Proceedings of the 2009 IEEE Symposium on Computational Intelligence
in Security and Defense Applications (CISDA 2009)

Incremental Adaptation of Fuzzy ARTMAP Neural Networks for Video-Based Face Classification

Jean-François Connolly, Eric Granger, and Robert Sabourin

Abstract*—In many practical applications, new training data is acquired at different points in time, after a classification system has originally been trained. For instance, in face recognition systems, new training data may become available to enroll or to update knowledge of an individual. In this paper, a neural network classifier applied to video-based face recognition is adapted through supervised incremental learning of real-world video data. A training strategy based on particle swarm optimization is employed to co-optimize the weights, architecture and hyperparameters of the fuzzy ARTMAP network during incremental learning of new data. The performance of fuzzy ARTMAP is compared under different class update scenarios when incremental learning is performed according to 3 cases – (*A*) hyperparameters set to standard values, (*B*) hyperparameters optimized only at the beginning of the learning process with all classes, and (*C*) hyperparameters re-optimized whenever new training data becomes available. Overall results indicate that when samples from each individual enrolled to the system are employed for optimization, a higher classification rate is achieved and the solutions produced are more robust to variations caused by pattern presentation order. When all classes are refined equally, this is true with incremental learning according to case (*C*), whereas, if one class is refined at a time, best performance is obtained with case (*B*). However, optimizing hyperparameters requires more resources: several training sequences are needed to find the optimal solution and fuzzy ARTMAP with hyperparameters optimized according to classification rate tends to generate a high number of category nodes over longer convergence time.

I. Introduction

Although an easy task for humans, face recognition using machines can be a daunting task. With the availability of affordable and high performance technologies, a wide range of commercial and law enforcement applications are now possible, and face recognition is emerging as one of the most active areas of research in biometrics [1]. *Video-based* face recognition has the advantage that it does not require the cooperation of individuals involved in the process. As such, it is often used in video surveillance, where the system attempts to discreetly recognize individuals present in a scene, against a watch list. Still, this kind of application remains very challenging.

Face recognition systems typically employ statistical or neural pattern classifiers to map an $\Re^M$ input feature space to a set of predefined class labels that correspond to an individual's face models. The performance of those classifiers depends heavily on the availability of representative learning data sampled from, a priori, unknown class probability distributions in the feature space. For various applications, the collection and analysis of such data is expensive and time consuming. Learning data may therefore be incomplete in one of several ways: a limited number of observations, missing components of the input observations, missing class labels during training, and unfamiliar classes (not present in the training data set) [2]. In addition, previously acquired data may eventually become obsolete. Moreover, training samples acquired from *unconstrained* scenes are generally of poor quality with low resolution. They are also subject to considerable variations due to lack of control over acquisition conditions (e.g. illumination, pose, occlusion, orientation), and aging of individuals. These challenges make for very complex class distributions, mainly due to inter and intraclass variability, and to dynamically changing environment during system operations.

In many practical applications, new learning data is acquired after a classification system has originally been trained. Ideally, as new learning data becomes available, a video-based face recognition system should adapt to this information. The two practical scenarios that require adaptation of video-based face recognition systems are enrollment (new classes are added to the system) and update (pre-existing classes are refined using the new data). Adapting a classifier to new data under these scenarios should not require relearning from the start using all the cumulative learning data.

The majority of statistical and neural pattern classifiers proposed in literature perform *supervised batch learning* of a finite data set. To account for new data, they must relearn from the start using all previously-acquired learning data accumulated and stored in memory. In contrast, assuming that new data is available, a *supervised incremental learning* should (1) allow learning of 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 the new data [3]. Some classifiers proposed in literature are inherently able to perform supervised incremental learning (ARTMAP, Growing Self-Organizing Networks) [4], [5], while other well known neural networks (MLP, SVM, and RBF) have also been modified to perform such learning [6]–[8]. For this study, we focus on ARTMAP neural networks classifiers. These classifiers adapt their parameters (e.g. synaptic weights for a neural network) and their topology during incremental learning of new data.

In this paper, an adaptive classification system is proposed for supervised incremental learning of new blocks of data.

The authors are with the LIVIA (Laboratoire d'imagerie, de vision et d'intelligence artificielle), Department of Automated Production, École de technologie supérieure, Montréal, Québec, Canada. Email addresses: jfconnolly@livia.etsmtl.ca, Eric.Granger@etsmtl.ca, and Robert.Sabourin@etsmtl.ca.

