Factors of Overtraining with Fuzzy ARTMAP Neural Networks

Philippe Henniges, Eric Granger and Robert Sabourin
Laboratoire d'imagerie, de vision et d'intelligence artificielle
Département de génie de la production automatisée
École de technologie supérieure
1100 rue Notre-Dame Ouest
Montréal, Québec, Canada, H3C 1K3
philippe@livia.etsmtl.ca, {eric.granger, robert.sabourin}@etsmtl.ca

Abstract – *In this paper, the impact of overtraining on the performance of fuzzy ARTMAP neural networks is assessed for pattern recognition problems consisting of overlapping class distributions, and consisting of complex decision boundaries with no overlap. Computer simulations are performed with fuzzy ARTMAP networks trained for one epoch, through cross-validation, and until network convergence, using several data sets representing these pattern recognition problems. By comparing the generalisation error and resources required by these networks, the extent of overtraining due to factors such as data set structure, training strategy, number of training epochs, data normalisation, and training set size, is demonstrated. A significant degradation in fuzzy ARTMAP performance due to overtraining is shown to depend on the training set size and the number of training epochs for pattern recognition problems with overlapping class distributions.*

I. INTRODUCTION

Fuzzy ARTMAP is a neural network architecture based on Adaptive Resonance Theory (ART) that is capable of supervised learning of an arbitrary mapping between clusters of the input space and their associated class labels. They have been successfully applied in tasks such as data mining [14], remote sensing [7], identification of radar emitters [10], recognition of handwritten characters [17], and signature verification [13]. Nonetheless, fuzzy ARTMAP neural networks are known to suffer from overtraining, or overfitting, especially on data sets with overlapping class distributions. Overtraining is somewhat related to the category proliferation problem. Indeed, the effects of overtraining in fuzzy ARTMAP are two fold – an increase in the generalization error, and in the amount of resources (*i.e.*, category prototypes and training epochs) required for training.

The issue of overtraining in fuzzy ARTMAP has been addressed by some authors in literature. For instance, Lerner *et al*. state that training the fuzzy ARTMAP until convergence produces better test set accuracy than training with cross-validation, although both training modes lead to overtraining [12]. Koufakou *et al*. and Gomez *et al.* claim that cross-validation yields better generalization, and significantly fewer category prototypes [8,11]. Furthermore, fuzzy ARTMAP can overtrain after only one training epoch of a large training set. Verzi *et al*. confirm that the effect of overtraining is mostly present with significantly overlapping data sets [19]. Many approaches have also been proposed to address the related category proliferation problem. Carpenter *et al*. propose a pruning method to eliminated unwanted categories [5]. Several models have also been proposed [9,19,20] to refine decision boundaries, and thus reduce overtraining for overlapping class distributions.

The main objectives of this paper are to confirm that fuzzy ARTMAP neural networks can overtrain, and to observe the extent to which such overtraining is linked to the data set structure, the training strategy, the number of training epochs, the normalisation of data, and the size of the training data set. An experimental protocol has been defined such that the contribution of these factors may be isolated for two different types of pattern recognition problems. The first type consists of overlapping class distributions, whereas the second type involves complex decision boundaries with no overlap. During computer simulations, fuzzy ARTMAP networks have been trained (1) for one complete epoch, (2) until their performance is maximized on a validation set, through cross-validation, (3) until their weight values remain constant for two successive epochs, and (4) until no training set patterns are misclassified. The performance of these networks has been assessed in terms of their generalisation error on test sets, and of the amount of resources required during training.

In the next section, fuzzy ARTMAP is briefly reviewed, along with typical training strategies used for supervised learning. Then, the data sets, experimental protocol, and performance measures used for proof-of-concept computer simulations are described in Section III. Finally, the results of simulations are presented and discussed in Section IV.

II. TRAINING STRATEGIES FOR THE FUZZY ARTMAP NEURAL NETWORK

ARTMAP refers to a family of neural network architectures capable of fast, stable, online, unsupervised or supervised, incremental learning, classification, and prediction [3,4]. ARTMAP is often applied using the simplified version shown in Figure 1. It is obtained by combining an ART unsupervised neural network [1] with a map field. The fuzzy ARTMAP variant [4] can process both analog and binary-valued input patterns by employing fuzzy ART [2] as the ART network.

