Classification Characteristics of SOM and ART2

J.J. Aleshunas
U.S. Army Reserve Personnel Ctr.

Daniel C. St. Clair
University of Missouri-Rolla

W.E. Bond
University of Missouri-Rolla

Keywords: Neural Network, SOM, ART2, Unsupervised Learning

ABSTRACT

Artificial neural network algorithms were originally designed to model human neural activities. They attempt to recreate the processes involved in such activities as learning, short term memory, and long term memory. Two widely used unsupervised artificial neural network algorithms are the Self-Organizing Map (SOM) and Adaptive Resonance Theory (ART2). Each was designed to simulate a particular biological neural activity. Both can be used as unsupervised data classifiers.

This paper compares performance characteristics of two unsupervised artificial neural network architectures; the SOM and the ART2 networks. The primary factors analyzed were classification accuracy, sensitivity to data noise, and sensitivity of the algorithm control parameters. Guidelines are developed for algorithm selection.

Introduction

The training of artificial neural networks can be separated into two main categories: supervised and unsupervised. Supervised training requires prior knowledge of what the network is expected to do. The network must be trained to map the exemplars of the training set to known outcomes. Supervised network training can monitor the convergence of the network toward the expected outcome and use it as a criterion to stop training. Unfortunately, because the training is focused on the expected outcome, unexpected possibilities may be either incorrectly mapped or are rejected as noise. The prior knowledge associated with the training set creates a classification bias in the network.

Unsupervised training requires no prior knowledge of the problem domain. The network groups exemplars in the data set with other exemplars having similar characteristics. Competitive training is the procedure normally used to control training in unsupervised machine learning algorithms. In competitive training, the output node with the "best" or maximal output is selected for training. Other nodes may receive reduced training or no training at all. This training algorithm reinforces dominant patterns in the training data and therefore adapts the resulting network to these patterns.

Because prior knowledge is not used in the training of the network, bias toward specific expectations is not introduced. This does not mean that no bias exists. A critical assumption is that the training data set is representative of the problem domain. When the training data is not totally representative of the problem domain, the network's classification accuracy is affected. Also, since there is no expected outcome, stopping criterion other than the convergence of the network results must be chosen. Typically, the algorithm is trained for a predetermined number of epochs. This criterion can sometimes allow the network to overtrain.

This paper focuses on comparing the performance characteristics of two unsupervised artificial neural network architectures; the Self-Organizing Map (SOM) and the Adaptive Resonance Theory (ART2) networks. The SOM maps vectors from an n-dimensional space to a two-dimensional output network. The ART2 algorithm maps vectors from an n-dimensional space to a one-dimensional output network. The primary factors analyzed were classification accuracy, sensitivity to data noise, and sensitivity of the algorithm control parameters. Accuracy was determined by comparing algorithm classifications to the expected classifications.

Self-Organizing Map

Many neurological systems exhibit a self-organizing inclination. Nerves in the human auditory system are organized so that neighboring nodes respond to similar sound frequencies. This spatial distribution inspired Teuvo Kohonen to develop his Self-Organizing Map [6]. The SOM is a competitive unsupervised learning algorithm. Mutual lateral interactions are developed between the output nodes by training neighborhoods of nodes to respond to an input vector. The size of these training neighborhoods linearly decreases over a training session. The result is a trained network where neighboring nodes share similar properties and distant nodes are obviously different. This property is what allows SOM to organize and group input data over its two-dimensional output surface.

The SOM defines a map

$$\mathbb{R}^n \longrightarrow \mathbb{R}^2 \qquad (1)$$

This mapping can also be defined as

$$f : p(x_1, x_2, \ldots, x_n) \longrightarrow (a,b) \quad (2)$$

where f is a non-linear mapping of the probability distribution of an n-dimensional space to a two dimensional space.

