

# An Experimental Comparison of Semi-Supervised ARTMAP Architectures, GCS and GNG Classifiers

Quang Le  
Computer Sciences<sup>1</sup>  
ql@fit.edu

Georgios C. Anagnostopoulos  
Electrical & Computer Eng.<sup>1</sup>  
georgio@fit.edu

Michael Georgiopoulos  
Electrical Eng.<sup>2</sup>  
michaelg@mail.ucf.edu

Ken Ports  
Electrical & Computer Eng.<sup>1</sup>  
kports@fit.edu

<sup>1</sup>Florida Institute of Technology, Melbourne, FL 32901

<sup>2</sup>University of Central Florida, Orlando, FL 32816

**Abstract** – In this paper we present an experimental comparison of four neural-based classifiers, namely Growing Cell Structures (GCS), Growing Neural Gas (GNG), Semi-Supervised Fuzzy ARTMAP (ssFAM) and Semi-Supervised Ellipsoid ARTMAP (ssEAM). The comparison is performed in terms of classification accuracy and structural complexity of the resulting classifiers. Earlier studies that had appeared in the literature showed that Fuzzy ARTMAP, which utilizes fully-supervised learning, may suffer from poor generalization performance, when compared to GCS and GNG classifiers. This phenomenon typically occurs, when class distribution overlap is significant. Here, we present new results indicating that ARTMAP classifiers equipped with semi-supervised learning capabilities can improve their performance with respect to GCS and GNG classifiers, while maintaining lower structural complexity.

## I. INTRODUCTION

Neural networks have been successfully applied in a plethora of application domains as classification models. Among these, a particularly successful group is the family of *incremental neural networks*. The three main characteristics of these networks are (i) their ability to improve their classification performance by gradually increasing their structural complexity during training on an as-needed basis, (ii) their ability to perform *online (incremental) learning* and (iii) their *response transparency*. In the context of classification, incremental learning refers to the ability of these networks to learn how to correctly classify individual patterns on a one-by-one basis in the order they become available for learning. During this process only a small part of the network has to undergo appropriate changes. These updates are performed via local adaptation rules and typically only a small number of synaptic weights are affected per presented pattern. Also, by response transparency we mean our ability to easily explain a network's response to a specific input. This is a highly desirable feature in many real applications, where assurance is needed about the validity of the model's classification results.

These characteristics contrast other neural architectures, such as Multi-Layer Perceptrons [1], [2] among others, whose

structure and topology must be defined prior to the training process. Furthermore, learning a single pattern typically involves updating a large proportion (if not all) of the network's synaptic weights. Moreover, since the network's knowledge is highly distributed over all synaptic weights due to the utilization of global adaptation rules, it is difficult, in general, to explain the network's outputs.

Incremental neural networks can also be thought of belonging to the broader family of *exemplar-based classifiers*, which cluster the training patterns via the use of exemplars or prototypes prior or simultaneously to predicting class labels. These exemplars may have different geometrical representations, such as single points (vectors), hyper-spheres and other constructs, whose role is to represent the existence and influence of training patterns in the feature space. From that aspect the set of exemplars substitutes the role of the training set. After clustering the resulting number of exemplars is far less than the size of the training set and, therefore, the collection of exemplars can be thought of as a compressed representation of the knowledge implied by the training data.

This paper will focus on four types of incremental neural architectures: Growing Cell Structures (GCS), Growing Neural Gas (GNG), Semi-Supervised Fuzzy ARTMAP (ssFAM) and Semi-Supervised Ellipsoidal ARTMAP (ssEAM) classifiers. GCS is a neural network architecture consisting of a Self-Organizing Feature Map-type layer [3] (hidden layer) followed by a Radial Basis Function output layer [4]. Neighborhoods of hidden neurons are defined with the aid of k-dimensional topological structures (k-dimensional *simplexes*) that relate the neurons with each other [5]. The GNG [6] architecture is a modification to the GCS family, in which the dimensionality of the topological structures is not predefined, but is discovered during training. Both ssFAM and ssEAM neural networks have their roots in *Adaptive Resonance Theory* first proposed in [7] and are relatively recent extensions (although in slightly different aspects) of the basic Fuzzy ARTMAP (FAM) architecture [8]. While FAM and ssFAM employ hyper-rectangles as exemplars [9], ssEAM uses hyper-ellipsoids to summarize input data [10].

