SAND2006-4225C

The Synaptic Morphological Perceptron

Daniel S. Myers*
Sandia National Laboratories, P.O.Box 5800, Albuquerque, NM USA 87185-MS0401

ABSTRACT

In recent years, several researchers have constructed novel neural network models based on lattice algebra. Because of computational similarities to operations in the system of image morphology, these models are often called *morphological neural networks*. One neural model that has been successfully applied to many pattern recognition problems is the *single-layer morphological perceptron with dendritic structure* (SLMP). In this model, the fundamental computations are performed at *dendrites* connected to the body of a single neuron Current training algorithms for the SLMP work by enclosing the target patterns in a set of hyperboxes orthogonal to the axes of the data space. This work introduces an alternate model of the SLMP, dubbed the *synaptic morphological perceptron* (SMP). In this model, each dendrite has one or more *synapses* that receive connections from inputs. The SMP can learn any region of space determined by an arbitrary configuration of hyperplanes, and is not restricted to forming hyperboxes during training. Thus, it represents a more general form of the morphological perceptron than previous architectures.

Keywords: Morphological neural network, morphological perceptron, lattice algebra, pattern recognition

1. INTRODUCTION

Artificial neural network models based on lattice algebra have become increasingly popular in recent years. These networks are based on algebraic structures that include the discrete maximum and minimum operators, giving them properties different from more traditional network architectures based on lattice algebra. In particular, the model of the single-layer morphological perceptron with dendritic structure has demonstrated applicability to a wide variety of learning tasks.

In the SLMP, the fundamental computations are performed by *dendrites* connected to the neuron body. This is in accord with recent research on biological neurons, which suggests dendrites have a fundamental computational role in the human brain. Thus, the SLMP architecture more closely imitates brain structure than many other artificial neural network models.

Current techniques for training the SLMP focus on enclosing the target patterns with hyperboxes that are orthogonal to the axes of the data space. These techniques have many desirable properties, including fast convergence, clear geometric interpretation, and 100% accurate classification of training data. However, they can only approximate data configurations that are not determined by orthogonal hyperboxes.

This work presents a new model of the SLMP, and a new training method based on the error back-propagation techniques developed for the classical multi-layer perceptron. This model has the ability to accurately learn regions of space that are not defined by orthogonal hyperboxes. Thus it is a morphological perceptron with more general learning capabilities.

The rest of this work is organized as follows: Section 2 discusses lattice algebra and the computational basis for morphological perceptrons. Section 3 briefly discusses current morphological perceptron architectures. Section 4 presents the new model of the synaptic morphological perceptron and its training algorithm, and Section 5 presents an extension to multi-class prblems. Section 6 gives the results of experimental tests on two benchmark data sets. Finally, Section 7 offers conclusions and avenues for future research.

2. COMPUTATIONAL BASIS FOR MORPHOLOGICAL NEURAL NETWORKS

Morphological neural networks are a family of artificial neural network models that use computations based on lattice algebra. Most existing neural network paradigms use the basic operations of addition and multiplication performed over the real numbers. Algebraically, this structure is called a *ring* and is denoted $\{\Re, +, \times\}$. The fundamental neural computation performed by these networks is a sum-of-products, such as

$$\tau_j = \sum_{i=1}^{n} w_{ij} x_i \ , \tag{1}$$

where x_i is the output from the *i*th neuron, w_{ij} is the weight of the connection between the *i*th and *j*th neurons, and τ_j is the total input to the *j*th neuron.

In contrast, morphological neurons are based on the *semi-ring* $\{\Re, \wedge, \vee, +\}$, and the basic computation of a morphological neuron is a max-of-sums,

$$\tau_j = \bigvee_{i=1}^{n} w_{ij} + x_i \ , \tag{2}$$

or min-of-sums,

$$\tau_j = \bigwedge_{i=1}^{n} w_{ij} + x_i \ . \tag{3}$$

Equations (2) and (3) correspond respectively to the operations of dilation and erosion in the system of image morphology developed by J. Serra [SERRA]. This is the origin of the name *morphological neural networks*.

Neural networks based on lattice algebra have several interesting properties. First, the maximum and minimum operators are inherently nonlinear. Traditional network models must use an activation function, such as a sigmoid, to achieve nonlinear behavior. Second, lattice-based methods lack multiplication operations, or considerably reduce their number if they are included. Because additions and comparisons execute more quickly than multiplications, algorithms based on lattice algebra may out perform their similar linear algebra-derived algorithms, even if the two methods have the same computational complexity. Finally, lattice-based algorithms are well-suited for implementation on programmable logic devices or application specific circuits, since they require comparatively simpler logic.

