Supervised Incremental Learning with the Fuzzy ARTMAP Neural Network

Jean-François Connolly, Eric Granger, and Robert Sabourin

Laboratoire d'imagerie, de vision et d'intelligence artificielle
Dépt. de génie de la production automatisée
École de technologie supérieure
1100 rue Notre-Dame Ouest,
Montreal, Quebec,
Canada, H3C 1K3*
jfconnolly@livia.etsmtl.ca, eric.granger@etsmtl.ca,
robert.sabourin@etsmtl.ca

Abstract. Automatic pattern classifiers that allow for on-line incremental learning can adapt internal class models efficiently in response to new information without retraining from the start using all training data and without being subject to catastrophic forgeting. In this paper, the performance of the fuzzy ARTMAP neural network for supervised incremental learning is compared to that of supervised batch learning. An experimental protocole is presented to assess this network's potential for incremental learning of new blocks of training data, in terms of generalization error and resource requirements, using several synthetic pattern recognition problems. The advantages and drawbacks of training fuzzy ARTMAP incrementally are assessed for different data block sizes and data set structures. Overall results indicate that error rate of fuzzy ARTMAP is significantly higher when it is trained through incremental learning than through batch learning. As the size of training blocs decreases, the error rate acheived through incremental learning grows, but provides a more compact network using fewer training epochs. In the cases where the class distributions overlap, incremental learning shows signs of over-training. With a growing numbers of training patterns, the error rate grows while the compression reaches a plateau.

1 Introduction

The performance of statistical and neural pattern classifiers depends heavily on the availability of representative training data. The collection and analysis of such data is expensive and time consuming in many practical applications. Training data may, therefore, be incomplete in one of several ways. In an environments where class distributions remain fixed, these include a limited number of training observations, missing components of the input observations, missing class labels during training, and missing classes (*i.e.,* some classes that were not present in the training data set may be encountered during operations) [7].

* This research was supported in part by the Natural Sciences and Engineering Research Council of Canada.

L. Prevost, S. Marinai, and F. Schwenker (Eds.): ANNPR 2008, LNAI 5064, pp. 66–77, 2008.

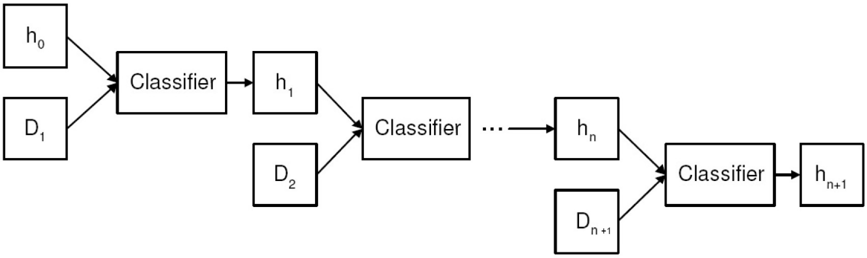


Fig. 1. A generic incremental learning scenario where blocks of data are used to update the classifier in an incremental fashion over time. Let $D_1, D_2, ..., D_{n+1}$ be the blocks of training data available to the classifier at discrete instants in time $t_1, t_2, ..., t_{n+1}$. The classifier starts with initial hypothesis h_0 which constitutes the prior knowledge of the domain. Thus, h_0 gets updated to h_1 on the basis of D_1, and h_1 gets updated to h_2 on the basis of data D_2, and so forth [5].

Given an environment where class distributions are fixed, and in which training data is incomplete, a critical feature of future automatic classification systems is the ability to update their class models incrementally during operational phases in order to adapt to novelty encountered in the environment [5] [9]. As new information becomes available, internal class models should be refined, and new ones should be created on the fly, without having to retrain from the start using all the cumulative training data.

For instance, in many practical applications, additional training data may be acquired from the environment at some point in time after the classification system has originally been trained and deployed for operations (see Fig. 1). Assume that this data is characterized and labeled by a domain expert, and may contain observations belonging to classes that are not present in previous training data, and classes may have a wide range of distributions. It may be too costly or not feasible to accumulate and store all the data used thus far for supervised training, and to retrain a classifier using all the cumulative data[1]. In this case, it may only be feasible to update the system through *supervised incremental learning*.

Assuming that new training data becomes available, incremental learning provides the means to efficiently maintain an accurate and up-to-date class models. Another advantage of incremental learning is the lower convergence time and memory complexity required to update a classifier. Indeed, temporary storage of the new data is only required during training, and training is only performed with the new data. Strategies adopted for incremental learning will depend on the application – the nature of training data, the environment, performance constraints, etc. Regardless of the context, updating a pattern classification system in an incremental fashion raises several technical issues.