ssFAM and ssEAM make use of *semi-supervised learning* [11] during their training phase that allows these networks to feature zero post-training error, while preserving finite-time learning.

An experimental comparison of GCS, GNG, MLP and FAM networks presented in [12] suggested that FAM models, although having the fastest training time and the lowest, final network size, have the tendency of performing poorly, when tackling classification problems that feature significant class distribution overlap as well as non-linear, decision boundaries. The later phenomenon was indirectly attributed to the hyper-rectangular exemplars used by FAM networks, which induce piece-wise linear decision boundaries. The objective of our experimental comparison is to show that ARTMAP classifiers making use of semi-supervised learning, such as ssFAM and ssEAM, achieve generalization performance comparable to the one of GCS and GNG models, while maintaining lower structural complexity and shorter training times. Towards this goal we draw a new comparison of GCS, GNG, ssFAM and ssEAM on four classification problems of varying degree of difficulty.

The rest of the paper is organized as follows. In Section II we provide more details about the main characteristics of the architectures we compare, whereas Section III describes the data sets that were used, our experimental settings and the results we obtained. Finally, Section IV summarizes our findings and conclusions.

## II. BRIEF DESCRIPTION OF ARCHITECTURES

### A. GCS & GNG Networks

As mentioned earlier in the text, GCS networks consist of three layers (input, hidden and output layers), the hidden layer being a special, self-organizing map and the output layer being a radial basis function layer. Unlike the traditional self-organizing maps, neighborhoods of hidden nodes in GCS are being defined with the aid of a graph embedded in a  $k$ -dimensional space, where  $k$  is a network parameter. Each node of the graph corresponds to a hidden node and neighborhood relations between two hidden nodes are depicted via edges connecting the corresponding graph nodes. The purpose of the hidden layer is to perform clustering of the input data and the exemplars/prototypes used are the input-to-hidden layer synaptic weight vectors. Appropriate updates performed on the graph by inserting and deleting nodes allows neighborhoods to change dynamically in an incremental, online fashion and capture the characteristics of the underlying data distribution. Insertion and deletion of nodes occurs in such a manner so that the graph consists of  $(k+1)$ -dimensional simplexes at all times. This process is depicted in Figures 1(a) and 1(b). Typically, insertions happen every  $\lambda$  presentations of patterns and node deletions after the hidden layer has reached a prescribed size. The maximum size a hidden layer can reach constitutes an additional training parameter for the GCS network. Upon

presentation of a training pattern, the hidden node featuring the highest activation function value (the winning node) and its immediate neighbors (according to the graph) are updated using a learning rate of  $\epsilon_b$  and  $\epsilon_n$  respectively. The weights from the hidden to output layer are updated with a learning rate of  $\eta$ . The main reason behind node insertion is to make the selection of winning nodes approximately equiprobable. We must note here that the hidden units feature Gaussian activation functions and that all the node updates occur concurrently. Finally, the activation signals of the output layer act as discriminant values used to determine the final classification label of a pattern.

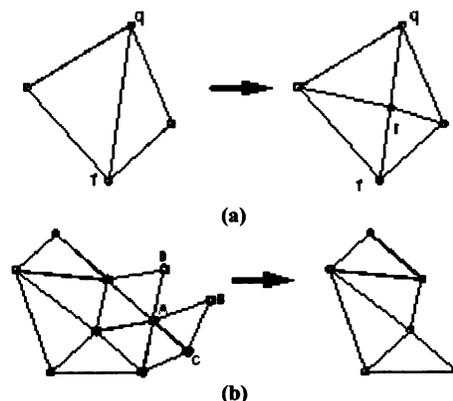


Fig. 1. Graph modification (a) after inserting a hidden node and (b) after removing a hidden node in a GCS network.