3. THE SINGLE-LAYER MORPHOLOGICAL PERCEPTRON

This section provides an overview of existing models of the morphological perceptron. The reader is referred to [RITTER-INTRO] for a more complete general introduction to the single-layer morphological perceptron. Iancu developed, tested, and compared many different learning algorithms for the SLMP in [IANCU].

The structure of the morphological neuron is inspired by the structure of real neurons in the brain [REF]. In biological neurons, the body of the neuron – called the *soma* – receives input from one or more *dendrites*, and transmits output along an *axon*. The dendrites and axon may branch and divide to create complicated connections between the soma and other neurons. Signals from the soma are carried along the axon and transmitted to the dendrites of other neurons. The point where an axon meets a dendrite is called a *synapse*, and communication between neurons takes place at synapses. Further, synapses may be *excitatory* and play a role in inducing the neuron to fire, or *inhibitory*, and prevent the neuron from spiking.

As its name implies, the single-layer morphological perceptron consists of a single neuron that receives input from a layer of input nodes. Unlike traditional artificial neural network models, a hidden layer of nodes is not required to solve nonlinear classification problems. For the purposes of this discussion, we will consider a one-class learning problem, where patterns belong to either class 1 or class 0, but we are only interested in constructing a model that positively identifies class 1 patterns.

The SLMP neuron possesses a number of dendrites; each dendrite receives input from one or more neurons in the input layer. Each connection between a dendrite and an input neuron possesses an additive weight, which determines values for which that dendritic connection is "active." If a given input activates all of the connections on a particular dendrite, then the dendrite itself is activated. In current SLMP architectures, each dendrite encodes an orthogonal hyperbox in $\Re^N$. If the input pattern lies inside the dendritic hyperbox, the dendrite is activated and passes a positive value to the neuron's soma.

Dendrites in the SLMP may be either excitatory or inhibitory. If an input pattern lies in the hyperbox of an excitatory dendrite, but not in any inhibitory hyperbox, then the final value at the soma is positive, and the neuron fires. If an input pattern lies inside the hyperbox of an inhibitory dendrite, then the final value at the soma will be negative. Likewise, patterns that lie outside any dendritic hyperbox will generate a negative output at the soma.

The SLMP learns by adjusting its connection weights so that all target patterns lie inside an excitatory hyperbox, but not in an inhibitory hyperbox. The following basic training algorithm is based on the one detailed in [RITTER-INTRO]:

1) Construct an excitatory dendrite whose hyperbox encloses all the class 1 patterns.
2) Select a pattern from class 0 that is misclassified by the current configuration of weights.
3) Construct an inhibitory dendrite whose hyperbox surrounds the misclassified pattern, but does not include any class 1 patterns.
4) If no class 0 patterns are misclassified, stop training. Otherwise, go to step (2).

This algorithm produces one large hyperbox that encloses all of the class 1 patterns in the training set. If any class 0 patterns are incorrectly classified by this dendrite, the algorithm grows additional inhibitory dendrites that "carve out" regions surrounding the class 0 patterns, so that the neuron does not fire when presented with a pattern in an inhibitory hyperbox.

This algorithm has several desirable properties. In particular, training is guaranteed to complete in a finite number of steps, and a trained model will always yield 100% correct classification on the training data. However, there are many data configurations that cannot be easily modeled by orthogonal hyperboxes. Consider the triangular region of Fig. 1. The region is described by only three lines, but can only be covered by an infinite number of orthogonal hyperboxes.

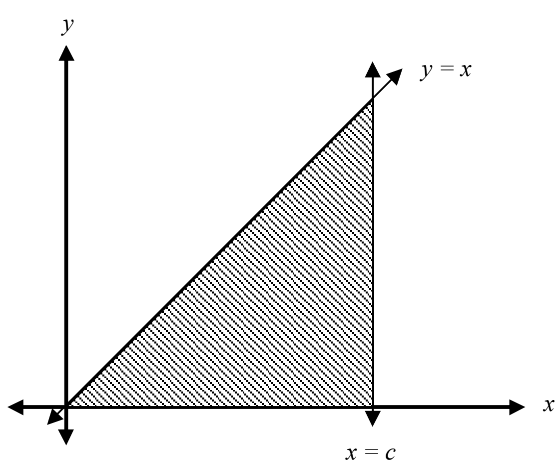

Fig. 1. A simple triangular region that cannot be modeled by a finite number of orthogonal hyperboxes.