The fuzzy ART neural network consists of two fully connected layers of nodes: an M node input layer F_1, and an N node competitive layer, F_2. A set of real-valued weights $\mathbf{W} = \{\omega_{ij} \in [0, 1]: i = 1, 2, \ldots, M; j = 1, 2, \ldots, N\}$ is associated with the F_1-to-F_2 layer connections. Each F_2 node j represents a recognition category that learns a prototype vector $\boldsymbol{w_j} = (\omega_{1j}, \omega_{2j}, \ldots, \omega_{Mj})$. 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 $\mathbf{W}^{ab} = \{ \omega_{jk}^{ab} \in \{0, 1\}: j = 1, 2, \ldots, N; k = 1, 2, \ldots, L\}$ is associated with the F_2-to-F^{ab} connections. The vector $\boldsymbol{w}_j^{ab} = (\omega_{1L}^{ab}, \omega_{2L}^{ab}, \ldots, \omega_{jL}^{ab},)$ links F_2 node j to one of the L output classes.

During training, fuzzy ARTMAP classifiers perform supervised learning of the mapping between training set patterns $\mathbf{a} = (a_1, a_2, \ldots, a_m)$ and output labels $\mathbf{t} = (t_1, t_2, \ldots, t_L)$, where $t_K = 1$ if K is the target class label for $\mathbf{a}$, and zero elsewhere. The following four training strategies are typically used for supervised learning of a training data set.

One epoch (1EP): The learning phase ends after one epoch of the training data set. (An epoch is defined as one complete presentation of all the training set patterns.)

Convergence based on weight values (CONV$_\mathbf{w}$): The learning phase ends after the epoch for which the weight values have converged. Convergence occurs when the sum-squared-fractional-change, $\mathbf{SSFC} = \sum_i (\mathbf{W}_i - \mathbf{W}_{i-1})^2$ of weights for a two successive epochs, i-1 and i, is less than 0.001 (SSFC < 0.001).

Convergence based on training set classifications (CONV$_\mathbf{p}$): The learning phase ends after the epoch for which all patterns from the trainings data set have been correctly classified by the network. Convergence occurs when $\sum_i (\mathbf{t}_i - \mathbf{y}_i^{ab}) = 0$ over all training set patterns.

Cross-validation (CV): The learning phase ends after the epoch for which the generalisation error is minimized for an independent validation set. Learning is performed using a hold-out cross-validation technique [15,16].

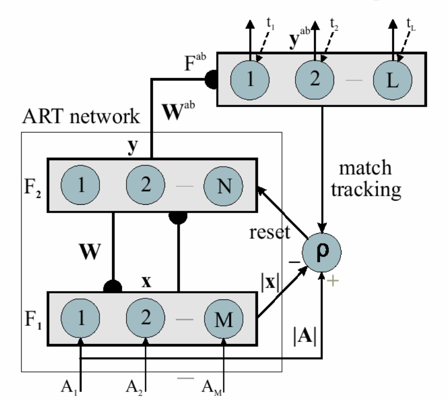


Figure 1 – An ARTMAP neural network architecture specialized for patter classification [10].

Once the learning phase has ended, fuzzy ARTMAP can predict a class label for an input pattern **a**.

III. EXPERIMENTAL PROTOCOL

In order to observe the effects of overtraining from a perspective of different data structure, several data sets were selected for computer simulations. The four following synthetic data sets are representative of pattern recognition problems that involve either (a) overlapping class distributions, or involve (b) complex decision boundaries, were class distributions do not overlap on decision boundaries.

(a) Data with overlapping class distributions:

$\mathbf{P_\mu(e_{tot})}$: As represented in Figure 2(a), this data consists of two classes, each one defined by a multi-variate normal distribution in a two dimensional input feature space. It is assumed that data is randomly generated by sources with the same Gaussian noise. Both sources are described by variables that are independent and have equal variance σ^2, therefore distributions are hyperspherical. In fact, $P_\mu(e_{tot})$ represents 13 data sets, where the degree of overlap between classes differs for each set. The degree of overlap is varied from a total probability of error, e_{tot} = 1% to e_{tot} = 25% (with increments of 2%), by adjusting the mean vector $\boldsymbol{\mu}_2$ of class 2. Table I presents the specific parameters employed to create the 13 $P_\mu(e_{tot})$ data sets.

$\mathbf{P_\sigma^{\,2}(e_{tot})}$: As represented in Figure 2(b), this data is identical to $P_\mu(e_{tot})$, except that the degree of overlap between classes is varied by adjusting the variance $\boldsymbol{\sigma_2^2}$ of both classes. Table II presents the parameters employed to create the 13 $P_\sigma^{\,2}(e_{tot})$ data sets. Note that for a same degree of overlap, $P_\sigma^{\,2}$ data sets have a larger overlap boundary than P_μ yet they are not as dense.