In particular, accommodating new training data may corrupt the classifier's previously acquired knowledge structure and compromise its ability to achieve a high level of generalization during future operations. The *stability-plasticity dilemma* [1] refers to

[1] The vast majority of statistical and neural pattern classifiers proposed in literature can only perform *supervised batch learning* to learn new data. They must accumulate and store all training data in memory, and retrain from the start using all previously-accumulated training data.

the problem of learning new information incrementally, yet overcoming the problem of catastrophic forgetting.

This paper focuses on techniques that are suitable for supervised incremental learning in an environment where class distributions remain fixed over time. According to Polikar [12], an incremental learning algorithm should:

1. allow to learn additional information from new data,
2. not require access to the previous training data,
3. preserve previously acquired knowledge, and
4. accommodate new classes that may be introduced with new data.

In literature, some promising pattern classification algorithms have been reported for supervised incremental learning in environments where distributions are fixed. For example, the ARTMAP [2] and Growing Self-Organizing [6] families of neural network classifiers, have been designed with the inherent ability to perform incremental learning. In addition, some well-known pattern classifiers, such as the Support Vector Machine (SVM), and the Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) neural networks have been adapted to perform incremental learning [10] [11] [13]. Finally, some high-level architectures, based on well-known pattern classifiers, *e.g.*, Ensemble of Classifiers, have also been proposed [12].

In this paper, the performance of the fuzzy ARTMAP [4] neural network is characterize for supervised incremental learning of new blocks of training data in an environment where class distributions are fixed. Fuzzy ARTMAP is the most popular ARTMAP network. While its incremental learning capabilities are often cited in literature, to the authors knowledge, these capabilities have not been assessed. An experimental protocole has thus been defined such that the impact on performance of learning a new block of training data incrementally, after each network has previously been trained, is assessed for different types of synthetic pattern recognition problems. The first type of problem consists of data with overlapping class distributions, whereas the second type involves data with complex decision boundaries but no overlap. With this protocole, the advantages and drawbacks of the ARTMAP architectures are discussed for incremental learning data using different data block sizes, and using different data set structures (overlap, dispersion, etc.).

In the next section, fuzzy ARTMAP is briefly reviewed. Then, the experimental protocol, performance measures and synthetic data sets, used for proof-of-concept computer simulations, are described in Section 3. Finally, experimental results are presented and discussed in Section 4.

2 ARTMAP Neural Networks

ARTMAP refers to a family of neural network architectures based on Adaptive Resonance Theory (ART) [1] that is capable of fast, stable, on-line, unsupervised or supervised, incremental learning, classification, and prediction [2]. A key feature of the ARTMAP networks is their unique solution to the stability - plasticity dilemma.

Several ARTMAP networks have been proposed in order to improve the performance of these architectures. Members of the ARTMAP family can be broadly divided according to their internal matching process, which depends on either deterministic or probabilistic category activation. The deterministic type consists of networks such as fuzzy ARTMAP, ART-EMAP, ARTMAP-IC, default ARTMAP, simplified ARTMAP, distributed ARTMAP, etc., and represent each class using one or more fuzzy set hyper-rectangles. In contrast, the probabilistic type consists of networks such as PROBART, PFAM, MLANS, Gaussian ARTMAP, ellipsoid ARTMAP, boosted ARTMAP, μARTMAP, etc., and represent each class using one or more probability density functions.

This paper focuses on the popular fuzzy ARTMAP neural network [4]. It integrates the fuzzy ART [3] in order to process both analog and binary-valued input patterns to the original ARTMAP architecture [2]. The rest of this section provides a brief description of fuzzy ARTMAP.

2.1 Fuzzy ARTMAP

The fuzzy ART neural network consists of two fully connected layers of nodes: a $2M$ node input layer F_1 to accomodate complement-coded input patterns, and an N node competitive layer, F_2. A set of real-valued weights $\mathbf{W} = \{w_{ij} \in [0,1] : i = 1, 2, ..., 2M; j = 1, 2, ..., N\}$ is associated with the F_1-to-F_2 layer connections. The F_2 layer is connected, through learned associative links, to an L node map field F_{ab}, where L is the number of classes in the output space. A set of binary weights $\mathbf{W}^{ab} = \{w_{jk}^{ab} \in \{0,1\} : j = 1, 2, ..., N; k = 1, 2, ..., L\}$ is associated with the F_2-to-F_{ab} connections. Each F_2 node $j = 1, ..., N$ corresponds to a category that learns a prototype vector $\mathbf{w}_j = (w_{1j}, w_{2j}, ..., w_{2Mj})$, and is associate with one of the output classes $K = 1, ..., L$. During the training phase, fuzzy ARTMAP dynamics is govern by four hyperparameters: the choice parameter $\alpha > 0$, the learning parameter $\beta \in [0,1]$, the baseline vigilance parameter $\bar{\rho} \in [0,1]$, and the matchtracking parameter ϵ. In term of incremental learning, the learning algorithm is able to adjusts previously-learned categories, in response to familiar inputs, and to creates new categories dynamically in response to inputs different enough from those already seen.