The normal activation of the nodes in a SOM network is the main principle governing the SOM learning algorithm. This activation is usually evaluated using the dot product of the input vector, x, with the weight vectors, m_i (i = 1, ... , n) of the connecting network.

$$x \cdot m_i = \|x\| \|m_i\| \cos\theta \quad (3)$$

The value $\|x\|$ is the Euclidean magnitude of the vector x, m_i is the weight vector associated with a given output node (a,b), and θ is the angle between the vectors x and m_i. These weight vectors are randomly initialized within the bounds of the maximum and minimum values of the training input vectors. Kohonen referred to these weight vectors as the codebook vectors since their final values would define the encoded mapping. The maximum activated node is selected for training. A node is maximal when the value of its dot product is maximal. The dot product increases to its maximum as the angle between the input vector and the weight vector approaches zero. Therefore, minimizing the angle or distance between the input vector and a given weight vector maximizes the activation output. This comparison involves calculating the Euclidean distance between input vector and the weight vectors and choosing a weight vector m_c such that

$$\|x - m_c\| = \min_i \{\|x - m_i\|\} \quad (4)$$

for a given input vector x and all weight vectors m_i.

Once the maximum output node, $(a, b)_c$, is selected, the corresponding weight vector, m_c, associated with this node is adjusted to minimize the distance between m_c and the input vector. A learning rate factor, $h_{ci}(t)$ is used to control the amount of adjustment a particular weight vector receives. The updated value of m at time t is computed as

$$m_i(t) = m_i(t-1) + h_{ci}(t)\{x(t-1) - m_i(t-1)\} \quad (5)$$

The SOM trains nodes in a neighborhood, N_c surrounding an activated node, $(a, b)_c$ to react to similar input stimuli. This feature of the SOM provides the self-organizing property of the algorithm. The radius of each training neighborhood decreases linearly with time during the training session. In later training epochs, the training radius decreases until only the activated node is trained. The result is a trained network where neighboring nodes share similar properties as shown in Equation 6.

$$\|m_c - m_i\| < \|m_c - m_{i-n}\| \quad \text{for } n>0 \quad (6)$$

Two types of neighborhoods are used by SOM. The "Bubble" type neighborhood defined by

$$h_{ci}(t) = \begin{cases} \alpha(t) & i \in N_c \\ 0 & i \notin N_c \end{cases} \quad (7)$$

where $h_{ci}(t)$ is the amount of training that node i in N_{cr} receives at time t. This method applies a constant training factor to all nodes within the neighborhood of the selected node. Figure 1 shows this situation.

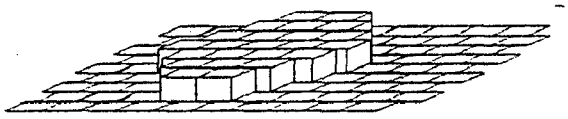

Figure 1 Bubble neighborhood

The "Gaussian" type neighborhood is defined by

$$h_{ci} = \alpha(t) \cdot \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right) \quad (8)$$

where $h_{ci}(t)$ is the amount of training that node i in N_{cr} receives at time t and $\sigma(t)$ defines the width of the neighborhood. This method applies a training factor that non-linearly decreases as the distance from the selected node increases. Figure 2 shows this situation.

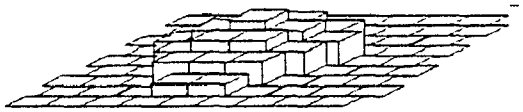

Figure 2 Gaussian neighborhood

The SOM provides for the use of two possible array training topologies, rectangular and hexagonal. In the rectangular topology, nodes are arranged in rows and columns. Each node, m_c, can interact with the eight nodes surrounding it. All distances are measured by simple comparisons of the node's X and Y coordinates. Neighborhoods in this topology are rectangular regions surrounding the node m_c. Figure 3 illustrates this topology.

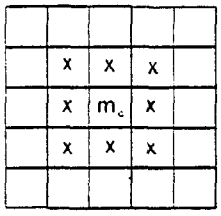

Figure 3 Rectangular topology

The hexagonal topology allows for radial node interaction. Starting with the second row, every other row of nodes is shifted half a node's width to the right. This shift changes the topology so that each interior node is bounded by six, rather than eight nodes. Therefore, a node, m_c, can interact with the two nodes above, the two nodes below, the node to the right, and the node to the left. Neighborhoods in this topology are hexagonal regions surrounding node m_c. Figure 4 illustrates this topology.