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

978-1-4244-3764-1/09/$25.00 ©2009 IEEE

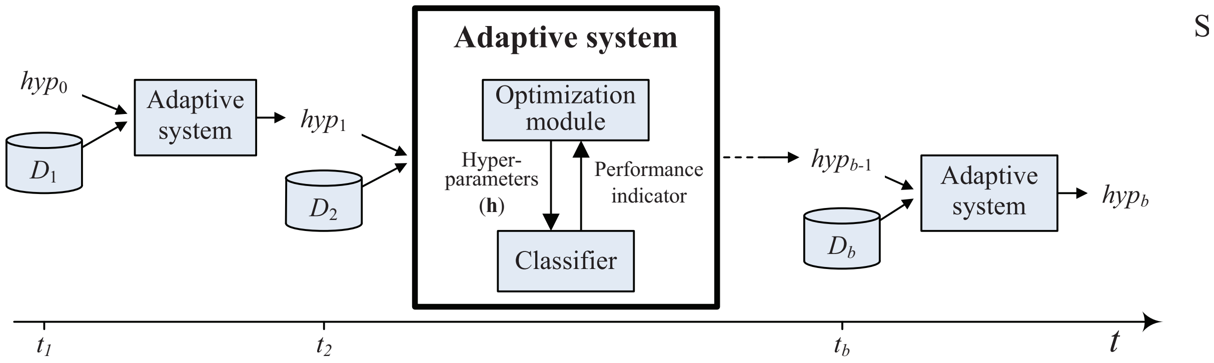


Fig. 1. An incremental learning scenario where new blocks of data are used by an adaptive system to update a classifier over time. Let D_b be blocks of learning data available to the classifier at different instants in time. The system starts with an initial hypothesis hyp_0, which constitutes the prior knowledge of the domain, that gets updated by the adaptive system on the basis of each D_b. Adaptation is defined by an iterative process where the optimization method finds the optimal vector of hyperparameters **h**. With each try, the optimization method gets a response from the classifier asserting the vector's performances.

Given a new block of learning data, it co-jointly optimizes the parameters, architecture and user-defined hyperparameters such that classification rate is maximized. To illustrate the benefits of this system, fuzzy ARTMAP [9] hyperparameters are optimized along with network architecture and weights using a particle swarm optimization learning strategy [10]. This study focuses on the surveillance scenario in which the model of individuals enrolled in a watch list is updated from new data. Classes are either refined one at a time, or all at the same time. Performance of the classification system is compared when fuzzy ARTMAP is trained using (*A*) hyperparameters set to standard values, (*B*) through optimization of hyperparameters only on the first data block, and (*C*) through optimization of hyperparameters on each new data block.

In the next section, the system for adaptive classification is presented, along with the fuzzy ARTMAP neural net used for classification, and the particle swarm optimization algorithm used to adjust its hyperparameters. Then, the data base and the experimental protocol are described in section III. Finally, experimental results are presented and discussed in section IV.

II. Adaptive classification

Fig. 1 depicts the evolution of an adaptive classification system in the context of the incremental learning scenario discussed in this paper, and that may appear in video-based face recognition applications. Unlike on-line learning, it assumes that the new learning data is collected and assembled into blocks. The adaptive classification system consist of a pattern classifier interacting with a module for the optimization of its hyperparameters. When a new block of learning data D_b becomes available, the classifier enters a supervised incremental learning state, and trains on the new data over several epochs. The optimization module is then triggered to adapt classifier's user-defined hyperparameters to D_b. In this work, the classifier is a fuzzy ARTMAP neural network [9] and optimization is performed through particle swarm optimization (PSO) [11].

Most optimization techniques applied to fuzzy ARTMAP hyperparameters found in literature have been developed for batch supervised learning, and allow to optimize only one or two hyperparameters, even though there are four inter-dependent parameters [12]–[14]. In previous work, the authors have introduced a PSO-based learning strategy for mono-objective optimization of all four hyperparameters [10]. It is based on the concept of neural network evolution in that it determines the optimal vector hyperparameters and network weights and architecture such that classification rate is maximized. The PSO strategy has been shown to provide a higher classification rate on synthetic and real-world data [10], [15].

The optimization method seeks to find an optimal vector of hyperparmeters **h** by optimizing the network's performances during incremental learning of new data blocks D_b. Each time a vector **h** is proposed by the optimization module, the classifier estimates a classification rate for **h**, either confirming, or invalidating its optimality. For optimization algorithms using a population of solutions, such as PSO, the optimization method can then decide the next step in the optimization process based on the knowledge of the entire swarm and preserve the best solution in memory. Fuzzy ARTMAP can there by adapt, not only its architecture and weights, but also its hyperparameters, and thus its internal learning dynamics. The rest of this section provides additional details on fuzzy ARTMAP and the PSO learning strategy.