The following describes fuzzy ARTMAP during supervised learning of a finite data set. When an input pattern $\mathbf{a} = (a_1, ..., a_M)$ is presented to the network and the vigilance parameter $\rho \in [0,1]$ is set to its baseline value $\bar{\rho}$. The original M dimensions input pattern $\mathbf{a}$ is complement-coded to make a $2M$ dimensions network's input pattern: $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c) = (a_1, a_2, ..., a_M; a_1^c, a_2^c, ..., a_M^c)$, where $a_i^c = (1 - a_i)$, and $a_i \in [0,1]$. Each F_2 node is activated according to the *Weber law choice function*: $T(\mathbf{A}) = |\mathbf{A} \wedge \mathbf{w}_j|/(\alpha + |\mathbf{w}_j|)$, and the node with the strongest activation $J = \text{argmax}\{T_j : j = 1, ..., N\}$ is chosen. The algorithm then verifies if $\mathbf{w}_J$ is similar enough to $\mathbf{A}$ using the vigiliance test: $|\mathbf{A} \wedge \mathbf{w}_J|/2M \geq \rho$. If node J fails the vigilance test, it is disactivated and the network searches for the next best node on the F_2 layer. If the vigilance test is passed, then the map field F^{ab} is activated through the category J and fuzzy ARTMAP makes a class prediction $K = k(J)$. In the case of an incorrect class prediction $K = k(J)$, a match tracking signal raises $\rho = (|\mathbf{A} \wedge \mathbf{w}_J|/2M) + \epsilon$. Node J is disactivated, and the search among F_2 nodes begins anew. If node J passes the vigilance test, and makes the correct prediction, its category is updated by adjusting its prototype vector: $\mathbf{w}_J$ to $\mathbf{w}'_J = \beta(\mathbf{A} \wedge \mathbf{w}_J) + (1 - \beta)\mathbf{w}_J$. On the other hand, if none of the nodes can satisfy both conditions (vigilance test and correct prediction), then a new F_2 node is initialed. This new node is assigned to class K by setting w_{Jk}^{ab} to 1 if $k = K$ and 0 otherwise.

Once the weights $\mathbf{W}$ and $\mathbf{W}^{ab}$ have been found through this process, the fuzzy ARTMAP can predict a class label from a input pattern by activating the best F_2 node J, which activates a class $K = k(J)$ on the F_{ab} layer. Predictions are obtained without vigilance and match tests.

3 Experimental Methodology

3.1 Experimental Protocole

In order to observe the impact on performance of training a classifier with supervised incremental learning for different data structures, several data sets were selected for computer simulations. The synthetic data sets are representative of pattern recognition problems that involve either (1) simple decision boundaries with overlapping class distributions, or (2) complex decision boundaries without overlap on decision boundaries. The synthetic data sets correspond to 2 classes problems, with a 2 dimensional input feature space. Each data subset is composed of an equal number of 10,000 patterns per class, for a total of 20,000 (2 classes) randomly-generated patterns.

Prior to a simulation trial, each data set is normalized according to the min-max technique and partitioned into two equal parts – the learning and test subsets. The learning subset is divided into training and validation subsets. They respectively contain 2/3 and 1/3 of patterns from each class of the learning subset. In order to perform block-wise hold-out validation over several training *epochs*[2], the training and validation subsets are again divided into b blocks. Each block D_i $(i = 1, 2, ..., b)$ contains an equal number of patterns per class. To observe the impact on performance of learning new blocks of training data incrementally for different data block sizes, two different cases are observed. The first case consists in training with $b = 10$, where $|D_i| = 1000$ patterns, while the second one consists in training with $b = 100$, where $|D_i| = 100$ patterns.

During each simulation trial, fuzzy ARTMAP is trained using a batch learning and incremental learning process. For *batch learning*, $|D_i|$ is set to the smaller block size, in our case $|D_i| = 100$, and the number of blocks D_n used for training is progressively increased from 1 to 100. For the n^{th} trial, performance is assessed after initializing a fuzzy ARTMAP network and training it until convergence on $B_n = D_i \cup ... \cup D_n$. Since there is 100 blocks D_i, there will be 100 trials.

On the other hand, *incremental learning* consists in training the ARTMAP networks, until convergence, over one or more training epochs on successive blocks of data D_i. The training of each data block is done in isolation without reinitializing the networks. In the case of incremental learning two block sizes will be tested: $|D_i| = 100$ and when $|D_i| = 1000$. At first, performance is assessed after initializing an ARTMAP network and training on D_1. Then it is assessed after training the *same* ARTMAP network incrementally on D_2, and so on, until all b blocks are learned.