The GNG network is a variation to the GCS that is designed to overcome, primarily, the difficulty of pre-assigning a particular dimensionality to the space, in which the neighborhood graph is embedded. The growing mechanism of GNG's hidden layer is similar to the one of GCS, the main difference being that GNG uses a type of competitive Hebbian learning (CHL) to modify the topology structure of the graph. In particular, CHL will relate two hidden units as neighbors (an edge connecting the corresponding two graph nodes will be created), if they are the two most active nodes, when a pattern is presented. On the other hand, a neighborhood relationship between two nodes is dissolved, if the activation signals of those two hidden nodes are not positively correlated. The adoption of CHL principles for the training of the hidden layer allows GNG networks to form neighborhood graphs embedded in a space, whose dimensionality varies by adapting to local, data distribution profiles. Apart from the aforementioned differences, GNG and GCS are structurally the same.

### B. Semi-Supervised FAM & EAM Networks

Semi-supervised FAM and EAM classifiers are real-time, recurrent neural networks that were derived from the original FAM architecture. Their basic building block is the ART-module, a three-layered (for ssEAM, two-layered) neuronal structure, whose output and hidden layer are bi-directionally

interconnected. Moreover, the output layer features lateral connections and acts as a competitive layer. Prior to predicting class labels, these architectures summarize the input data in a self-organizing fashion into clusters, which in the adaptive resonance theory literature are referred to as categories. The geometric interpretation of categories for ssFAM and ssEAM are respectively hyper-rectangles and hyper-ellipsoids embedded in the feature space. Figure 2 shows an example of category representations for a two-dimensional input space.

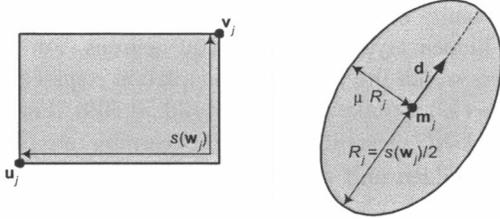


Fig. 2. Geometrical representations of ssFAM (on the left) and ssEAM (on the right) categories for a 2-dimensional input space.

ssFAM categories are specified by the min- and max-points of the corresponding hyper-rectangle, while ssEAM categories are described by a center, a Mahalanobis radius and a direction vector. The eccentricity of ssEAM's hyper-ellipsoids is determined via the network parameter  $\mu$ , which is defined as the ratio of the hyper-ellipsoid's major axis length over the length of each other minor axis. The maximum size ssFAM and ssEAM categories can reach is dictated by the value of another network parameter, the baseline vigilance parameter  $\bar{\rho}$ . When a pattern is being presented, output neurons of the ART-module compete and the one with the highest activation either assigns (according to its associated category) a class label to the pattern during performance phase or is eligible to learn the new pattern during training. These activation signals are influenced by the choice parameter  $a$  of ssFAM and ssEAM.

Learning in the two networks occurs via expansion of already existing categories or via the creation of new ones. The expansion mechanism is illustrated in Figure 3 for a special mode of learning, called fast learning. The hyper-rectangle or hyper-ellipsoid expands enough to include the pattern to be learnt. Also, in rough terms, if the pattern falls inside a category's geometrical representation it is assumed to be already known. Unlike GCS and GNG, there is no need to pre-specify an upper limit for the number of clusters to be created. If a pattern does not fit the particular characteristics of already existing categories a new category will be created. Note that in ssFAM/ssEAM categories (clusters) are never removed to ensure stability of the learning process.