A. Fuzzy ARTMAP

ARTMAP refers to a family of self-organizing neural network architectures that is capable of fast, stable, on-line, unsupervised or supervised, incremental learning, classification, and prediction [4]. The fuzzy ARTMAP integrates the fuzzy ART to process both analog and binary-valued input patterns to the original ARTMAP architecture [9]. It consist of three layers: (1) an input layer F_1 of $2M$ neurons (for a $\Re^M$ input feature space), (2) a competitive layer F_2, where each node is associated to a category in the feature space, and (3) a map field F_{ab} of k neurons (the number of classes).

Each input **a** learns to *predict* an output class K. During training, the network creates internal recognition categories, with the number of categories determined on-line by predictive success. Components of the vector $\mathbf{a}$ are scaled so that each $a_i \in [0,1]$, $(i = 1 \ldots M)$. Complement coding doubles the number of components in the input vector, which becomes $\mathbf{A} \equiv (\mathbf{a}, \mathbf{a}^c)$, where the i^{th} component of $\mathbf{a}^c$ is $a_i^c \equiv (1 - a_i)$. With fast learning, the weight vector $\mathbf{w}_j$ records the largest and smallest component values of input vectors placed in the j^{th} category. The $2M$-dimensional vector $\mathbf{w}_j$ may be visualized as the hyperbox R_j that just encloses all the vectors $\mathbf{a}$ that selected category j during training.

Activation of the coding field F_2 is determined by the Weber law choice function $T_j(\mathbf{A}) = |\mathbf{A} \wedge \mathbf{w}_j| / (\alpha + |\mathbf{w}_j|)$, where $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$ and $|\mathbf{p}| \equiv \sum_{i=1}^{2d} |p_i|$. With winner-take-all coding, the F_2 node J that receives the largest $F_1 \rightarrow F_2$ input T_j becomes active. Node J remains active if it satisfies the matching criterion: $|\mathbf{A} \wedge \mathbf{w}_J| / |\mathbf{A}| = |\mathbf{A} \wedge \mathbf{w}_J| / M > \rho$, where $\rho \in [0,1]$ is the dimensionless *vigilance parameter*. Otherwise, the network resets the active F_2 node and searches until J satisfies the matching criterion. If node J then makes an *incorrect* class prediction, it is deactivated until another pattern is presented to the network. A *match tracking* signal then adjusts vigilance and induce a new search, which continues until either some F_2 node becomes active for the first time, in which case J learns the correct output class label $k(J) = K$; or a node J that has previously learned to predict K becomes active. During testing, a pattern $\mathbf{a}$ that activates node J is predicted to belong to the class $K = k(J)$.

Fuzzy ARTMAP training and testing is governed by four user-defined hyperparameters: the choice parameter α, the learning parameter β, the match tracking parameter ϵ, and the baseline vigilance parameter $\bar{\rho}$. Each of these hyperparameters are inter-related, and has a distinct impact on network dynamics. While, α and ϵ influences the choice of the winning F_2 nodes during winner-takes-all competition, $\bar{\rho}$ will limit the maximum size of the categories and β determines how fast they are expanded to fit the new data. A standard vector of hyperparameters $\mathbf{h}_{std} = (\alpha, \beta, \epsilon, \bar{\rho}) = (0.001, 1, 0.001, 0)$ is commonly used to minimize network complexity [9].

A key feature of ARTMAP networks is their ability to learn new information incrementally, yet limit the problem of catastrophic forgetting. However, previous research by the authors has revealed that the average classification rate of an ARTMAP network trained through incremental learning is usually significantly lower than if trained on all the data through batch learning [16], [17].

B. PSO-based based learning strategy

Particle swarm optimization (PSO) is a population-based stochastic optimization technique that was inspired by social behavior of bird flocking or fish schooling [11]. With PSO, each particle corresponds to a single solution in the optimization space, and the population of particles is called a swarm. Particles move through the optimization space and change their course under the guidance of a cognitive influence (i.e. their own previous search experience) and a social influence (i.e. their neighborhood previous search experience).