(b) Data with complex decision boundaries:

$\mathbf{P_{cis}}$: As represented in figure 2(c), the circle-in-square problem requires a classifier to identify the points of a square that lie inside a circle, and those that lie outside a circle. The circle's area equals half of the square [3]. It consists of one complex decision boundary where classes do not overlap.

$\mathbf{P_2}$: As represented in figure 2(d), each decision region of the P_2 problem is delimited by one or more of the four following polynomial and trigonometric functions:

$$f1(x) = 2 * \sin(x) + 5$$

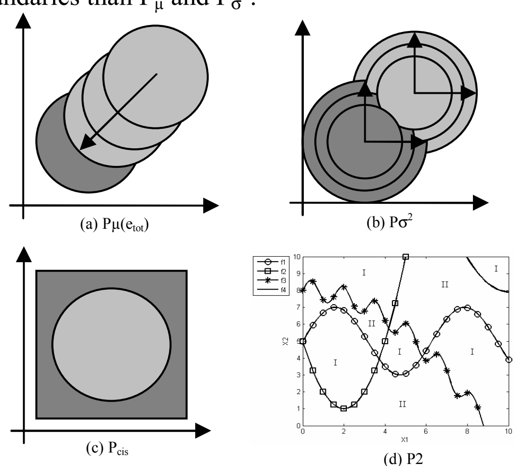
$$f2(x) = (x-2)^2 + 1$$

$$f3(x) = -0.1 \cdot x^2 + 0.6\sin(4x) + 8$$

$$f4(x) = \frac{(x-10)^2}{2} + 7.902\,,$$

and belongs to one of the two classes, indicated by the Roman numbers I and II [18]. It consists of four complex boundaries, and class definitions do not overlap. Note that equation $f_4(x)$ was slightly modified from the original equation such that the area occupied by each class is approximately equal.

All four data sets described above are composed of a total of 30,000 randomly-generated patterns, and correspond to 2 class problems, with a 2 dimensional input feature space. In addition, the numbers of patterns per class, as well as the area occupied by each class are equal. Note however that with P_{cis} and P_2 data sets, the length of decision boundaries between class distributions is longer, and fewer training patterns are available in the neighborhood of these boundaries than P_μ and P_σ^2.



(a) $P\mu(e_{tot})$ (b) $P\sigma^2$ (c) P_{cis} (d) P2

Figure 2 - Representation of the data sets used for computer simulations.

TABLE I - Parameters used to generate the Pμ data sets.

Data set	μ_1	μ_2	σ_1^2 and σ_2^2
$P_\mu(e_{tot} = 1\%)$	(0, 0)	(3.290, 3.290)	(1, 1)
$P_\mu(e_{tot} = 3\%)$	(0, 0)	(2.660, 2.660)	(1, 1)
$P_\mu(e_{tot} = 5\%)$	(0, 0)	(2.326, 2.326)	(1, 1)
$P_\mu(e_{tot} = 7\%)$	(0, 0)	(2.087, 2.087)	(1, 1)
$P_\mu(e_{tot} = 9\%)$	(0, 0)	(1.896, 1.896)	(1, 1)
$P_\mu(e_{tot} = 11\%)$	(0, 0)	(1.735, 1.735)	(1, 1)
$P_\mu(e_{tot} = 13\%)$	(0, 0)	(1.593, 1.593)	(1, 1)
$P_\mu(e_{tot} = 15\%)$	(0, 0)	(1.466, 1.466)	(1, 1)
$P_\mu(e_{tot} = 17\%)$	(0, 0)	(1.349, 1.349)	(1, 1)
$P_\mu(e_{tot} = 19\%)$	(0, 0)	(1.242, 1.242)	(1, 1)
$P_\mu(e_{tot} = 21\%)$	(0, 0)	(1.141, 1.141)	(1, 1)
$P_\mu(e_{tot} = 23\%)$	(0, 0)	(1.045, 1.045)	(1, 1)
$P_\mu(e_{tot} = 25\%)$	(0, 0)	(0.954, 0.954)	(1, 1)

TABLE II - Parameters used to generate the P_σ^2 data set.