One of the main, attractive characteristics of the original FAM classifier is its ability to complete its training in a finite number of steps (finite, stable learning), when fast learning of patterns is employed. By completion of training we mean that the model has learnt how to correctly label the entire training

set. While this feature of FAM is desirable in some cases, it will exhibit poor generalization due to over-fitting the training data, when there is an inherent, high class distribution overlap.

Both semi-supervised architectures address this issue by allowing some errors to be committed during training, while maintaining the finite, stable learning feature of FAM. In ssFAM and ssEAM this is accomplished via a tunable misclassification tolerance parameter  $\epsilon$  that eventually determines the level of the networks' post-training (resubstitution) error. A value of 0 corresponds to fully-supervised learning, according to which a category can learn only patterns of a specific label; under these circumstances ssFAM behaves like FAM. On the other hand a value of 1 allows a category to learn any pattern independently of class label (unsupervised learning of categories).

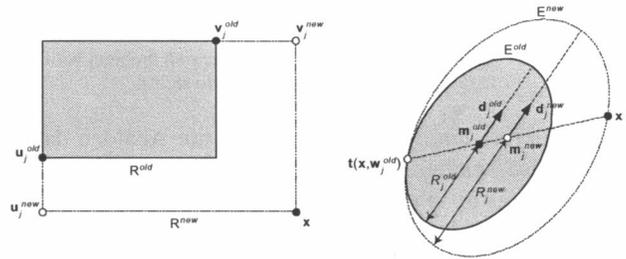


Fig. 3. Learning in ssFAM (on the left) and ssEAM (on the right) through category expansion.

We realize that in this section we were only able to provide a rudimentary, brief portrayal of the neural architectures we chose to focus on. Therefore, we refer the interested reader to references [5-11] for more detailed descriptions.

### III. EXPERIMENTAL RESULTS

In order to compare the GCS, GNG, ssEAM, and ssFAM classifiers we conducted a series of experiments using three artificially-generated data sets as well as four real data sets. Below we provide a brief description for each one of them.

#### A. Description of data sets

1) *G4 data sets*: Three data sets were generated by sampling from a bi-variate mixture of isotropic Gaussian distributions with equal priors consisting of 4 components; each component corresponded to a separate class distribution. The class means were positioned in such a way so that the featured four-fold symmetry with respect to the horizontal and vertical axes. We selected this particular distribution scheme in order to know *a priori* the exact Bayes error for the associated classification problems. By changing the spatial separation of the 4 means while preserving their symmetry we generated 3 datasets named G4LO, G4ME and G4HI with

Bayes error 0.05, 0.15 and 0.4 respectively. In the sequel will be referring to the Bayes error of these datasets as *overlap* expressed in percentage. For each one of the aforementioned 3 cases we generated a training set, a cross-validation set and a test set of 500, 5,000 and 5,000 patterns respectively. We kept the training set size low to minimize the required training time of the models and we chose to use larger cross-validation and test sets to achieve a capability for finer resolution of the subsequent statistical tests we conducted. Figure 4 depicts scatter plots of test patterns from each data set.

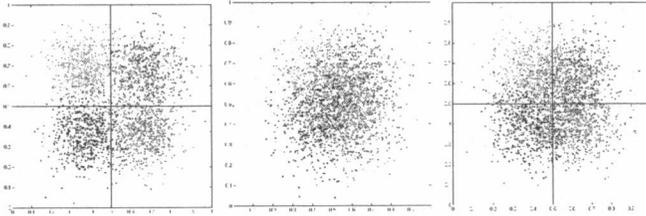


Fig. 4. Scatter plots of test patterns for the 4-Gaussian problem with 5%, 15%, and 40% overlap (from left to right).

2) *Abalone data set*: We also used the Abalone data set [13], whose patterns 8-dimensional with each feature corresponding to physical measurements of the abalone marine creature. In particular, the features are Sex, Length, Diameter, Height, Whole weight, Shucked weight, Viscera weight and Shell weight. The learning task associated to this data set is to predict the abalone's age by predicting its number of rings, which ranges from 1 to 29. The number of rings plus 1.5 equals the age of the creature.