4. THE SYNAPTIC MORPHOLOGICAL PERCEPTRON

This section describes the model of the SMP, which has the ability to learn regions of space bounded by non-orthogonal hyperplanes in $\Re^N$. First, the structure of the perceptron is discussed, then the back-propagation training algorithm is derived and presented. Fig 2 shows the model of the SMP with a single input neurons.

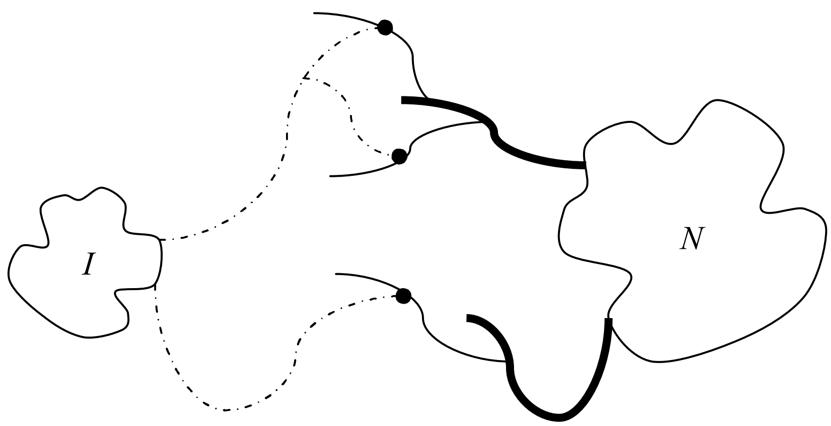

Fig. 2. An example of the synaptic morphological perceptron with a single input neuron, *I*. The neuron *N* has two dendrites, with a total of three synapses that are connected to the input neuron.

4.1 Structure of the Neuron

The SMP neuron strongly resembles the neuron of the SLMP discussed in Section 3. The neuron possesses a set of dendrites D and each dendrite $d \in D$ possesses a set of synapses, denoted as S^d. The input neurons are connected to these synapses.

Though the overall structure of the neuron is similar, the computation performed at the synapse is different from the described in Section 3. Let the vector $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ be the input to the SMP. We assume a fully connected model; that is, every input $x_i \in \mathbf{x}$ has exactly one connection to each synapse on the SMP. Further, each of these synaptic connections has two associated weights: a multiplicative weight, denoted w_i^s -- interpreted as the *i*th weight connected to synapse s -- and an additive weight a_i^s. Finally, all connections in the SMP are excitatory. The effect of inhibitory connections will emerge naturally as the weights are adjusted during training.

The computation at the synapse *s* is given by

$$\tau_s = \sum_{i=1}^{n} w_i^s \left(x_i + a_i^s \right). \tag{4}$$

This expression can be rewritten by combining all of the weights and inputs in vector form as follows:

$$\tau_s = \mathbf{w}^s \left(\mathbf{x} + \mathbf{a}^s \right). \tag{5}$$

Thus, the computation at the synapse determines a hyperplane with the equation

$$\mathbf{w^s x} + \mathbf{w^s a^s} = 0 . \tag{6}$$

The computation at the dendrite is the minimum of the dendrite's synaptic computations:

$$\tau_d = \bigwedge_{s \in S^d} \tau_s , \tag{7}$$

where S^d is the set of synapses connected to dendrite $d \in D$. Note that the output of the dendrite is positive if and only if all of the values computed at its synapses are positive.

Finally, the value at the neuron is given by the *maximum* of all the values computed at the dendrites:

$$\tau = \bigvee_{d \in D} \tau_d . \tag{8}$$

Note that this differs from original SLMP model described in Section 3, which used the minimum operator.

4.2 Geometric Interpretation of the Neuron

We can develop an explicit geometrical interpretation for the computations performed by the SMP. Each synapse encodes a hyperplane that divides $\Re^N$ into positive and negative halfspaces. A pattern **x** lies in the positive halfspace determined by the synapse *s* if

$$\mathbf{w^s x} + \mathbf{w^s a^s} \geq 0 , \tag{9}$$

where we consider points that lie on the hyperplane itself to be part of the positive halfspace. The computation performed at a dendrite *d* is positive if and only if $\tau_s \geq 0$ for all $s \in S^d$. Geometrically, the region learned by the dendrite is the *intersection* of the positive halfspaces learned by its synaptic hyperplanes. Finally, the region learned by the neuron itself is the *union* of the regions learned by each of its dendrites.