Data set	μ_1	μ_2	σ_1^2 and σ_2^2
$P_\sigma^2\ (e_{tot} = 1\%)$	(0, 0)	(3.290, 3.290)	(1, 1)
$P_\sigma^2\ (e_{tot} = 3\%)$	(0, 0)	(3.290, 3.290)	(1.530, 1.530)
$P_\sigma^2\ (e_{tot} = 5\%)$	(0, 0)	(3.290, 3.290)	(2.000, 2.000)
$P_\sigma^2\ (e_{tot} = 7\%)$	(0, 0)	(3.290, 3.290)	(3.485, 2.485)
$P_\sigma^2\ (e_{tot} = 9\%)$	(0, 0)	(3.290, 3.290)	(3.011, 3.011)
$P_\sigma^2\ (e_{tot} = 11\%)$	(0, 0)	(3.290, 3.290)	(3.597, 3.597)
$P_\sigma^2\ (e_{tot} = 13\%)$	(0, 0)	(3.290, 3.290)	(4.266, 4.266)
$P_\sigma^2\ (e_{tot} = 15\%)$	(0, 0)	(3.290, 3.290)	(5.038, 5.038)
$P_\sigma^2\ (e_{tot} = 17\%)$	(0, 0)	(3.290, 3.290)	(5.944, 5.944)
$P_\sigma^2\ (e_{tot} = 19\%)$	(0, 0)	(3.290, 3.290)	(7.022, 7.022)
$P_\sigma^2\ (e_{tot} = 21\%)$	(0, 0)	(3.290, 3.290)	(8.322, 8.322)
$P_\sigma^2\ (e_{tot} = 23\%)$	(0, 0)	(3.290, 3.290)	(9.914, 9.914)
$P_\sigma^2\ (e_{tot} = 25\%)$	(0, 0)	(3.290, 3.290)	(11.90, 11.90)

To observe the impact on overtraining incurred by data normalisation, these four data sets were normalized according to both the min-max and Gaussian techniques. Min-max normalisation is described by:

$$a_i' = \frac{a_i - \min_i}{\max_i - \min_i} \qquad \textbf{(1)}$$

where a_i' and a_i represent the ith feature of the normalized and non-normalized pattern, respectively, $\min_i$ and $\max_i$ are the minimum and the maximum value of the ith feature contained in the entire data set. Gaussian normalization is described by:

$$a_{i,k}' = \frac{a_{i,k} - \mu_i}{3 \cdot \sigma_i} \qquad \textbf{(2)}$$

where $a_{i,k}'$ and $a_{i,k}$ represent ith feature of the kth pattern for the normalized and non-normalized pattern respectively, μ_i and σ_i are the mean and the variance of the i feature for all the data set. This normalization technique guarantees that approximately 99% of pattern will fall in [-1, 1]. Patterns falling outside this range are set to the closets boundaries. MinMax normalisation is then use to set pattern range into [0, 1].

Prior to simulations, each data set was normalized, and partitioned into three equal parts: the training subset, the validation subset, and the test subset. Each subset was composed of an equal number of 5,000 patterns per class. To assess the extent of overtraining caused by typical training strategies, this data was used to compare the performance of fuzzy ARTMAP neural networks trained according to the 1EP, $CONV_w$ and $CONV_p$, and CV. Since the CV strategy does not suffer from overtraining based on the number of training epochs, the impact of the number of training epochs on overtraining may also be assessed.

To observe the effect of training subset size on overtraining, the number of training subset patterns used for supervised learning was progressively increased from 10 patterns to 10,000 patterns according to a logarithmic rule. That is, the first simulation trial used the first 5 patterns per class, the second simulation trail used the first 6 patterns per class, and so on. This corresponds to 30 different simulation trials over the entire 10,000 pattern training subset. In addition, since fuzzy ARTMAP is sensitive to the presentation order of the training data, these 30 trials was repeated 10 times with different randomly generated data sets. Pattern presentation orders were always held constant from one epoch to the next.

After each simulation trial, the performance of fuzzy ARTMAP was assessed in terms of resources required during training, and its generalization error on the test sets. The amount of resources required during training is measured by compression and convergence time. *Compression* refers to the average ratio of training patterns to category prototypes created in the F2 layer. *Convergence time* is the number of epochs required to complete learning for a training strategy. It does not include data presentations of the validation set used for cross validation. *Generalization error* is estimated as the ratio of correctly classified test subset patterns over all test set patterns. Average results, with corresponding standard error, are always obtained, as a result of the 10 independent simulation trials (from the 10 randomly generated data sets).

The Quadratic Bayes and *k*-Nearest-Neighbour classifiers were included for reference with generalisation error results. These are classic parametric and non-parametric classification techniques from statistical pattern recognition, which are immune to the effects of overtraining. Cross-validation was employed to select the *k* parameter in *k*NN. For each computer simulation, the value of k was selected among k = 1, 3, 5, 7, and 9 based on the validation subset.