For each trial, learning is performed using a hold-out validation technique, with network training halted for validation after each epoch [8]. The performance of fuzzy

[2] An epoch is defined as one complete presentation of all the patterns of a finite training data set.

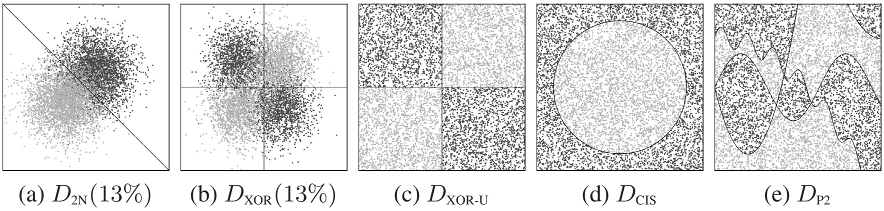
ARTMAP was measured when using standard parameter settings that yield minimum network resources (internal categories, epochs, etc.): $\beta = 1$, $\alpha = 0.001$, $\bar{\rho} = 0$ and $\epsilon = 0.001$ [4].

Since ARTMAP performance is sensitive to the presentation order of the training data, the pattern presentation orders were always randomized from one epoch to the next. In addition, each simulation trial was repeated 10 times with 10 different randomly generated data sets (learning and test). The average performance of fuzzy ARTMAP was assessed in terms of resources requirements and generalisation error. The amount of resources is measured by compression and convergence time. *Compression* refers to the average number of training patterns per category prototype created in the F_2 layer. *Convergence time* is the number of epochs required to complete learning for a training strategy. It does not include presentations of the validation subset used to perform hold-out validation. *Generalisation error* is estimated as the ratio of incorrectly classified test subset patterns over all test set patterns. The combination of compression and convergence time provides useful insight into the amount of processing required by fuzzy ARTMAP during training to produce its best asymptotic generalisation error. Average results, with corresponding standard error, are always obtained, as a result of the 10 independent simulation trials.

3.2 Data Sets

Of the five synthetic data sets selected for simulations, two have simple decision boundaries with overlapping class distributions ($D_{\text{2N}}(\xi_{tot})$ and $D_{\text{XOR}}(\xi_{tot})$) and three have complex decision boundaries without overlap ($D_{\text{XOR-U}}$, D_{CIS} and D_{P2}). The two classes in D_{2N} and D_{XOR} are randomly generated with normal distributions and the total theoretical probability of error associated with these problems is denoted by ξ_{tot}. Data from the classes in $D_{\text{XOR-U}}$, D_{CIS} and D_{P2} are uniform distributions, and since class distributions do not overlap on decision boundaries, the total theoretical probability of error for these data sets is 0.

The $D_{\text{2N}}(\xi_{tot})$ data (Fig. 2a) consists of two classes, each one defined by a normal distribution in a two dimensional input feature space [8]. Both sources are described by variables that are independent, have equal covariance Σ, and their distributions are hyperspherical. With the $D_{\text{XOR}}(\xi_{tot})$ problem, data is generated by 2 classes according to bi-modal distributions (Fig. 2b). The four normal distributions are centered in the 4 squares of a classical XOR problem. For those two problems, the degree of overlap



(a) $D_{\text{2N}}(13\%)$ (b) $D_{\text{XOR}}(13\%)$ (c) $D_{\text{XOR-U}}$ (d) D_{CIS} (e) D_{P2}

Fig. 2. Exemple of the five data sets generated for a replication of each problem

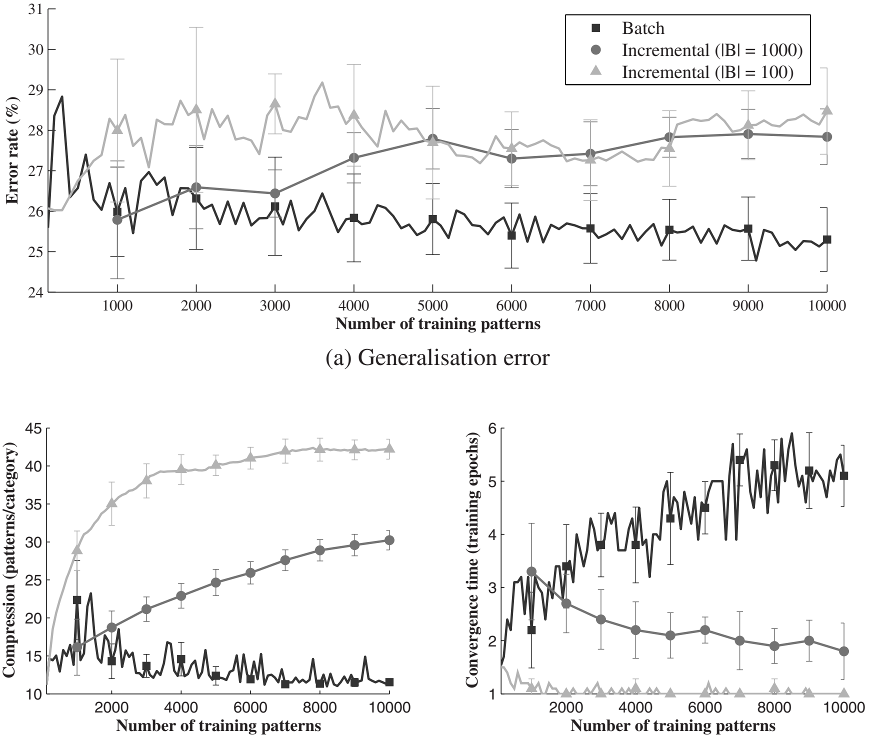
is varied from a total probability of error ξ_{tot} = 1%, 13%, and 25%, by changing the covariance of each normal distribution.