This data set is treated as a 3-category classification problem (grouping ring classes 1-8, 9 and 10, and 11 on) with a total number of 4177 instances. Out of this many patterns, we used 1000 to make up a training set, another 2133 for a cross-validation set, and the remaining (1044 patterns) for testing purposes. We chose the Abalone set because the classification of its data has been reported in the literature to be a relatively difficult task with low percentage of correct classification.

3) *Glass, Pima Indian Diabetes, Cancer data sets*: Finally, we drew model comparisons on 3 out of the 4 data sets utilized in [12]. Due to lack of space, we refer the reader to [13] for a detailed description of the sets. We only mention in brief, that the sets are relatively small in size containing only 214, 768 and 690 total examples respectively. Furthermore, we used identical training, cross-validation and test subset sizes as in [12].

### B. Experimental Setup

In this subsection we explain our experimental setup and the particular methodology we utilized to conduct the comparisons on classification accuracy and size complexity of the 4 classifiers. For each model-dataset pair we used a large number of combinations of training parameter values.

More specifically, the same combinations of parameter values for each classifier type were used for all 4 data sets. Furthermore, each combination gave rise to an individual trained model. From the set of all trained models of each classifier type (GCS, GNG, ssEAM and ssFAM) we selected the 100 most accurate models in terms of classification performance on the cross-validation set. Finally, we assessed the accuracy and size complexity of those 100 best networks on a separate test set. We followed this particular scheme to avoid selection bias.

In order to train GCS models we used a GCS graph dimensionality of 1 through 6 and we limited the maximum size of hidden layer to 50 and 100 neurons. For the GNG classifiers we let the number of adaptation steps before node insertion ( $\lambda$ ) to take values in [100:50:300]. For both the GCS and GNG classifiers we let the learning rate for the best matching hidden unit ( $\epsilon_b$ ) to take values in [0.1:0.05:0.25], of the neighboring hidden nodes ( $\epsilon_n$ ) in {0.006, 0.012} and the learning rate of output nodes ( $\eta$ ) in [0.1:0.05:0.3]. Let us note here that these ranges of training parameter values are a superset of those specified in [12].

With respect to ssFAM and ssEAM the following values were used:  $\epsilon$  in the range [0.0:0.1:1.0], baseline vigilance  $\bar{\rho}$  in [0.0:0.05:0.75]  $\cup$  [0.75:0.01:0.99] and choice parameter  $a$  in {0.0001, 0.001, 0.01, 0.1, 1.0}. For ssEAM the axes ratio parameter  $\mu$  took values in [0.3:0.1:1.0]. Also, for both architectures a value of 0 was used for  $\bar{\rho}$  during the networks' performance phases to force class label assignments on all cross-validation and test set patterns. Finally, for all 4 network types the training set patterns were presented in 100 different orders during training; the selection of each particular order can be thought of as an additional training parameter. The aforementioned parameter values and value ranges gave approximately 20,000 training parameter value combinations per network type and data set. In other words, about 20,000 models of each kind were trained on each of the 4 data sets resulting in a total of 320,000 models having being trained approximately.

### C. Observations

In the following presentation and discussion of the results we have obtained, PIC will stand for percent incorrect classification, that is, the error rate of a classifier. Also, we define as size of the architecture to be its structural complexity: for GCS and GNG classifiers it is the number of hidden neurons utilized, while for ssFAM, ssEAM and FAM it is the number of categories employed. Ideally, a classifier should have the lowest possible PIC and simultaneously have the smallest possible size for a given classification problem. The comparison of the 4 models is conducted strictly with respect to the test set. Finally, we call champion model the network that has lowest PIC within the set of models of the same model family for a given data set.

Table I depicts the maximum, median, minimum and standard deviation for the PIC on the test set for each classifier type and each data set considered. In this table, best values for each row are depicted in bold. Also, Figures 5(a) through 5(d) are generalization performance (PIC on test set) versus structural complexity (size) plots of then 10 best models from each family for the G4LO, G4HI, Diabetes and Abalone data sets respectively.