IV. SIMULATION RESULTS AND DISCUSSIONS

Figure 3 shows an example of the average generalization error, compression and convergence time achieved by fuzzy ARTMAP as a function of the training subset size for $P_\mu(e_{tot}=9\%)$. These results are shown for all four training strategies: 1EP, $CONV_w$, $CONV_p$, and CV. For reference, the generalization errors for the Quadratic Bayes and kNN classifiers, well as the analytical probability of error (e_{tot}) value, are also shown.

These results on this figure clearly indicate the effects of overtraining due to a number of factors. Fuzzy ARTMAP trained with CV achieves a lower average generalization error than if it were trained with $CONV_w$ or $CONV_p$. This effect diminishes as the training set size grows. In addition, CV yields an average compression and convergence time between that obtained by 1EP and by $CONV_w$ or $CONV_p$ training strategies. Since the CV strategy allows to avoid overtraining based on the number of epochs, the relatively higher generalisation error and lower compression obtained with $CONV_w$ and $CONV_p$ are an indication that these training strategies can in fact lead to overtraining based on the number of epochs.

The larger the training data set, the lower the impact of overtraining due to the number of training epochs. However, beyond a training set size of about 50 patterns per class, a significant degradation in performance occurs. The compression tends to stabilize, while the generalization error and convergence time tend to increase significantly, regardless of the training strategy. (Compression and convergence time do not degrade for 1EP.) This combined behaviour highlights the strong impact of overtraining related to the training set size. Increasing the amount of training data beyond a certain point, for data with overlapping class distributions, requires significantly more resources, while yielding a higher generalisation error.

Similar tendencies are found in simulation results where fuzzy ARTMAP is training using different $P_\mu(e_{tot})$ and $P_{\sigma^2}(e_{tot})$ data sets. However, as e_{tot} increases, the performance degradation due to training set size becomes more pronounced, and occurs for fewer training set patterns. With $P_\mu(e_{tot})$ and $P_{\sigma^2}(e_{tot})$ data sets, the generalisation error of fuzzy ARTMAP may only approach those of the reference classifiers for very small data sets. No significant differences were found in results obtained using either the min-max or Gaussian normalization techniques on the same data set.

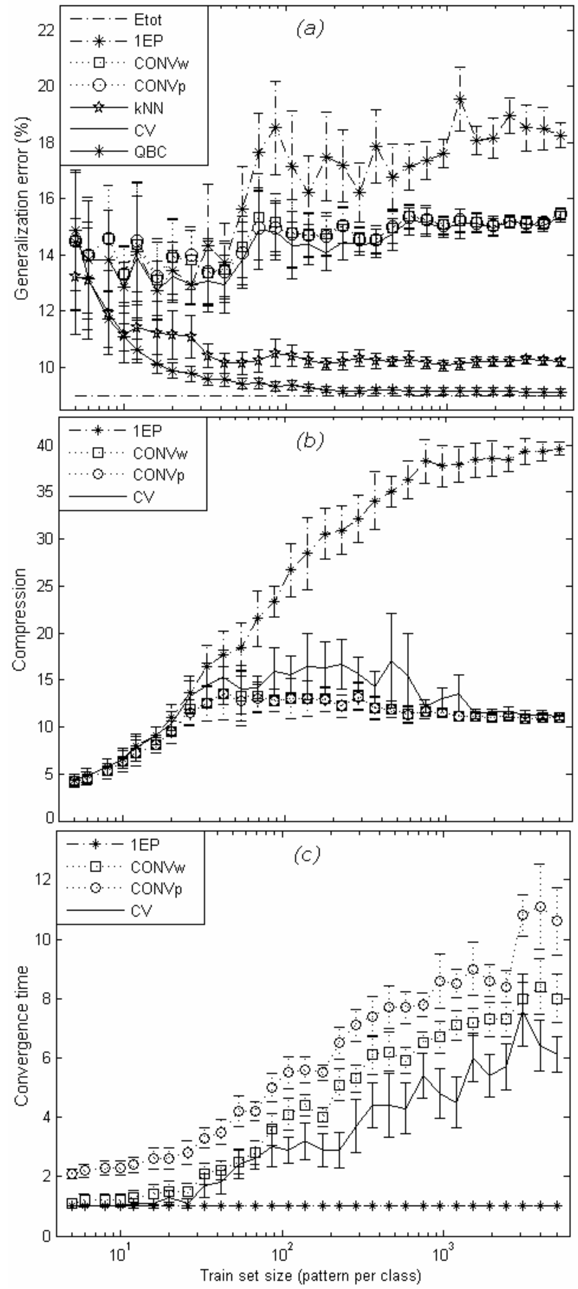


Figure 3 - Average (a) generalization error, (b) compression, and (c) and convergence time of fuzzy ARTMAP versus the training set size, for Pμ(e=9%).