With the $D_{\text{XOR-U}}$ data, the 'on' and 'off' classes of the classical XOR problem are divided by a horizontal decision bound at $y = 0.5$, and a vertical decision bound at $x = 0.5$ (Fig. 2c). The Circle-in-Square problem D_{CIS} (Fig. 2d) requires a classifier to identify the points of a square that lie inside a circle, and those that lie outside a circle [2]. The circle's area equals half of the square. It consists of one non-linear decision boundary where classes do not overlap. Finally, with the D_{P2} problem (Fig. 2e), each decision region of its 2 classes is delimited by one or more of its four polynomial and trigonometric functions, and belongs to one of the two classes [14].

4 Simulation Results

4.1 Overlapping Class Distributions

Figure 3 presents the average performance achieved as a function of the training subset size, when fuzzy ARTMAP is trained using batch and incremental learning on the



(a) Generalisation error

(b) Compression

(c) Convergence time

Fig. 3. Average performance of the fuzzy ARTMAP versus training subset size for $D_{\text{XOR}}(13\%)$ using batch and incremental learning. Each curve is shown along with 90% confidence interval.

$D_{\text{XOR}}(13\%)$ data set. For incremental learning, block sizes of 100 and 1000 are employed. Very similar tendencies are found in simulation results with other data sets with class distributions overlap ($D_{\text{2N}}(\xi\%)$ and $D_{\text{XOR}}(\xi\%)$).

As shown in Fig. 3a, the error rate obtained by training fuzzy ARTMAP through incremental learning is generally significantly higher than that obtained through batch learning. Using the smaller block size ($|D_i| = 100$) yields a higher error rate that with the larger block size ($|D_i| = 1000$), but this difference is not significant. In addition, error tends to grow with the number of blocks having been learned. For example, after the fuzzy ARTMAP network undergoes incremental learning of 100 blocks with $|D_i| = 100$, the average error is about 29.3%, yet after learning 10 blocks with $|D_i| = 1000$, the error is about 27,6%.

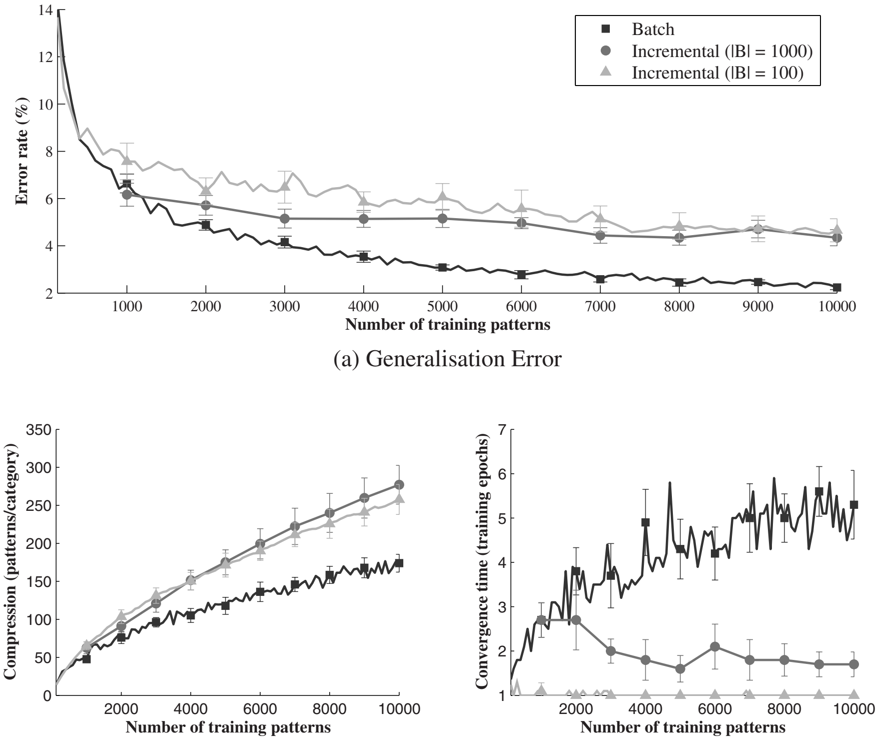
Although the error is greater, Fig. 3b indicates that the compression obtained when fuzzy ARTMAP is trained through incremental learning is significantly higher than if trained through batch learning, and it tends to grow as the block size is decreased. Incremental learning also tends to reduce the number of training epochs required for fuzzy ARTMAP to converge (see Fig. 3c). As the block size decreases, the convergence time tends towards 1. With incremental learning, the first blocks have a tendency to require a greater number of epochs. For example, after fuzzy ARTMAP undergoes incremental learning of 100 blocks with $|D_i| = 100$, the average compression and convergence time are about 40 patterns/category and 1.0 epoch, respectively. After learning 10 blocks with $|D_i| = 1000$, the compression and convergence time are about 30 patterns/category and 1.8 epochs. This compares favorably to fuzzy ARTMAP trained through batch learning, where the compression and convergence time are about 10 patterns/category and 5.0 epochs. In this case, the performance of fuzzy ARTMAP as the training set size grows is indicative of overtraining [8].