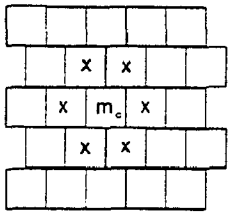

Figure 4 Hexagonal topology

The SOM algorithm consists of (for each training vector):

1) Determine the radius for training N_{cr}.
2) For vector, x, select maximal node, m_c.
3) Calculate $\|x - m_i\|$.
4) Train all nodes within neighborhood N_{cr} of m_c.
5) Repeat steps 1 through 4.

A training epoch applies this sequence once to each vector in the training set. Training continues until the predetermined maximum number of training epochs is reached.

Adaptive Resonance Theory

One of the intriguing features of human memory is the ability to learn new things while retaining previously learned information. Unfortunately, most artificial neural networks require retraining using the complete set of exemplars if new information is added.

The problem of adding new information to the memory of an already trained network describes what Stephen Grossberg [2] calls the *stability-plasticity* dilemma. The dilemma can be stated as a series of questions: How can a learning system remain adaptive (plastic) in response to significant input, yet remain stable in response to irrelevant input? How does the system know to switch between its plastic and its stable modes? How can the system retain previously learned information while continuing to learn new things?

Carpenter and Grossberg [2] designed the ART2 architecture with these considerations in mind. The key to solving the stability-plasticity dilemma is to add a feedback mechanism between the competitive classification layer (F_2) and the input layer (F_1) of a network (see Figure 5). A pattern is entered on the input nodes and fed forward to the classification nodes through a weighted network. Once a node is selected in the competitive classification layer, then results are fed back to the input layer through another weighted network. If the feedback information matches that at the input layer, then this classification is considered a match and the feedforward and feedback networks are trained for this pattern. If the feedback does not produce a match, then the selected classification node is eliminated from further active competition and the input is fed forward again without providing any training to the feedforward or the feedback networks. An input pattern may be matched to previously learned pattern classifications or it may be form a new classification by itself.

This feedback mechanism facilitates the learning of new information without destroying old information. The network quickly searches for an appropriate output classification and, only when a match is found, is any training applied to the network weights reinforcing this classification. Previous training, supporting other pattern classifications, is not affected by this new training.

A series of differential equations govern the activities of the individual processing elements. To deal successfully with analog input patterns, they split the F_1 input layer into a number of sublayers containing both feedforward and feedback connections. Figure 5 shows the resulting structure. Carpenter and Grossberg [2] provide detailed instructions for calculating the values shown in Figure 5.

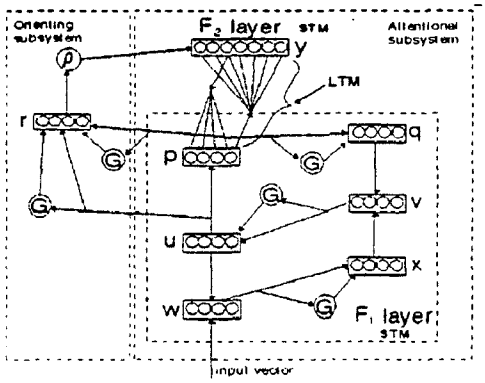

Figure 5 ART2 topology [5]

Patterns of activity that develop within the nodes in the F_1 and F_2 layers of the attentional subsystem are called short-term memory (STM) traces because they exist only in association with a single application of an input vector. The weights associated with the bottom-up and top-down connections between F_1 and F_2 are called long-term memory (LTM) traces because they encode information that remains a part of the network for an extended period.

Evaluation of SOM and ART2

Two data sets were used in this evaluation. The first set is the Iris data set [3] which is a good set containing noise. The second set is a generated set devised by Breiman [1]. The Breiman approach allows one to control the amount of noise present in the data. Three specific areas of performance; classification accuracy, sensitivity to noise in the data, and

sensitivity to adjustments in the algorithm parameters, were measured.