4.3 Training the Synaptic Morphological Perceptron Using Error Back Propagation

Training the SMP requires manipulating the additive and multiplicative weights of its synapses. Training is done in the familiar supervised learning paradigm: the neuron is presented with a pattern from the training set, whose class is known. If the neuron classifies the pattern incorrectly, the parameters of the SMP are adjusted so that correct classification is more likely in future trials. For this section, all training will be done in a one-class framework – input patterns either belong to a single class of interest, or they belong to no class. Patterns are assigned to the class if $\tau \geq 0$, where τ is the final output of the SMP neuron. An extension to multi-class problems will be presented in Section 5.

Nobuharu, Bede, and Hirota previously studied back-propagation training in morphological networks in [REF]. However, they used a simpler multi-layer morphological perceptron, which does not include dendritic structures, and lacks the learning power of the SMP, as it uses only additive weights and the maximum operator.

The key to morphological back-propagation in the SMP lies in a simple observation: for a given input pattern, exactly one synaptic computation determines the class output by the neuron. This is the "winning" synapse on the "winning" dendrite, that is, the synapse s^* on the dendrite d^* such that

$$\tau_{s^*}^{d^*} = \bigvee_{d \in D} \bigwedge_{s \in S} \tau_s^d , \tag{10}$$

where τ_s^d is the value computed at synapse s on dendrite d. Since any error at the output depends only on error in the parameters of this synapse, we only need to update one set of weight vectors at each training step. Let **w** and **a** be the multiplicative and additive weights of the winning synapse, respectively.

We can adjust the winning weights using the gradient descent method. Let the error at the output neuron be given by

$$E = \frac{1}{2} \| \tau - \tau' \|^2 , \tag{11}$$

where τ is the output of the neuron and τ' is the desired output. Taking the derivative of the error with respect to w_i and a_i yields

$$\frac{\partial E}{\partial w_i} = x_i + a_i \tag{12}$$

$$\frac{\partial E}{\partial a_i} = w_i . \tag{13}$$

Using these gradients, we can adjust the synaptic weights to reposition the synaptic hyperplane and decrease the error:

$$w_i = w_i + \eta(\tau' - \tau)(x_i + a_i) \tag{14}$$

$$a_i = a_i + \eta(\tau' - \tau)(w_i), \tag{15}$$

where η is the learning rate. Notice that if $(\tau' - \tau) = 0$, the neuron's output matches the desired value, and no weights are updated.

The full training algorithm for the synaptic morphological perceptron is as follows:

1) Randomly initialize the multiplicative and additive synaptic weights.
2) Randomly select an input pattern from the training set.
3) Compute the values at each synapse using equation (5).
4) Compute the values at each dendrite using equation (7). Record the index of the minimum synapse at each dendrite.
5) Compute the value at the neuron using equation (8). Determine the overall winning synapse using the indices recorded in step (4).
6) Compute the error and update the parameters of the winning synapse.
7) Repeat steps (2) through (6) until all training patterns are classified correctly, or another stopping condition is satisfied.

4.4 Comparison to Previous SLMP Models

This method differs from existing SLMP training algorithms in a number of ways. First, training using non-orthogonal hyperplanes and back-propagation will usually require fewer dendrites than training using hyperboxes. Second, hyperbox-based training methods guarantee 100% correct classification on the training set as a consequence of their construction. The SMP training algorithm makes no such guarantee. However, this property makes hyperbox-based methods sensitive to noise in the training set, as these methods cannot avoid learning patterns that may be erroneously positioned. Finally, the hyperbox-based SLMP is guaranteed to converge in a finite number of training epochs. The back-propagation method is not guaranteed to converge, and requires a stopping condition – such as a bound on the maximum number of epochs – to ensure that training will terminate.

5. EXTENSION TO MULTI-CLASS PROBLEMS

Thus far, we have seen how the synaptic morphological perceptron can solve a one-class learning problem. Recall that in a one-class problem there is only a single class of interest – which we denote as class one. Patterns either belong to class one, or belong to no class at all. Of course, many interesting real-world learning problems incorporate multiple classes. This section presents an SMP-based architecture capable of solving multi-class learning problems.