Figure 4 shows the average generalization error, compression rate and convergence time achieved by fuzzy ARTMAP as a function of the training subset size for P_{cis}. These results are displayed for all four training strategies, along with those of kNN for reference. The effects of overtraining are not apparent from results on this figure. As

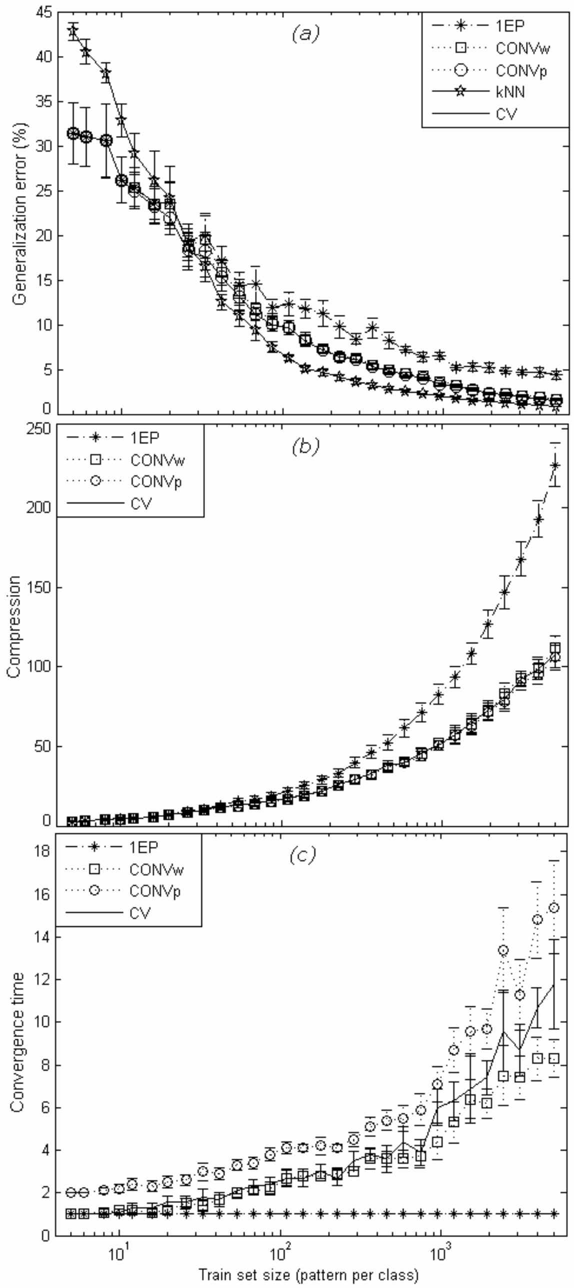


Figure 4 - Average (a) generalization error, (b) compression, and (c) and convergence time of fuzzy ARTMAP versus the training set size, for $P_{cis.}$

would be expected, when training set size increases, the compression and convergence time also increase. Meanwhile, the generalisation error of fuzzy ARTMAP decreases asymptotically to its minimum. Results are comparable for $CONV_w$, $CONV_p$, and CV training strategies. Similar tendencies are found in simulation results where fuzzy ARTMAP is trained using the P_2 data set. However, since the decision boundaries are more complex with P_2, a greater number of training patterns are required for fuzzy ARTMAP to asymptotically reach its minimum generalisation error. With P_{cis} and P_2 data sets, the generalisation errors of fuzzy ARTMAP are comparable to those of the reference *k*NN classifier. Again, no significant differences were found in results obtained by using the min-max and Gaussian normalization technique on the same data set.

One can conclude from our results that overtraining in fuzzy ARTMAP is linked to its sequential learning of training patterns from overlapping class distributions. Figure 5 presents an example of a decision boundary obtained when fuzzy ARTMAP is trained through cross-validation on 100 and then on 10,000 training patterns of the $P_\mu(e_{tot}=9\%)$ data set. Fuzzy ARTMAP tends to create a growing number of small prototype categories (hyper-rectangles), and forms granular decision boundaries as it attempts to learn overlapping class distributions. However, increasing the number of training patterns, increases the amount of resources needed to learn overlapping data, but does not necessarily improve discrimination. This is not an issue for data from non-overlapping class distribution, even when complex decision boundaries must be created - performance tends to increase with the amount of training data.