In this paper, canonical PSO [18] is used with the general best topology to optimize the classification rate of fuzzy ARTMAP when learning new data blocks D_b. The optimization space is defined by the four fuzzy ARTMAP hyperparameters: $\alpha \in [0.001, 100]$, $\beta \in [0,1]$, $\epsilon \in [-1,1]$, and $\bar{\rho} \in [0,1]$, and a particle's position is the vector of hyperparameters $\mathbf{h} = (\alpha, \beta, \epsilon, \bar{\rho})$. The particles position are updated using

$$\begin{aligned} \mathbf{h}_i(t+1) = \mathbf{h}_i(t) &+ w_1 (\mathbf{h}_i(t) - \mathbf{h}_i(t-1)) \\ &+ rand_1(t) w_2/2 \, (p_{gbest} - \mathbf{h}_i(t)) \\ &+ rand_2(t) w_2/2 \, (p_{ibest} - \mathbf{h}_i(t)), \end{aligned} \tag{1}$$

where $\mathbf{h}_i(t)$ is the position of particle i at iteration t, w_1 and w_2 are two inertia weights, $rand_1(t)$ and $rand_2(t)$ are two random numbers generated at each iteration, p_{gbest} and p_{ibest} are respectively the position of the swarm's general best (social influence) and particle i personal best (cognitive influence).

Algorithm 1 describes the PSO-based learning strategy modified for supervised incremental learning of new data blocks D_b with fuzzy ARTMAP. This algorithm requires new learning data blocks D_b, a particle swarm with P particles, and $P+1$ fuzzy ARTMAP networks: one attached to each particle (*FAM*$_i$), used to preserve the model associated to the hyperparameter vector $\mathbf{h}_i$, and an optimal neural network (*FAM*$_{optimal}$), utilized to store the optimal set of hyperparameters and network *at the end* of the PSO algorithm.

First, at line 1, the swarm's parameters are set (population's size P, maximum velocity for each dimension $\vec{v}_{max}$, and

Algorithm 1 PSO learning strategy for incremental learning with fuzzy ARTMAP

```
Input: A particle swarm with parameters: $P$, $\mathbf{v}_{max}$, $w_1$, $w_2$,
    neural networks: $FAM_i$, where $1 \leq i \leq P$, $FAM_{optimal}$,
    and new data sets $D_b$ for learning.
Output: Classification rate of $FAM_{optimal}$
 1: Set the swarm parameters ($P$, $v_{max}$, $w_1$, $w_2$).
 2: Randomly initialize particles positions and velocities
        within their range.
 3: Initialize $FAM_{optimal}$ and all $FAM_i$, where $1 \leq i \leq P$.
 4: for each new $D_b$ do
 5:     Set PSO iteration counter at $t = 0$.
 6:     Randomly reinitialize particles position and velocity
            within their range.
 7:     while PSO did not reach stopping condition do
 8:         for each particle $p_i$, where $1 \leq i \leq P$ do
 9:             $FAM_i \leftarrow FAM_{optimal}$
10:             Train and estimate the classification rate of $FAM_i$
                    using data from $D_b$.
11:             Performance of $p_i \leftarrow FAM_i$'s classification rate
12:         end for
13:         Update position for each particle $p_i$ with (1).
14:         $t = t + 1$
15:     end while
16:     $FAM_{optimal} \leftarrow FAM_{gbest}$
17: end for
```

TABLE I

NUMBER OF LEARNING AND TEST PATTERNS PER INDIVIDUAL FOR THE NRC DATA BASE

Number of ROIs	Individuals											Total
	1	2	3	4	5	6	7	8	9	10	11	
Learning data	140	39	160	130	175	128	180	97	178	160	140	1527
Test data	142	40	159	131	186	134	190	100	188	168	147	1585

inertia weights w_1 and w_2). Each particle's position and velocity are then initialized randomly within its allowed range (line 2), and the neural networks used are also initialized (3). For each new block D_b, a new optimization process begins: the number of iteration is set to 0 and the particles are reinitialized randomly (lines 5 and 6). Unless a stopping criterion is reached, the PSO algorithm will iteratively evaluate each particle's performance, and updates the swarm accordingly. On line 9, the optimal network found from D_{b-1} ($FAM_{optimal}$) is used to initialize all networks FAM_i prior training (for the first learning block, $FAM_{optimal}$ will be in an initial state). For each particle, FAM_i is then trained and its performance is evaluated using data from D_b (line 10). Once the performance of all the swarm has been evaluated, the PSO algorithm computes the particles new position using (1) and updates the number of iteration. Finally, to preserve an optimal set of hyperparameters and network throughout the learning process, the FAM_{gbest} network, associated to the best vector of hyperparameters $\mathbf{h}_{gbest}$ is stored as $FAM_{optimal}$ (line 16).