TABLE I

	PIC Test for 100 best classifiers				
G4LO	GCS	GNG	ssEAM	ssFAM	FAM
Max	10.58	<b>10.48</b>	10.74	<b>10.48</b>	14.34
Median	10.48	10.4	10.62	<b>10.38</b>	13.72
Min	<b>10.14</b>	10.18	10.56	<b>10.14</b>	12.6
Std.	0.09211	0.07888	<b>0.06457</b>	0.10203	0.30566
G4ME					
Max	25.5	<b>24.86</b>	26.36	25.54	33.56
Median	25.36	<b>24.74</b>	26	25.2	31.98
Min	24.88	<b>24.4</b>	25.4	24.78	31.52
Std.	0.14061	<b>0.12173</b>	0.26508	0.25875	0.39804
G4HI					
Max	42.34	<b>41.72</b>	42.76	42.3	49.46
Median	42.16	<b>41.58</b>	42.4	42.06	48.52
Min	41.36	<b>41.02</b>	41.52	41.28	47.44
Std.	0.21283	<b>0.1378</b>	0.34975	0.25216	0.48651
CANCER					
Max	0.5747	<b>0</b>	<b>0</b>	0.5747	5.1724
Median	0.5747	<b>0</b>	<b>0</b>	<b>0</b>	1.1494
Min	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Std.	0.2671	<b>0</b>	<b>0</b>	0.0809	1.5883
GLASS					
Max	<b>35.8491</b>	43.3962	39.6226	<b>35.8491</b>	50.9434
Median	35.8491	39.6226	39.6226	<b>33.9623</b>	39.6226
Min	32.0755	33.9623	<b>30.1887</b>	32.0755	33.9623
Std.	<b>1.1024</b>	2.4243	1.8443	1.3564	3.8905
DIABETES					
Max	<b>26.3542</b>	27.3958	26.5625	26.5625	35.938
Median	25.8333	<b>26.3542</b>	<b>25</b>	<b>25</b>	27.604
Min	20.1042	21.1458	22.3958	<b>19.7917</b>	20.313
Std.	1.0586	1.0824	<b>1.0232</b>	1.6368	3.0864
ABALONE					
Max	47.51	47.797	46.935	<b>46.839</b>	56.897
Median	47.126	47.51	46.743	<b>46.456</b>	53.161
Min	45.498	45.881	45.594	<b>45.211</b>	49.234
Std.	0.39718	0.40604	<b>0.33461</b>	0.4532	1.4949

From Table I we observe that for the G4 databases GCS and GNG champion models exhibit the smallest PIC point estimates, for the Glass, Diabetes and Abalone data sets the best models appear to be either ssEAM or ssFAM, while for the Cancer database there is tie. However, at significance level of 0.01 the differences in PIC point estimates between the 4 types of champion (minimum PIC) models are statistically insignificant for the all datasets. For example, when comparing GCS and GNG to the semi-supervised ARTMAP architectures on the G4 data sets, the differences in PIC of the champion models is below 1%, which is statistically insignificant, while a difference of slightly more than 1% in PIC would have lead to the opposite conclusion. An analogous statement also holds for the results pertaining to the rest of the databases.

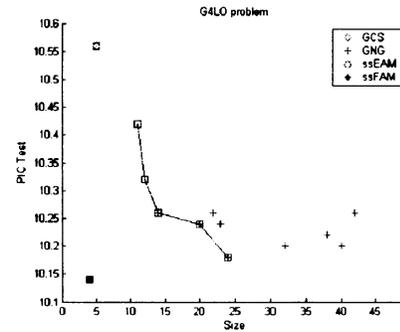


Fig. 5(a). PIC on test set versus network size; 4-G dataset; 5% overlap.