We can create the multi-class SMP network by simply incorporating more neurons, with one dedicated perceptron for each class. A depiction of one possible SMP architecture for a three-class problem is shown in Fig. 3. It has three SMP neurons – each one dedicated to learning a particular class. In essence, this approach solves the multi-class learning problem by transforming it into a number of one-class learning problems, and assigning each one-class learning problem to a dedicated SMP neuron. We train the multi-class network by simply training each neuron to solve its particular one-class problem. The synaptic connections of the different neurons are independent of each other, so the neurons can be trained in any order.

When working with multi-class problems, using a fuzzy classifier may provide greater classification accuracy than simply relying on the crisp outputs of each neuron. In the fuzzy classifier, the output values of all neurons are recorded and compared. The input pattern is assigned to the class corresponding to the neuron with the largest output. This allows for a definite classification in cases where no output neuron achieves a positive value, or where multiple output neurons produce positive values. On the other hand, the fuzzy classifier cannot be used in cases where patterns can belong to multiple classes, or where outlying patterns should not be assigned to any class.

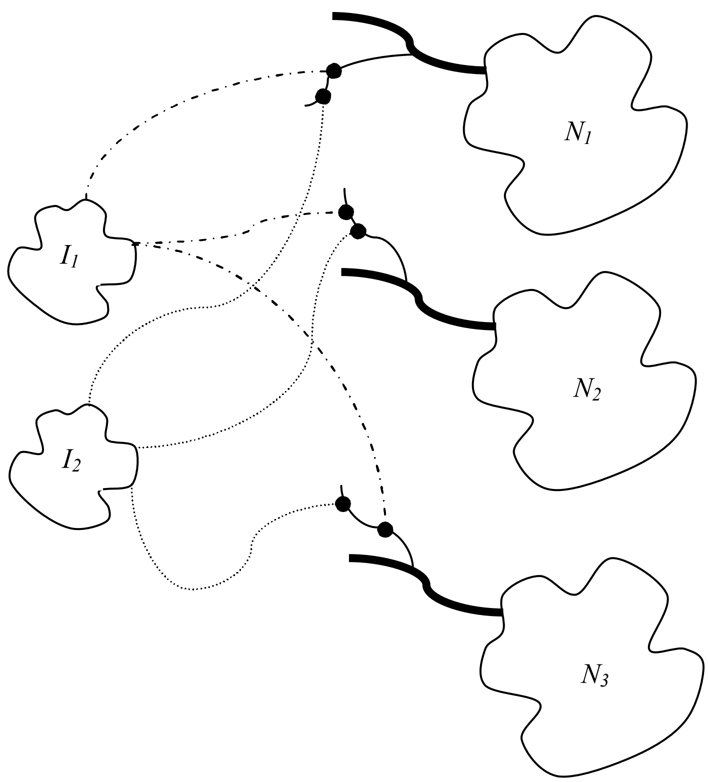

Fig. 3. An example of a multi-class SMP network with two inputs and three classes. Each of the three neurons, N_1, N_2, and N_3 is tasked with learning one of the three classes. Each of the neurons has one dendrite with one synapse.

6. EXPERIMENTAL RESULTS

This section presents the results of two experimental tests on the synaptic morphological perceptron. The first test – a pair of embedded spirals – will be used to highlight some general properties related to the neuron's learning capabilities. The second test – Fisher's iris data set – applies the multi-class SMP architecture of Section 5 to a common benchmark test.

6.1 Embedded Spirals

The data set for the first test consists of the two Archimedean spirals shown in Fig. 4. One of the spirals, denoted S_1, comprises the class 1 patterns that the neuron must learn, and has Cartesian coordinates given by the parametric expressions

$$x_1(\theta) = \left(\frac{2}{\pi}\right)\theta \cos\theta \tag{16}$$

$$y_1(\theta) = \left(\frac{8}{3\pi}\right)\theta \sin\theta \ , \tag{17}$$

where the parameter θ is sampled in the interval $[\pi/2, 4\pi)$ [LATTICE APPROACH]. The second spiral S_2 consists of the class 0 patterns and has coordinates given by

$$x_2(\theta) = -x_1(\theta) \tag{18}$$

$$y_2(\theta) = -y_1(\theta). \tag{19}$$

Thus, the two spirals are symmetric about the origin.