The Iris data set consists of measurements from 150 iris flowers. The measurements are part of the Anderson iris data set made famous by Fisher [4]. The four measurements are sepal width, sepal length, petal width, and petal length for each flower. The sample contains 50 flowers from each of three varieties, *iris setosa*, *iris virginica*, and *iris versicolor*. This data can be regarded as 150 four dimensional vectors in R^4. An additional variable, the iris variety, is associated with each four dimensional vector. This can be used for labeling and classification verification.

The Breiman data set consists of 300 vectors in R^{21} representing three generated waveform classes (100 samples for each waveform). Three simple waveforms, $h_1(t)$, $h_2(t)$, and $h_3(t)$ are the basis for the generated classes. Figure 6 shows a graph of $h_1(t)$.

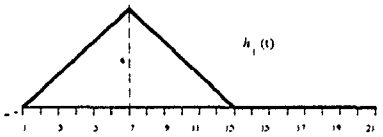

Figure 6 $h_1(t)$ waveform

The $h_2(t)$, and $h_3(t)$ waveforms are similar except they peak at 15 and 11 respectively.

Each class consists of a random convex combination of two of these waveforms sampled at 21 intervals with noise added. To generate a class 1 vector **x**, a single uniform (0,1) random number u and 21 N(0, σ^2) random numbers e_1, ..., e_{21} were generated. Each component x_t of vector **x** was generated by

$$x_t = uh_1(t) + (1-u)h_2(t) + e_t, \quad t = 1,...,21 \qquad (9)$$

Class 2 vectors are generated form $h_1(t)$, and $h_3(t)$. Class 3 vectors are generated from $h_2(t)$, and $h_3(t)$.

For training and testing, a method called v-fold cross-validation was used [1]. The data set L is randomly divided into V subsets of equal size. Then for every v, $v = 1, ..., V$, use $L - L_v$ as the training set and L_v as the test set. The resulting V classifications of the training sets will approximate the classification of L and the accuracy of the L_v test sets will approximate the accuracy of L.

Classification accuracy was determined by comparing the algorithm classifications to the expected classifications. For these experiments, when ART2 grouped a vector with other patterns whose classification did not match the vector's expected classification then this output mapping was called an error. This concept was used in evaluating the accuracy of the SOM also. Unlabeled nodes were treated as belonging to a labeled region. If a vector mapped into an unlabeled node in a region and the vector's classification matched that of the region, it was considered a correct mapping. If the mapping occurred on an unlabeled node in a border between two regions, and the vector's classification matched one of the regional classifications, this was treated as a completion of the region's classification into the neutral border nodes and was treated as a correct mapping. Only when a vector mapped into a node with a label different than the vector's classification or when a vector mapped into an unlabeled node within a region whose label did not match the vector's classification was the test mapping considered erroneous.

The Iris data set was presented to two SOM networks; a 10 x 10 output array and a 5 x 5 output array. The primary difference between the final trained networks was the increased number of unlabeled nodes in the larger array. The accuracy results for 4-fold cross-validation on both networks is shown in Table 1. Each SOM was trained for 11,000 epochs.

Overall, the total performance of the two mappings was consistent but the increased space of the 10 x 10 array and the assumption of regional accuracy helped the larger array to perform better. The main tradeoff in choosing a larger SOM network is that the training time is increased. Since the computations are mathematically simple, this is not a severe tradeoff.

The ART2 network quickly trained on the 4-fold Iris partitions. Three of the four partitions reached their maximum accuracy in less than ten training epochs. The fourth took fifteen epochs. This is significant since ART2 is computationally more complex than SOM. The accuracy for the 4-fold partitions and the overall average is shown in Table 1. In these experiments, ART2 outperformed SOM. In addition, the variance of the results for the four test sets was smaller for ART2.

v-fold	IR1	IR2	IR3	IR4	Average	σ
SOM 5X5	86.5%	100.0%	94.6%	94.7%	93.95%	4.82
SOM 10X10	89.2%	94.7%	97.3%	97.4%	94.65%	3.33
ART2	97.3%	94.7%	100.0%	97.4%	97.35%	1.87