III. EXPERIMENTAL METHODOLOGY

A. Data sets

In order to observe the impact of adapting fuzzy ARTMAP hyperparameters during supervised incremental learning, real-world video data for face recognition is considered. The National Research Council (NRC) data base [19] is composed of samples from eleven individuals sitting in front of a computer and contains a variety of challenging operational conditions such as motion blur, out of focus factor, facial orientation, facial expression, occlusion, and low resolution (each face occupies between $1/4$ to $1/8$ of the image). For each individual, 2 video sequences of about 15 seconds of color video were captured at a rate of 20 frames per seconds with a webcam mounted on a computer monitor. The video sequences are taken under approximately the same illumination conditions (no sunlight, only ceiling light evenly distributed over the room), the same setup and almost the same background, for all persons in the data base.

Face detection was performed using the the Viola-Jones algorithm included in the OpenCV C/C++ computer vision libraiy [20]. It produced regions of interest (ROIs) between 29×18 and 132×119 pixels. These ROIs are then converted in gray scale and normalized to 24×24 images where the eyes are detected and aligned horizontally, with a distance of 12 pixels between them. Of the two video sequences, one is dedicated to training and the other to testing. The number of ROIs detected per class for the NRC databased is displayed in Table I.

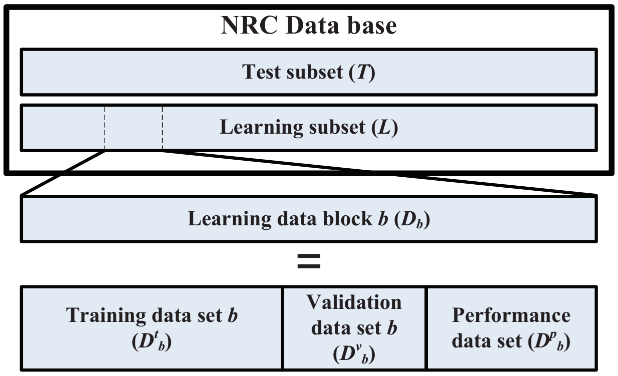

NRC Data base
Test subset (*T*)
Learning subset (*L*)
Learning data block *b* (D_b)
=
Training data set *b* (D^t_b)
Validation data set *b* (D^v_b)
Performance data set (D^p_b)

Fig. 2. For each learning block D_b, data are assigned to training of the network (D^t_b) with validation (D^v_b) and performance evaluation (D^p_b).

B. Experimental protocol

The learning data set L is organized to emulate incremental learning of B blocks of data, where each learning data block D_b is further divided into folds and organized to train fuzzy ARTMAP networks over several epochs using 10 folds cross-validation (Fig. 2). Out of the ten folds, seven are dedicated to training (D^t_b), one is used to validate the model and determine the number of fuzzy ARTMAP training epochs (D^v_b), and the remaining two are used to evaluate the performance of each particle during the PSO algorithm (D^p_b). These ten folds are alternated to perform ten replications.

During each simulation trial and after each D_b learned, performance is assessed with the test data set T for fuzzy ARTMAP trained using hyperparameters that

A) are set to standard values ($\alpha = 0.001$, $\beta = 1$, $\epsilon = 0.001$, and $\bar{\rho} = 0$) $\rightarrow \mathbf{h}_{std}$,

B) are optimized on only D_1 and then remain fixed $\rightarrow \mathbf{h}_1$,

C) are optimized on each learning block $D_b \rightarrow \mathbf{h}_b$.

Results obtained using the PSO learning strategy seek to maximize the classification rate with a population size of $P = 20$ particles, $w_1 = 0.7298$ and $w_2 = 2.992$. It is set to either stop after 100 iterations, or after 10 iterations without improvement to the $gbest$ classification rate.

During the PSO learning strategy, the impact of pattern presentation order on fuzzy ARTMAP is minimized by replacing lines $9 - 11$ of Algorithm 1 with Algorithm 2. A temporary neural network (FAM_{temp}) is used to define a particle's performance as the mean classification rate over D^p_b of five training sequences using the same data set D^t_b, but with different pattern presentation order, and to give the network trained with the best order presentation. For each random patterns presentation order of D^t_b, FAM_i is reinitialize with $FAM_{optimal}$ (line 3) and trained with validation (line 4), performance of FAM_i is estimated using D_b^p (line 5), and the best network so far is preserved in FAM_{temp}. Finally, FAM_i and the performance of p_i are properly defined (lines 10–11).