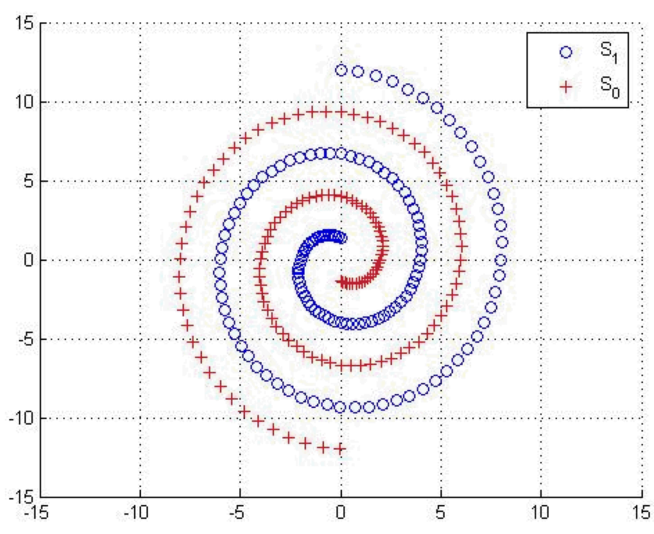

Fig. 4. The entwined Archimedean spirals. There are 128 points in each spiral.

For this test, a data set of 256 points – 128 from each spiral – was generated and used to train a single SMP neuron with ten dendrites and four synapses per dendrite. The learning rate η was initially set to .15, and then decreased by .00015 each training epoch. The neuron achieved 100% correct classification on the training set after 192 epochs.

Recall that each set of synaptic weights defines a hyperplane, and thus splits $\Re^N$ into positive and negative halfspaces. Each dendrite defines a region that is the intersection of the positive halfspaces of each of its synaptic hyperplanes. The neuron was initialized with ten dendrites, and four synapses on each dendrite. During training, however, many of the synaptic connections become irrelevant and are automatically "pruned" by the learning algorithm. We can track the importance of each synapse using a *synaptic counting matrix*. Each row of the matrix represents one dendrite, and each entry in a row is a synapse on that dendrite. When a synapse is selected as the "winner" during classification, its entry in the matrix incremented by one. Thus, when the neuron is fully trained, the synaptic counting matrix identifies the relative importance of each synaptic hyperplane in the final classifier. Table 1 shows the final synaptic counting matrix for the trained embedded spirals neuron. Entries with a value of zero were never selected as the winning synapse; thus, they have effectively been pruned out of the network by the training algorithm. The synaptic counting matrix reveals some interesting information about the neuron. For example, we see that dendrites six and eight have been totally removed, and are not being used to classify any patterns in the data set.

Fig. 5 shows a plot of the mean-squared error vs. training epoch for this test. The plot is extremely jagged, though it trends downwards and eventually reaches zero. This illustrates an important property of the SMP. Recall that only one set of synaptic weights is updated when a pattern is misclassified during training. Traditional back-propagation algorithms use a gradient-descent method to adjust all the weights of the network at each step, thereby decreasing the overall error. The SMP error curve is not monotonic, and frequently increases before making a larger corresponding decrease.

Table. 1. Synaptic counting matrix for the final embedded spirals neuron.

Dendrite	*Selection counts per synapse*			
1	8	22	9	0
2	1	17	5	25
3	6	0	6	19
4	22	0	0	0
5	0	0	19	0
6	0	0	0	0
7	4	14	16	0
8	0	0	0	0
9	0	8	17	15
10	10	0	15	0