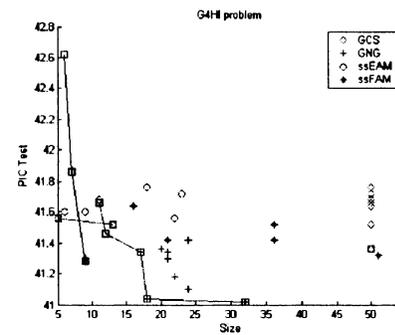


Fig. 5(b). PIC on test set versus network size; 4-G dataset; 40% overlap.

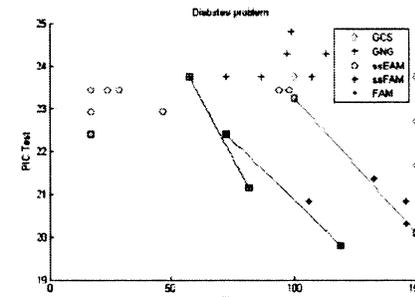


Fig. 5(c). PIC on test set versus network size; Diabetes dataset.

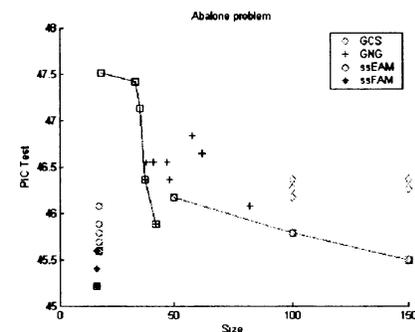


Fig. 5(d). PIC on test set versus network size; Abalone dataset.

In particular, for the Glass and Diabetes data sets we can attribute this fact to the small sizes of the test sets that had to be used (53 and 192 patterns respectively). A notable exception is the champion ssFAM network for the Abalone dataset, which differs by 1.04% with the champion GNG network, the difference being statistically significant. Also,

there are some indications that the generalization performance of ssFAM and ssEAM tends to increase relatively to that of GCS and GNG in high class distribution overlap, an observations which is aligned with the findings in [11]. For example, the difference in PIC Test of the best ssFAM network from the corresponding GNG model is 0.38%, 0.26%, and -0.67% for the G4ME, G4HI, and Abalone data sets respectively. The corresponding quantities for ssEAM and GNG are 1.0%, 0.5%, and -0.29%. The aforementioned remarks can also be observed in Figures 5(a) through 5(d) for the G4Lo, G4HI, Diabetes and Abalone data sets. With respect to FAM, all champion GCS, GNG, ssFAM and ssEAM are superior in terms of PIC on the G4 and Abalone datasets, since all differences in PIC are rather large (ranging from 2% to 6%) and, thus, turn out to be statistically significant. However, the contrary holds for the remaining datasets, where the differences are too small.

With respect to structural complexity, it is clearly exhibited in our results (for example, as shown in Figures 5) that ssFAM and ssEAM networks achieve comparable generalization performance to the GCS and GNG classifiers, while requiring a smaller network size. Notice that for all 4 data sets ssFAM and ssEAM feature network sizes below 20 and in certain cases below 10. An extreme difference in size occurs for the Abalone data set, where both semi-supervised ARTMAP need 17 categories compared to GNG's and GCS' champion models of sizes 42 and 150 respectively. Finally, let us mention in passing that the ssFAM and ssEAM training phases were less time consuming than the ones of GCS and GNG; combined with the fact that the former models are smaller size, GCS and GNG are associated with higher computational overhead, whether with respect to training or performance phase.

On balance, our results demonstrate that FAM's performance challenges pointed out in [12] are more likely to arise from the fully-supervised learning nature of FAM, which may lead to over-training and even to larger models, rather than from the piece-wise linear nature of its induced decision boundaries, since ssFAM also creates decision boundaries with this property. Furthermore, by using semi-supervised learning, ARTMAP architectures like ssFAM and ssEAM may become comparable in generalization abilities to GCS and GNG classifiers, while requiring smaller network sizes.