4.2 Complex Decision Boundaries

Fig. 4 presents the average performance achieved as a function of the training subset size, when the fuzzy ARTMAP is trained using batch and incremental learning on the D_{CIS} data set. Very similar tendencies are found in simulation results for other data set where complex boundaries and class distributions that do not overlap ($D_{\text{XOR-U}}$ and D_{P2}).

As shown in Fig. 4a, when the training set size increases, the average generalisation error of fuzzy ARTMAP trained with either batch or incremental learning decreases asymptotically towards its minimum. However, the generalisation error obtained by training fuzzy ARTMAP through incremental learning is generally significantly higher than that obtained through batch learning. As with the data that has overlapping class distributions, the error tends to grow as the block size decreases. However, after the fuzzy ARTMAP network performs incremental learning of 100 blocks with $|D_i| = 100$, the average error is comparable to after learning 10 blocks with $|D_i| = 1000$ (about 4.5%).

Again, training fuzzy ARTMAP through incremental learning yields a significantly higher compression than with batch learning (Fig. 4b). Furthermore, the convergence time associated with incremental learning is considerably lower than with batch learning (Fig. 4c). Results indicate that as the block size is decreased and the number of learned blocks increases, the convergence time with incremental learning tends towards



(a) Generalisation Error

(b) Compression

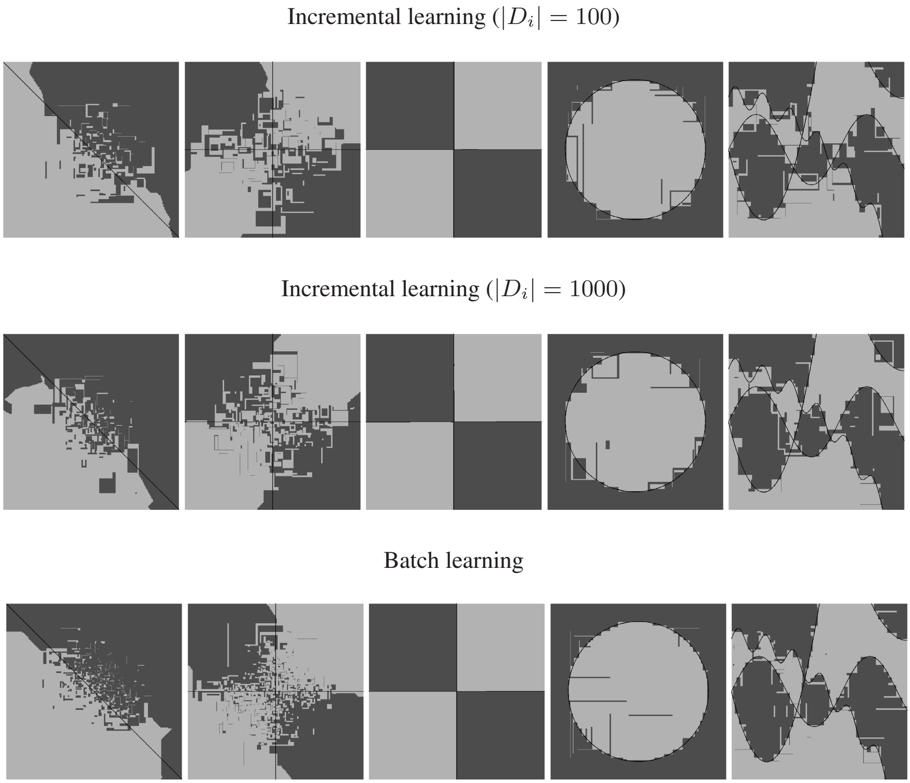
(c) Convergence time

Fig. 4. Average performance of the fuzzy ARTMAP versus training subset size for D_{CIS} using batch and incremental learning. Each curve is shown along with 90% confidence interval.