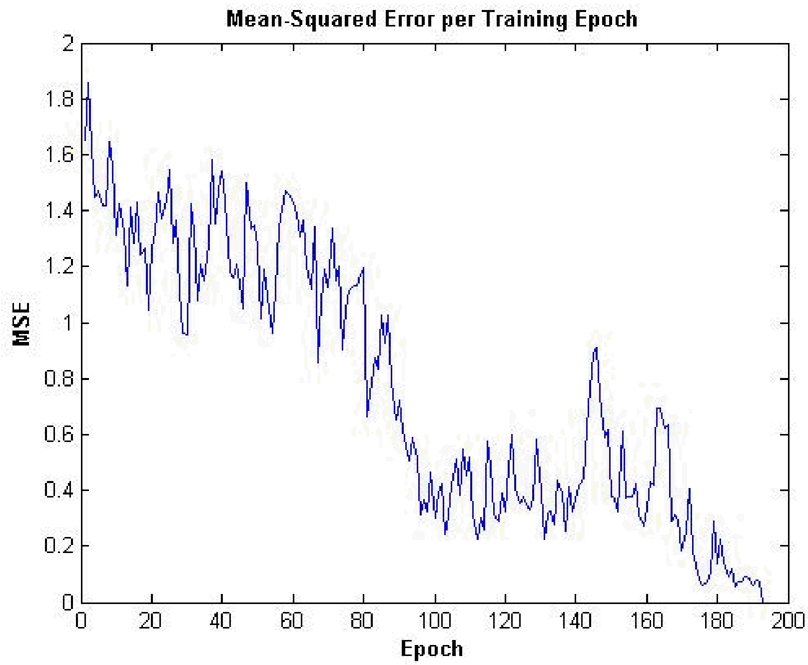

Fig. 5. Mean-squared error per training epoch collected while training the entwined spirals neuron. The extremely non-monotonic error curve is characteristic of SMP training.

6.2 Fisher's Iris Data Set

Fisher's iris data set is a frequently used benchmark for multi-class learning techniques. The set consists of 150 four-dimensional patterns, divided equally among three classes. Each class corresponds to a species of iris flower, and the four measurements correspond to sepal length, sepal width, petal length, and petal width of each iris specimen. The first class – the species *iris setosa* – is linearly separable from the other two classes. However, classes two and three – the species *iris versicolor* and *iris virginica* – cannot be linearly separated. Thus, the iris data set allows us to explore a multi-class problem where the boundaries between classes are nonlinear.

For these experiments, we trained a multi-class network of three SMP neurons as described in section 5. For each experimental trial, we randomly selected 38 patterns from each class – this is approximately 75% of the total number of patterns in each class – and used this training set to adjust the synaptic weights for each of the three neurons in the mutli-class network. During the training, the learning rate η was initially set to .01 and decreased by .0002 each training epoch. Training ended after the network achieved 100% correct classification on the training set, or after a maximum of 500 epochs. After training, the network was asked to classify each of the 150 patterns in the complete iris data set using the fuzzy classification technique, and the number of misclassified patterns was recorded. For this test, we used the fuzzy classification technique described in Section 5.

We repeated this process using training sets consisting of 25 and 13 randomly selected patterns from each class – corresponding to approximately 50% and 25% of the total number of patterns in each class. Table 2 summarizes the results of our thirty trials, ten for each training set size.

The results show that the SMP network generally performs quite well, even with only 13 training patterns from each class. Misclassifying an average of 3.8 out of 150 patterns corresponds to a classification accuracy of 97.466%, and misclassifying an average of 5.6 patterns corresponds to an accuracy of 96.266%. Thus, the network is still able to effectively discriminate between the three iris classes, even for relatively small training sets.

Table. 2. Number of misclassified patterns for each experimental trial with Fisher's iris data set.

Training patterns per class	*Misclassified patterns per trial*										*Mean*
38	4	4	2	3	3	4	7	4	3	4	3.8
25	2	4	4	4	11	4	10	4	3	6	5.2
13	6	5	4	3	8	5	7	4	7	7	5.6

7. CONCLUSION

The synaptic morphological perceptron represents a new approach to learning and classification in morphological neuron networks. The SMP can learn any region of space bounded by arbitrary hyperplanes, and is trained using an error-backpropagation method, similar to traditional neural network architectures.

There are still several opportunities for futher research on the SMP. For example, our experiments used a predetermined configuration of dendrites and synapses on each neuron. Ideally, the SMP training algorithm should have a way to dynamically adjust its own configuration, so that the final neural configuration is optimal in some sense. Ritter has recently shown how hyperbox-based SLMP architectures can be used to construct robust autoassociative memories [BIG THICK RITTER PAPER], and the SMP may have applications in this domain as well. Future work with the synaptic morphological perceptron will focus on these topics as well as extensions into areas of image processing, target recognition, and unsupervised learning.

REFERENCES

1. A. Eisenberg, *Guide to Technical Editing*, Oxford University, New York, 1992.
2. N. Bluzer and A. S. Jensen, "Current readout of infrared detectors," *Opt. Eng*. 26(3), 241-248 (1987).
3. C. Jones (private communication).
4. J. Rivers, http://awebsiteref.com