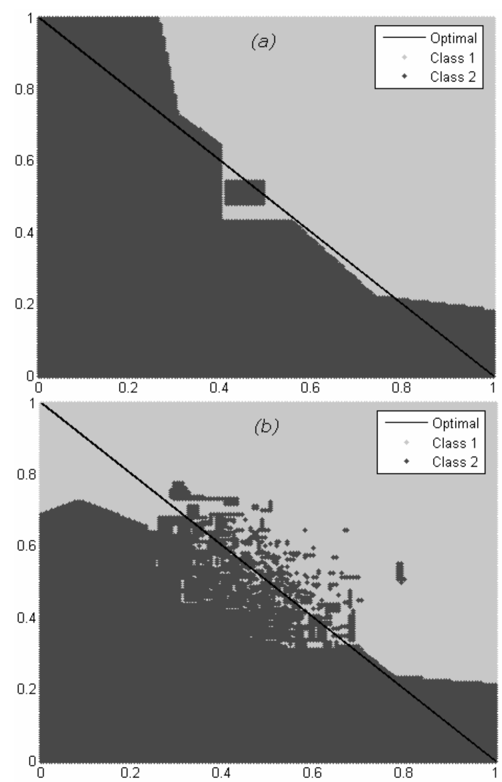


Figure 5 - Decision boundary formed by fuzzy ARTMAP in the input pattern space. Training is performed with cross-validation (a) on 100 patterns and (b) on 10,000 patterns of the $P_\mu(e_{tot}=9\%)$ data set. The optimal decision boundary for $P_\mu(e_{tot}=9\%)$ is also shown for reference.

Let us define the overtraining error from the effect of training set size as the difference between the maximum and the minimum generalization error obtained with CV. (Recall that CV eliminates the generalization error due to the number of epochs.) Figure 6 displays the average overtraining error from the effect of the training set size for P_μ as a function of e_{tot}. It also shows the compression and the number of training patterns required to achieve the best average generalization error. For instance, with the $P_\mu(e_{tot}=9\%)$ data set, the size of the training data set accounts for about 7% of the average overtraining error. In general, better fuzzy ARTMAP networks are obtained when less then 20 categories are created, and with a training set size less than 50 patterns per classes.

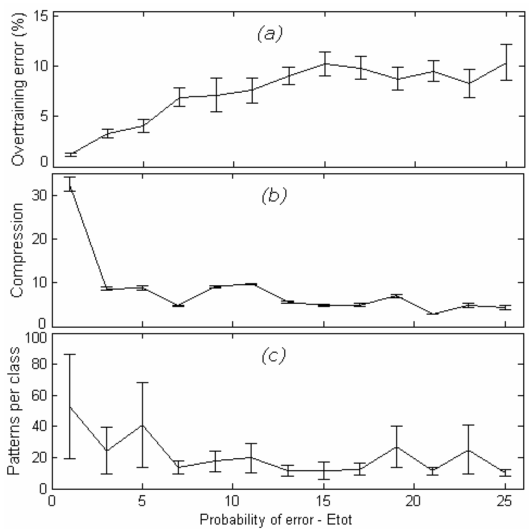


Figure 6 - (a) Average overtraining error from the training set size, and (b) compression and (c) number of pattern per class associated with the best average generalization error for fuzzy ARTMAP, as a function e_{tot}. Training is performed with cross-validation on the 13 $P\mu(e_{tot})$ data sets.

V. CONCLUSION

In this paper, the extent to which fuzzy ARTMAP neural networks can overtrain due to factors such as the data set structure, the training strategy, the number of training epoch, the normalisation of data, and the size of the training data set, as been explored. An experimental protocol has been defined such that the contribution of these factors may be isolated for pattern recognition problems consisting of overlapping class distributions, and involving complex decision boundaries with no overlap. During computer simulations, fuzzy ARTMAP networks have been trained for one complete epoch, through cross-validation, and until convergence. The performance of these networks has been assessed in term of their generalisation error on test sets, and by the amount of resources required during training.