1. For example, after fuzzy ARTMAP undergoes incremental learning of 100 blocks with $|D_i| = 100$, the average compression and convergence time are about 260 patterns/category and 1.0 epoch, respectively. After learning 10 blocks with $|D_i| = 1000$, the compression and convergence time are about 280 patterns/category and 1.8 epochs.

4.3 Discussion

Overall results indicate that when fuzzy ARTMAP undergoes incremental learning, the networks tend to become more compact, but the error rate tends to degrade. As shown in Fig. 5, this is reflected by decision boundaries among classes that become coarser as the block size decreases. Note that batch learning is equivalent to $|D_1| = 10000$. Since training on each block is performed in isolation, when fuzzy ARTMAP is trained on large data blocks, it has sufficient information to converge toward an optimal solution. Small data blocks represent the higher bound on the error rate. In our study, the smallest block size considered is $|D_i| = 100$ where each pattern must only be presented, on average, one time to the neural network for convergence (Figs. 3c and 4c). In this case, the network can create or update the model, but it appears to lack the necessary information to truly converge toward a solution over several training epochs.


Incremental learning ($|D_i| = 100$)

Incremental learning ($|D_i| = 1000$)

Batch learning

(a) $D_{\mathrm{2N}}(13\%)$ (b) $D_{\mathrm{XOR}}(13\%)$ (c) $D_{\mathrm{XOR\text{-}U}}$ (d) D_{CIS} (e) D_{P2}

Fig. 5. Decision boundaries for the best replication after learning all of the training patterns through batch and incremental learning, using the five data sets. The boundaries are shown for incremental learning with $|D_i| = 100$, incremental learning with $|D_i| = 1000$, and batch learning (i.e. $|D_1| = 10000$).

Table 1. Average generalisation error of fuzzy ARTMAP classifiers trained using batch and incremental learning with blocks of $|D_i| = 1000$ and $|D_i| = 100$ on all data from synthetic sets. Values are shown with the 90% confidence interval.

Data Set	**Average generalisation error (%)**		
	Batch	**Incremental** ($|D_i| = 1000$)	**Incremental** ($|D_i| = 100$)
$D_{\mathrm{2N}}(1\%)$	2.1 ± 0.3	3.2 ± 0.5	2.8 ± 0.3
$D_{\mathrm{2N}}(13\%)$	21.6 ± 0.6	23.8 ± 0.9	24.9 ± 1.1
$D_{\mathrm{2N}}(25\%)$	35.8 ± 0.4	37.8 ± 0.4	38.4 ± 0.4
$D_{\mathrm{XOR}}(1\%)$	1.2 ± 0.1	1.7 ± 0.5	1.7 ± 0.4
$D_{\mathrm{XOR}}(13\%)$	25.3 ± 0.8	27.6 ± 0.6	29.3 ± 1.1
$D_{\mathrm{XOR}}(25\%)$	43.0 ± 0.5	44.0 ± 0.5	44.0 ± 0.6
$D_{\mathrm{XOR\text{-}U}}$	0.2 ± 0.1	0.6 ± 0.1	0.9 ± 1.1
D_{CIS}	2.2 ± 0.1	4.3 ± 0.3	4.7 ± 0.5
D_{P2}	5.1 ± 0.2	7.9 ± 0.3	9.4 ± 0.7

With overlapping data, even when the geometry of decision bounds matches the rectangular categories of fuzzy ARTMAP in $D_{\text{XOR}}(\xi_{tot})$, the network still leads to the well known category proliferation problem. Compared with $D_{\text{2N}}(\xi_{tot})$, the region of overlap isn't localized, so the proliferation is amplified and the error rate is higher. If the classes don't overlap, or if the overlapping is very low, results show that the complexity of boundaries with fuzzy ARTMAP is defined mainly by how well these boundaries can be represented using hyper-rectangles.

One key issue here is the internal mechanisms used by the network to learn new information. With the fuzzy ARTMAP network, only the internal vigilance parameter (ρ) is allow to grow dynamically during the learning process, over a range define by the baseline vigilance ($\bar{\rho}$). Since all network hyperparameter play an important role in fuzzy ARTAMP's ability to learn new data, one potential solution could imply optimizing the network's hyperparameters for incremental learning [8]. This way, all four fuzzy ARTMAP hyperparameters would be adapted such that the network would learn each data block $|D_i|$ to the best of its capabilities.