Although the hyperparameters are not necessarily optimized, Algorithm 2 is *always* applied to minimize the impact of patterns presentation order. In other words, even when $\mathbf{h}$ does not change ($\mathbf{h}_{std}$ and $\mathbf{h}_1$), the particles performance evaluation data set D_b^p is still used to find the best network out of the five replications.

The average performance of fuzzy ARTMAP was assessed in terms of classification rate and resources requirements. The amount of resources is measured by compression and convergence time. *Classification rate* is estimated as the ratio of incorrectly classified test subset patterns over all test set patterns, *compression* refers to the average number of training patterns per category prototype created in the F_2 layer, and *convergence time* is the number of training epochs required to complete learning. It does not include presentations of validation subsets used to perform cross-validation validation.

C. Class update scenarios

Refining one class at a time. The first experiment explores the case where a number of individuals are initially subscribed to the system, and, from time to time, faces of an individual is capture in a video sequence and then added to the system. All the classes are learned with the first data block D_1 and are refined one class at a time with blocks D_2 through D_{N+1}. Block D_1 is composed of 10% of the data for each class, and each subsequent block D_b, where $2 \leq b \leq N+1$, is composed of the remaining 90% of one specific class. To insure results invariance to class order presentation, this experimentation was done with five different random *class* presentation orders.

Refining all classes equally. The second update scenario studies the case where a face recognition system operator must refine the model of all individuals at the same time using the same amount of data for each individual. It is a near-ideal class update scenario with small learning blocks. Unlike the ideal update scenario, to use all data available, the learning data subset L is distributed as evenly possible, with respect to all classes, amongst learning blocks D_b. Within each D_b, classes are again distributed as evenly possible amongst the ten cross-validation folds. For some replications, some classes may be absent during either training, validation, or performance evaluation. In this paper, L is divided in ten learning blocs, meaning that about 10% of the data available are learned with each D_b.

The authors recognize that the balanced refinement scenario does not necessary correspond to a practical situation. However, this case was included in the experiment to observe the impact on local and global changes in the fuzzy ARTMAP network structure.

Algorithm 2 Particle performance evaluation.

```
Input: Best temporary network, $FAM_{temp}$.
Output: A particle's performance and the best neural network
    to obtained that performance.
1:  Initialize $FAM_{temp}$
2:  for 5 patterns presentation order do
3:      $FAM_i \leftarrow FAM_{optimal}$
4:      Train $FAM_i$ with validation using $D_b^t$ and $D_b^v$ .
5:      Estimate $FAM_i$ classification rate using $D_b^p$ .
6:      if classification is the best so far then
7:          $FAM_{temp} \leftarrow FAM_i$
8:      end if
9:  end for
10: $FAM_i \leftarrow FAM_{temp}$
11: $FAM_i$ performance $\leftarrow$ mean class. rate of the 5 replications
```