Table 1 SOM and ART2 Iris Accuracy

The Iris data was a good initial data set. The parameters of the SOM were not highly sensitive and tolerated quite a bit of change without much impact on the training results. The primary effect of most parameter changes was the amount of time the network took to train. More complex networks (array size, number of training epochs, initial size of training neighborhoods, etc.) took longer to train.

u \ σ^2	1.0	2.0	3.0
0.833	80.3 (5.10)	74.3 (5.44)	72.3 (7.26)
0.667	97.7 (0.78)	85.0 (2.77)	84.7 (2.20)
0.5	99.3 (0.83)	94.3 (2.26)	90.7 (3.11)

Table 2 Average SOM Breiman Accuracy

u \ σ^2	1.0	2.0	3.0
0.833	69.0 (5.02)	71.3 (4.14)	73.7 (3.56)
0.667	92.7 (1.31)	82.7 (1.69)	79.0 (3.08)
0.5	97.7 (1.32)	91.0 (2.70)	82.7 (2.72)

Table 3 Average ART2 Breiman Accuracy

The primary differences between SOM and ART2 involved the final representation of the classified data. The ART2 algorithm creates a set of data pattern classes. The starting set of output nodes is trimmed down to only those nodes that have some classification mapped onto them. A vector can either be correctly mapped to a exact classification node or incorrectly mapped to another node. There is little relationship between the nodes other than the classification of the nodes.

The SOM algorithm, on the other hand, imposes a spatial relationship on its two dimensional output array. If two vectors map into adjacent nodes a similarity of structure is implied. These vectors are more alike than vectors that map into more distant nodes. This relationship is not mathematically exact. For instance, it can not be said that two vectors, one node apart, are twice as similar as two vectors, two nodes apart. The only implication is that the vectors, one node apart, are more similar than those two nodes apart. All nodes are retained in the final network. The output mappings are good graphic depictions of the data and the relationships between the data vectors. Where definite separations exist, such as between the setosa and the other two cultivars, a clear boundary shows up on the map. Where noise blurs the distinctions between classes, such as between virginica and versicolor, the mapping shows vague, intermixed boundaries. This result creates classification regions on the output node array. This regionality attribute of SOM allows unclassified nodes to be included in the region with classified nodes.

The strength of ART2 is its ability to match patterns accurately, in few epochs, and later be modified to include additional information without losing formerly trained classifications. The increased computational complexity was offset by a powerful algorithm which needed few epochs to train. The classifications were clear and distinct with a solid grouping accuracy. The parameters associated with ART2 were more sensitive to adjustment. The ρ parameter was especially sensitive to small changes of its value and some experimentation was necessary to find a good value for the data used. Experimentation was used to choose the other parameters in this research so that they were accurate, efficient, and comparable between the two algorithms, though not necessarily optimal.

The Breiman data was generated using three specific uniform numbers, 0.5, 0.6667, and 0.8333, and three noise variances, 1.0, 2.0, and 3.0 resulting in nine data sets with known noise levels. This data generation plan introduced two types of classification noise to test the algorithms. First, as the uniform number, u, approaches one, the class 1 and the class 2 base waveforms become difficult to differentiate. Next, as the noise variance, σ^2, increases, it may overpower any minor base waveform differences. Each data set was partitioned into 5-fold cross-validation sets for training and testing.

The SOM algorithm was trained and tested using a 10 x 10 output array configuration based on the Iris dataset results. The average classification accuracy and standard deviation are shown in Table 2. As the variance σ^2 increases, the accuracy decreases. The classification confusion induced as u approaches one also reduces the classification accuracy, as would be expected.

The ART2 algorithm was trained and tested using the same Breiman datasets. The accuracy and standard deviation of the nine 5-fold sets is shown in Table 3. The algorithm's accuracy declines as the noise variance increases. It also has difficulty as u approaches one and the class 1 and class 2 waveforms become less distinct. Figure 7 graphs the accuracy of the datasets for each given value of u as the noise variance increases. The Breiman data was a good test for these algorithms since it was a generated dataset and its parameters could be controlled. The SOM algorithm did not have any real problem classifying each 5-fold dataset. The classification accuracy decreased as the variance of the noise was increased for each given value of u. As the value of u approached one, the class 1 and the class 2 waveforms became hard to distinguish and the accuracy dropped accordingly.

