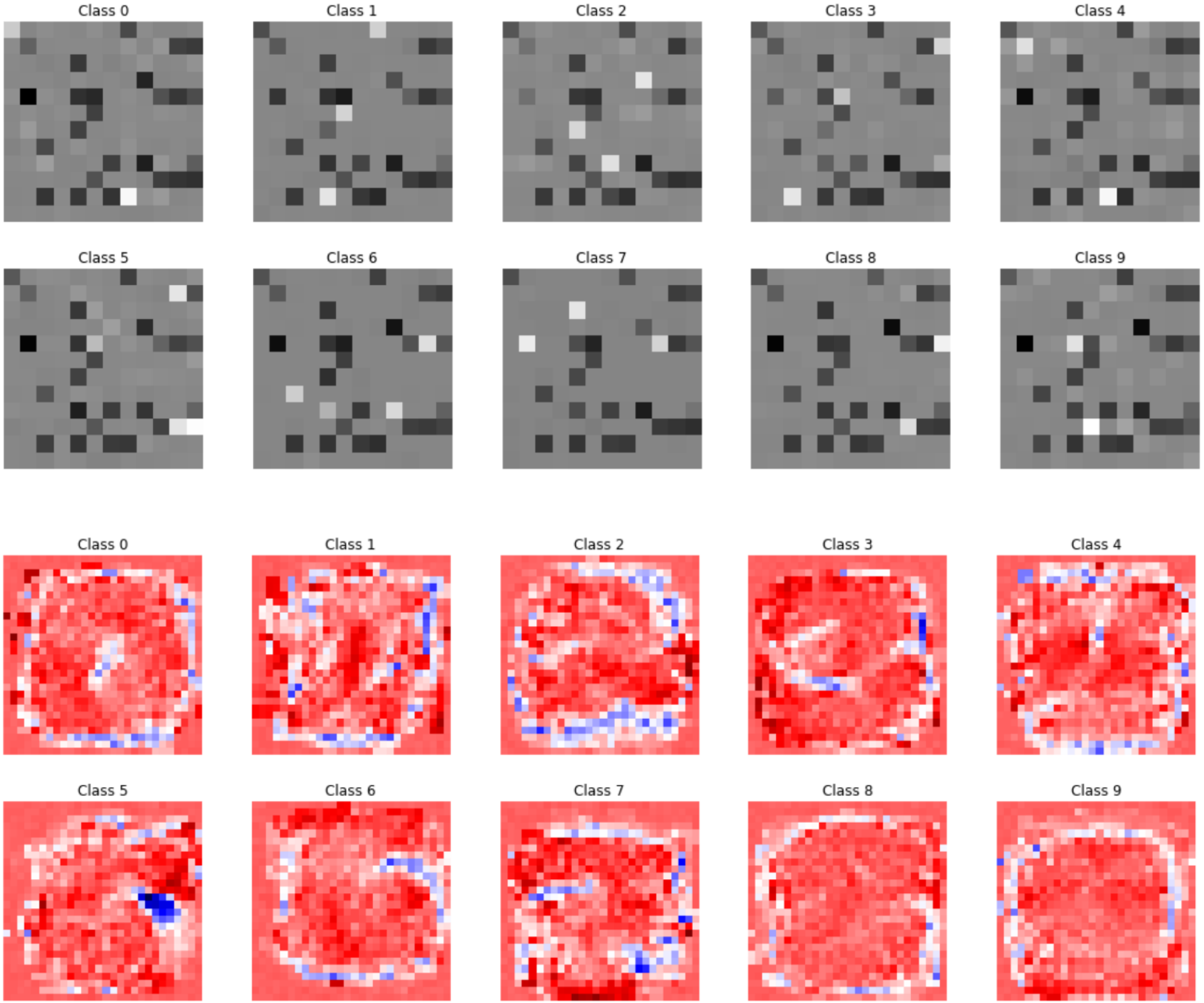
Yunxiang ZHANG,
Travaux de stage, 2018

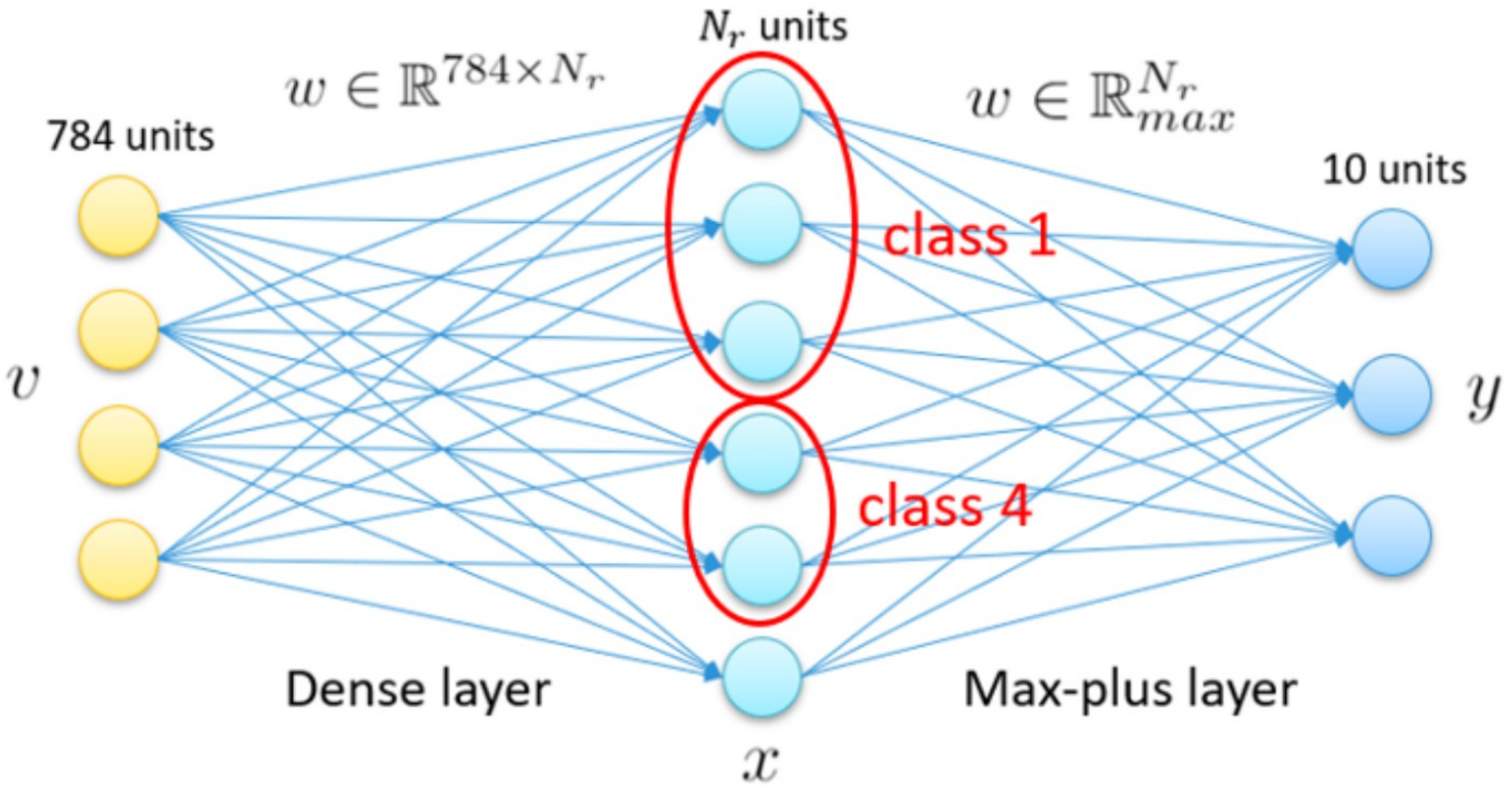

N units
$w \in \mathbb{R}^{784 \times N}$
$w \in \mathbb{R}^{N \times 10}_{max}$
784 units
10 units
v
y
Dense layer
Max-plus layer
x


N units
$w \in \mathbb{R}^{784 \times N}$
$w \in \mathbb{R}^{N \times 10}_{max}$
784 units
10 units
v
y
Dense layer
Max-plus layer
x


Class 0
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Class 7
Class 8
Class 9


Class 0
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Class 7
Class 8
Class 9


Class 0
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Class 7
Class 8
Class 9
Class 0
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Class 7
Class 8
Class 9


N_r units
$w \in \mathbb{R}^{784 \times N_r}$
$w \in \mathbb{R}^{N_r}_{max}$
784 units
10 units
class 1
class 4
v
y
Dense layer
Max-plus layer
x