Another potential solution could be to exploit the learning block by organising the data in a specific way. Since the first blocks form the basis for future updates, results underline the importance of initiating incremental learning with blocks that contain enough representative data from the environment. With overlapping data, the first blocks could be organized to grow classes from the inside towards the overlapping regions, through some active learning strategy. With complex boundaries, the first blocks could be organized to define the non-linear bounds between classes.

5 Conclusion

In many practical applications, classifiers found inside pattern recognition systems may generalize poorly as they are designed prior to operations using limited training data. Techniques for on-line incremental learning would allow classifiers to efficiently adapt internal class models during operational phases, without having to retrain from the start using all the cumulative training data, and without corrupting the previously-learned knowledge structure. In this paper, fuzzy ARTMAP's potential for supervised incremental learning is assessed. An experimental protocole is proposed to characterize its performances for supervised incremental learning of new blocks of training data in an environment where class distributions are fixed. This protocole is based on a comprehensive set of synthetic data with overlapping class distributions and with complex decision boundaries, but no overlap.

Simulation results indicate that the average error rate obtained by training fuzzy ARTMAP through incremental learning is usually significantly higher than that obtained through batch learning, and that error tends to grow as the block size decreases. Results also indicate that training fuzzy ARTMAP through incremental learning often requires fewer training epochs to converge, and leads to more compact networks. As the block size decreases, the compression tends to increase and the convergence time tends towards one. The subject for futur work involve designing fuzzy ARTMAP networks that can approach the error rates of batch learning with incremental learning.

Some promising directions include organizing the blocks of training data through active learning and optimizing fuzzy ARTMAP hyperparameter values for incremental learning.

References

1. Carpenter, G.A., Grossberg, S.: A Massively Parallel Architecture for a Self-Organizing. Neural Pattern Recognition Machine, Computer, Vision, Graphics and Image Processing 37, 54–115 (1987)
2. Carpenter, G.A., Grossberg, S., Reynolds, J.H.: ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a SONN. Neural Networks 4, 565–588 (1991)
3. Carpenter, G.A., Grossberg, S., Rosen, D.B.: Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System. Neural Networks 4(6), 759–771 (1991)
4. Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., Reynolds, Rosen, D.B.: Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps. IEEE Trans. on Neural Networks 3(5), 698–713 (1992)
5. Caragea, D., Silvescu, A., Honavar, V.: Towards a Theoretical Framework for Analysis and Synthesis of Agents That Learn from Distributed Dynamic Data Sources. In: Emerging Neural Architectures Based on Neuroscience. Springer, Berlin (2001)
6. Fritzke, B.: Growing Self-Organizing Networks - Why? In: Proc. European Symposium on Artificial Intelligence, pp. 61–72 (1996)
7. Granger, E., Rubin, M.A., Grossberg, S., Lavoie, P.: Classification of Incomplete Data Using the Fuzzy ARTMAP Neural Network. In: Proc. Int'l Joint Conference on Neural Networks, vol. iv, pp. 35–40 (2000)
8. Granger, E., Henniges, P., Sabourin, R., Oliveira, L.S.: Supervised Learning of Fuzzy ARTMAP Neural Networks Through Particle Swarm Optimization. J. of Pattern Recognition Research 2(1), 27–60 (2007)
9. Kasabov, N.: Evolving Fuzzy Neural Networks for Supervised/Unsupervised Online Knowledge-Based Learning. IEEE Trans. on Systems, Man, and Cybernetics 31(6), 902–918 (2001)
10. Maloof, M.: Incremental Rule Learning with Partial Instance Memory for Changing Concept. In: Proc. of the IEEE Int'l Joint Conf. on Neural Networks, vol. 14(1), pp. 1–14 (2003)
11. Okamoto, K., Ozawa, S., Abe, S.: A Fast Incremental Learning Algorithm with Long-Term Memory. In: Proc. Int'l Joint Conf. on Neural Network, Portland, USA, July 20-24, vol. 1(1), pp. 102–107 (2003)
12. Polikar, R., Udpa, L., Udpa, S., Honavar, V.: Learn++: An Incremental Learning Algorithm for MLP Networks. IEEE Trans. Systems, Man, and Cybernetics 31(4), 497–508 (2001)
13. Ruping, S.: Incremental Learning with Support Vector Machines. In: Proc. IEEE Int'l Conf. on Data Mining, San Jose, USA, November 29 - December 2, pp. 641–642 (2001)
14. Valentini, G.: An Experimental Bias-Variance Analysis of SVM Ensembles Based on Resampling Techniques. IEEE Trans. Systems, Man, and Cybernetics – Part B: Cybernetics 35(6), 1252–1271 (2005)