#### IV. SUMMARY

In this paper we present an experimental comparison of four neural-based classifiers, namely Growing Cell Structures (GCS), Growing Neural Gas (GNG), Semi-Supervised Fuzzy ARTMAP (ssFAM) and Semi-Supervised Ellipsoid ARTMAP (ssEAM). The experimental comparison is drawn in terms of generalization performance and structural complexity of the resulting classifiers. Earlier studies implied that Fuzzy ARTMAP, which utilizes fully-supervised learning, may suffer from poor generalization performance,

when compared to GCS and GNG classifiers, especially in hard classification problems featuring high class distribution overlap. Through our experimental study we concluded that such deficiencies are more likely linked to fully-supervised learning and that ARTMAP classifiers equipped with semi-supervised learning capabilities can improve their performance with respect to GCS and GNG classifiers, while maintaining lower structural complexity.

#### ACKNOWLEDGEMENT

The authors would like to thank Donald Wunsch and three, other anonymous reviewers for their helpful feedback on the manuscript's contents. Finally, the authors' research was partially supported by the National Science Foundation under CCLI-EMD grant No. 0341601.

#### REFERENCES

- [1] D. E. Rumelhart, J. L. McClelland, and PDP Research Group, *Parallel Distributed Processing; Explorations in the Microstructure of Cognition: Volume 1—Foundations*, Cambridge, MA: MIT Press, 1988.
- [2] P.J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," Ph.D. Thesis, Harvard University, Cambridge, MA, 1974.
- [3] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, pp. 59–69, 1982.
- [4] J. Moody and C. Darken, "Learning with localized receptive fields," in *Proc. 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, (Eds.) San Mateo, CA: Morgan Kaufmann, 1988, pp. 133–143.
- [5] B. Fritzke, "Growing cell structures—A self-organizing network for unsupervised and supervised learning," *Neural Networks*, 7(9), pp. 1441-1460, 1994.
- [6] B. Fritzke, "A Growing Neural Gas Network Learns Topologies," G. Tesauro, D.S. Touretzky and T.K. Leen (Eds.), *Advances in Neural Information Processing Systems 7*, Cambridge, MA: MIT Press, 1995.
- [7] S. Grossberg, "Adaptive Pattern Recognition and Universal Encoding II: Feedback, Expectation, Olfaction, and Illusions", *Biological Cybernetics*, 23, pp. 187-202, 1976.
- [8] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds and D.B. Rosen, "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps", *IEEE Transaction on Neural Networks*, Vol. 3:5, pp. 698-713, 1992.
- [9] Verzi, S.J., Heileman, G.L., Georgiopoulos, M. & Anagnostopoulos, G.C. (2002). Off-line Structural Risk Minimization and BARTMAP-S, *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN '02)*, Vol. 3, (pp. 2533-2538), 2002.
- [10] G.C. Anagnostopoulos, M. Georgiopoulos, S.J. Verzi & G.L. Heileman, "Reducing Generalization Error and Category Proliferation in Ellipsoid ARTMAP via Tunable Misclassification Error Tolerance: Boosted Ellipsoid ARTMAP", *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN '02)*, Vol. 3, (pp. 2650-2655), 2002.
- [11] G.C. Anagnostopoulos, M. Bharadwaj, M. Georgiopoulos, S.J. Verzi, & G.L. Heileman, "Exemplar-based Pattern Recognition via Semi-Supervised Learning", *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN '03)*, Vol. 3, (pp. 1350-1356), 2003.
- [12] F. Hamker and D. Heinke, "Implementation and comparison of growing neural gas, growing cell structures and fuzzy artmap," *Tech. Rep., Schriftenreihe des FG Neuroinformatik der TU Ilmenau, Report 1/97*, 1997.
- [13] P. M. Murphy, and D. W. Aha: UCI repository of machine learning databases. <http://www.ics.uci.edu/~mllearn/MLRepository.html> (1998).