IV. Results and discussion

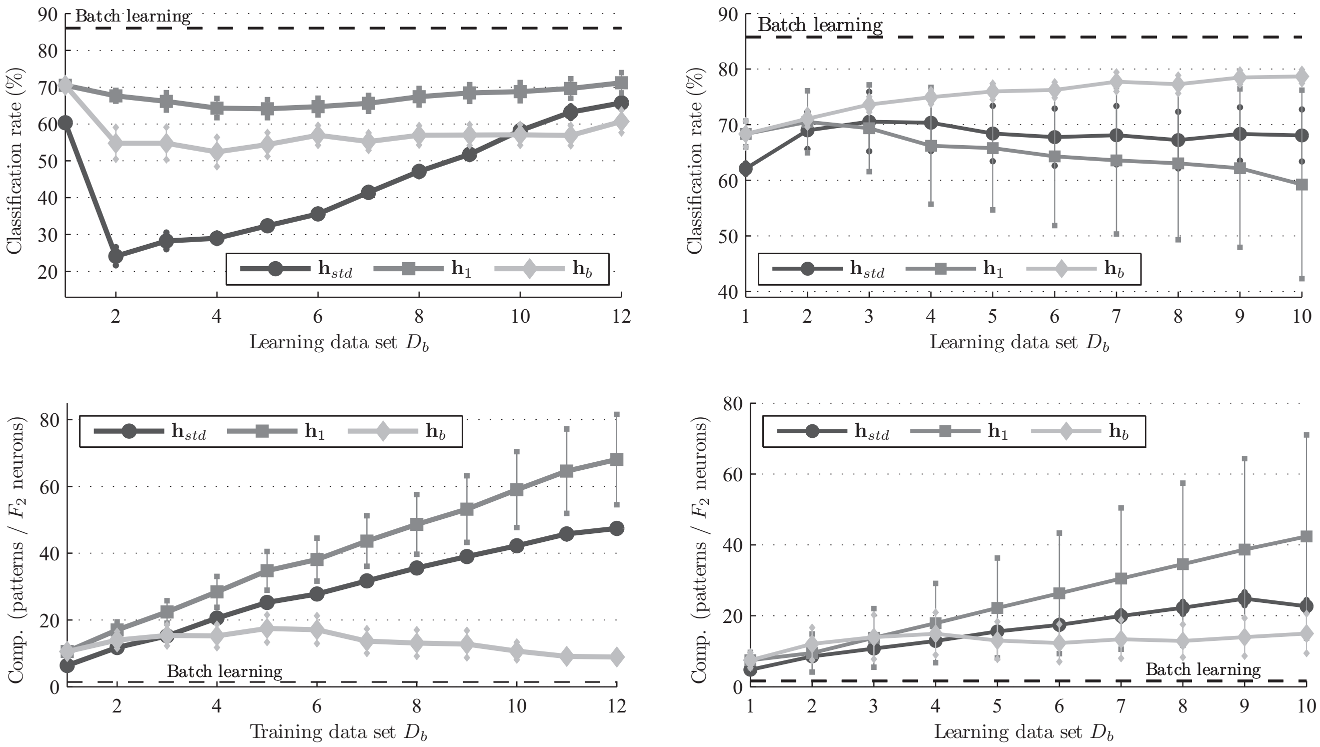
The average classification and compression rate achieved by fuzzy ARTMAP during incremental learning with $\mathbf{h}_{std}$, $\mathbf{h}_1$, and $\mathbf{h}_b$, for both learning scenarios, are presented in Fig. 3, while Table II shows the classification rate for each class after learning all training data. For reference, the average classification rate after learning all data with the PSO learning strategy in batch mode (i.e. all the data in one block) is 86±1%, with a compression of 1.6±0.4, and required 3.2±0.7 training epochs. This level of performance was achieved with the hyperparameter vector $\mathbf{h}_{batch}$ = (α, β, ϵ, $\bar{\rho}$) = (60 ± 10, 0.74 ± 0.07, 0.3 ± 0.2, 0.90 ± 0.04). Although the classification rate is high with the optimal setting, fuzzy ARTMAP required many F_2 nodes generated over several training epochs.

A. Standard hyperparameters ($\mathbf{h}_{std}$)

When classes are refined one at a time with $\mathbf{h}_{std}$, fuzzy ARMTAP classification rate tends to decrease from $60 \pm 1\%$ to $24 \pm 2\%$ after D_2 and, as other classes are presented, gradually increases to $66 \pm 2\%$ at D_{12}. Since 50 replications were performed in total (5 times 10 folds cross-validation), the confidence interval after D_1 is generally very small ($CI = 1$). While the impact of D_2 varies from replication to replication, leading to $CI = 3$, the classification rate stabilizes after D_4 ($CI = 1$). Meanwhile, compression increases linearly (from 6.4 ± 0.2 to 47 ± 1) and each D_b is learned with one training epoch.

As with the results using synthetic data [16], [17], when all classes are refined with each D_b, performance of fuzzy ARTMAP with $\mathbf{h}_{std}$ stabilizes early during incremental learning. Classification rate starts at $62 \pm 1\%$ at D_1, increases up to $70 \pm 5\%$ at D_2, and is maintained around $68 \pm 5\%$ from D_5 to D_{10}. Again, compression increases linearly (from 4.8 ± 0.3 to 23 ± 2) and the number of training epochs remains close to one.

As shown in Fig. 3, fuzzy ARTMAP learning algorithm with $\mathbf{h}_{std}$ requires few F_2 neurons and during training, the network becomes very sensitive to pattern presentation order and to classes presented in each learning data sets. When classes are updated one at a time, this makes for a considerable decline in classification rate for the first classes refined. On the other



(a) Refining one class at a a time

(b) Balanced refinement

Fig. 3. Average classification rate and compression rate of the fuzzy ARTMAP versus learning block during incremental learning for two learning scenarios: refining one class at a time (Fig. 3a) and balanced refinement (Fig. 3b). Performances were evaluated with $\mathbf{h}_{std}$, $\mathbf{h}_1$, and $\mathbf{h}_b$, and are shown with their 90% confidence interval.

hand, when all classes are updated at the same time, data complexity of each D_b is much higher and fuzzy ARTMAP classification rate is less stable, as more F_2 nodes are created.