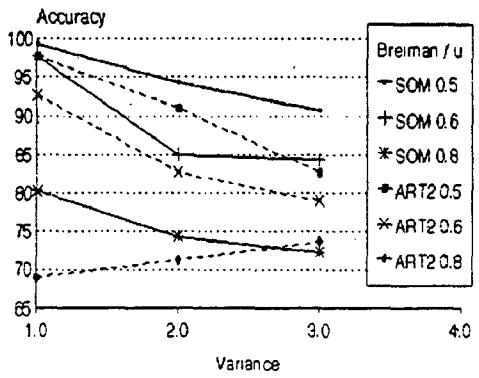

Figure 7 Accuracy vs. Noise for Breiman Data

The ART2 algorithm did not perform as well as the SOM on the Breiman data. It showed a sensitivity to noise as the variance increased when u equaled 0.5 and 0.667. This accuracy decline was not present when u equaled 0.833. The standard deviation values for this last plot indicate a fairly level range. This change in accuracy indicates that the confusion between the class 1 and the class 2 waveforms has more effect on ART2 than the noise once the value of u gets large enough.

Conclusions

The SOM algorithm is computationally simple. It is straight forward to find a set of parameters that produce reasonable results. The two-dimensional relationship of the SOM output nodes conveys additional information beyond the strict classification of a particular input vector. Based on the proximity of an activated output node to a classification region, one can infer a certain amount of similarity of the given input vector to vectors of that classification. The regionality associated with the classification nodes contributes to the good classification accuracy of this algorithm. Therefore SOM is a good tool to do quick analysis of multi-variate data because it is accurate, reasonably tolerant of noise, and allows easy parameter initialization.

The ART2 algorithm's strength is its ability to learn a new pattern without having to retrain on all of the already known patterns. This capability adds to the algorithm's computational complexity and training time. The network parameters are a bit more sensitive to adjustment than those in SOM. The vigilance parameter, ρ, seemed to be the most sensitive. This parameter is a measure of how well a classification matches the input pattern and how the plasticity-stability feedback mechanism is controlled. A little experimentation with several values will produce an accuracy versus ρ value curve from which a good value can be determined. As the base waveforms became less distinguishable the algorithm's classification accuracy was dominated by the waveform classification error rather than the noise. This indicates that there is a limit to how well ART2 can differentiate subtle input pattern differences for a particular value of ρ.

REFERENCES

1. Breiman, Leo, Friedman, Jerome H., Olshen, Richard A., and Stone , Charles J., *Classification and Regression Trees*, (1984), Wadsworth & Brooks, Pacific Grove, California, pp. 9-13, pp. 49-55.

2. Carpenter, Gail A. and Grossberg, Stephen, *ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns*, (1 December 1987), Applied Optics, Vol. 26, No. 23, pp. 4919-4930.

3. Chambers, John M., Cleveland, William S., Kleiner, Beat, and Tukey, Paul A., *Graphical Methods for Data Analysis*, (1983), Wadsworth & Brooks, Pacific Grove, California, pp. 82-87, 106-109, 130, 365-366.

4. Fisher, R.A., *The Use of Multiple Measurements in Taxonomic Problems*, Annals of Eugenics 7, pp. 179-188.

5. Freeman, James A. and Skapura, David M., *Neural Networks Algorithms, Applications, and Programming Techniques*, (1992), Addison-Wesley, New York, New York, pp. 263-339.

6. Kohonen, Teuvo, *The Self-Organizing Map*, (September 1990), Proceedings of the IEEE, Vol. 78, No. 9, pp. 1464-1480.

7. Kohonen, Teuvo, Kangas, Jari, and Laaksonen, Jorma, *SOM_PAK The Sale-Organizing Map Programming Package*, Version 1.1, (14 October 1992), Laboratory of Computer and Information Science, Helsinki University of Technology, Finland.