The results presented in this paper indicate that fuzzy ARTMAP can in fact lead to overtraining. In particular, significant degradation in fuzzy ARTMAP performance due to overtraining is shown for pattern recognition problems with overlapping class distributions. In this context, performance degradation is linked to granular decision boundaries. Overtraining error tends to increases according the number of training epochs, and to the training set size for overlapping data, depending on the specific training strategy. In any case, training fuzzy ARTMAP using cross-validation always allows to achieve a performance level equal to, of higher than training it until convergence. Furthermore, cross-validation is shown to effectively reduce overtraining caused by the number of training epochs with overlapping data. Applying cross-validation within epochs would reduce overtraining from training set size, but would be computationally expensive.

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., & Rosen D.B., "Fuzzy ART: Fast Stable Learning and Categorizing of Analog Patterns by an Adaptive Resonance System", *Neural Networks*, 4(6), 759-771, 1991.

[3] Carpenter G.A., Grossberg S., & Reynold J.H., "ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organising Neural Network", *Neural Networks*, 4, 565-588, 1991.

[4] Carpenter, G.A., Grossberg S., Markuzon N. Reynolds J.H. & 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] Carpenter G.A. and Tan A.H, "Rule Extraction, Fuzzy ARTMAP and Medical Databases", *Proceeding, World Congress on Neural Networks*, Vol. I, pp.501-506, 1993.

[6] Carpenter, G.A., "Distributed ART Networks for Learning, Recognition and Prediction", *World Congress on Neural Networks*, *International Neural Network Society 1996 Annual Meeting*, p 333-44, 1996.

[7] Carpenter, G.A., Gjaja, M.N., Gopal, S., & Woodcock, C.E., "ART Neural Networks for Remote Sesing: Vegetation Classification from Landsat TM and Terrain Data", *IEEE transaction on geosciences and remote sensing*, vol. 35, no. 2, 1997.

[8] Georgiopoulos, M., Koufakou, A., Anagnostopoulos, G.C., & Kasparis, T., "Overtraining in Fuzzy ARTMAP: Myth or Reality?", *Proceedings of the International Joint Conference on Neural Networks*, Vol. 2, pp. 1186-1190, 2001.

[9] Gomez Sanchez, E., Dimitriadis Y.A., Cano-Izquierdo, J.M., & Coronado, J.L., "MicroARTMAP: Use of Mutual Information for Category Reduction in Fuzzy ARTMAP". *Proceedings of the International Joint Conference on Neural Networks*, vol. 6, 47-52, 2000.

[10] Granger, E., Rubin, M., Grossberg, S., & Lavoie, P., "A What-and-Where Fusion Neural Network for Recognition and Tracking of Multiple Radar Emitters", *Neural Networks*, 14, 325-344, 2001.

[11] Koufakou, A., Georgiopoulos, M., Anagnostopoulos, G., & Kasparis, T., "Cross-Validation in Fuzzy ARTMAP for Large Databases", *Neural Networks*, 14, 1279-1291, 2001.

[12] Lerner B., & Vigdor B., "An Empirical Study of Fuzzy ARTMAP Applied to Cytogenetics", *IEEE Convention of Electrical & Electronics Engineers in Israel*, 301-304, 2004.

[13] Murshed N.A., Bortolozzi F., Sabourin R., "A Cognitive Approach to Signature Verification. International Journal of Pattern Recognition and Artificial Intelligence", Special issue on Bank Cheques Processing, vol. 11, no. 7, 1997, pp. 801-825, 1997.

[14] Sumathi, S., Sivanandam, S.N., Jagadeeswari., "Design of Soft Computing Models for Data Mining Applications", *Indian Journal of Engineering and Materials Sciences*, v 7, n 3, p 107-21, 2000.

[15] Stone, M., "Cross-Validatory Choice and Assessment of Statistical Predictions", *Journal of the Royal Statistical Society*, 111-47, 1974.

[16] Stone, M., "Asymptotics For and Against Cross-Validation", *Biometrica* 64 , 39-35, 1977.

[17] Tay, Y. H., Khalid, M., Tan, K. K., & Yusof, R., "Hand-Written Postcode Recognition by Fuzzy ARTMAP Neural Network", *COSTAM National Science Congress*, Genting Highlands, November, 1997.

[18] G. Valentini., "Ensemble Methods Based on Bias–Variance Analysis", University of Genova, DISI-TH-2003-June 24, 2003.

[19] Verzi, S.J., Heileman, G.L., Georgiopoulos, M., & Healy, M.J., "Boosted ARTMAP", *IEEE International Joint Conference on Neural Networks Proceedings*, vol. 1, 396-401, 1998.

[20] Williamson, J. R., "Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps", *Neural Networks*, 9 (5), 881-897, 1996.