When learning the NRC data base with fuzzy ARTMAP, Table II shows that performance degradation is mainly due to overspecializing the network in regard of certain classes. For example, several patterns from classes one, two, four and ten were classified as belonging to class five. No matter how classes are refined, after all ten blocks are learned, fuzzy ARTMAP classification rate and convergence time are comparable, while balanced refinement required more F_2 nodes, leading to lower compression.

B. Hyperparameters optimized on D_1 ($\mathbf{h}_1$)

When classes are refined one at a time and the PSO training strategy is applied only to D_1, fuzzy ARTMAP yields similar tendencies as with $\mathbf{h}_{std}$, albeit on a smaller scale. The classification rate, obtained with $\mathbf{h}_1$ = (α, β, ϵ, $\bar{\rho}$) = (53 ± 7, 0.7 ± 0.2, -0.3 ± 0.1, 0.5 ± 0.2), starts at $70 \pm 1\%$, decreases to $64 \pm 2\%$ at D_5, and then increases to $71 \pm 3\%$ after learing all data blocks. The confidence interval grows between D_1 and D_3, and remains between 2 and 3 for all other learning blocks. The compression rate also grows and is greater than with $\mathbf{h}_{std}$, but with a much larger confidence intervals (70 ± 10 after learning all data).

For balanced refinement, classification rate with $\mathbf{h}_1$ = (α, β, ϵ, $\bar{\rho}$) = (45 ± 15, 0.7 ± 0.1, -0.3 ± 0.2, 0.5 ± 0.2) starts higher than with $\mathbf{h}_{std}$ (68 ± 2), peaks at $71 \pm 1\%$ (D_2), and decreases to eventually reach $59 \pm 17\%$. As learning progresses, the confidence interval on the classification rate grows linearly up to 17 after D_{10}. Compression rate also increases linearly from 5 ± 3 to 40 ± 30, but since its confidence interval grows from one block to the next, it is always comparable to compression with $\mathbf{h}_{std}$.

In both scenarios, while most replications yielded a compression lower than $\mathbf{h}_{std}$, fuzzy ARTMAP generated very few F_2 neurons for some replications (some replications started *and* ended with only one F_2 neurons per class). For D_1, training converges after 1.9 ± 0.3 epochs. For all other learning blocks, except for few replications that needed up to three, only one training epoch was necessary.

These results shows that when $\mathbf{h}_1$ is properly defined on D_1 using data from all classes and remains fixed for subsequent blocks, complexity of the training data set (D_b^t) has more impact on network performances than using non-representative data (e.g. data coming from only one class). When classes are refined one by one using $\mathbf{h}_1$, data structure changes locally and only some missclassifications are introduced in class models as some individuals are refined. As all classes are refined, those errors are reduced and fuzzy ARTMAP improves. Moreover, the values of $\mathbf{h}_1$ lead to networks with a lower number of F_2 nodes that are capable of adapting properly to new classes. This leads to a maximal decline in classification rate of only 6% at D_5.

TABLE II

Average classification rate per class for different sets of hyperparameters ($\mathbf{h}_{std}$, $\mathbf{h}_1$, and $\mathbf{h}_b$). For the one class at time refinement, results are shown for only one class presentation sequence (out of five). Results are obtained after supervised incremental learning of all learning blocks and each cell is presented in percentage with the 90% confidence interval.

Learning scenarios	**h**	**Individual classes**											**All classes**
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	
Refining one class at a time	$\vec{h}_{\text{std}}$	46 ±4	7 ±2	94 ±1	36 ±3	84 ±2	86 ±1	73 ±2	47 ±2	84 ±1	48 ±3	72 ±2	66 ±2
	$\vec{h}_1$	88 ±3	61 ±5	90 ±5	72 ±3	50 ±6	89 ±2	75 ±5	80 ±2	81 ±1	66 ±2	88 ±2	71 ±3
	$\vec{h}_b$	47 ±9	41 ±7	96 ±2	54 ±7	33 ±6	64 ±10	90 ±2	40 ±8	79 ±1	64 ±6	51 ±9	61 ±3
Balanced refinement	$\vec{h}_{\text{std}}$	48 ±10	12 ±7	92 ±5	59 ±12	80 ±7	89 ±6	74 ±9	43 ±2	81 ±4	46 ±6	69 ±9	68 ±5
	$\vec{h}_1$	73 ±23	53 ±20	64 ±24	51 ±19	35 ±18	76 ±21	51 ±22	60 ±18	71 ±10	54 ±18	66 ±21	59 ±17
	$\vec{h}_b$	92 ±5	42 ±12	95 ±2	74 ±5	60 ±5	93 ±3	79 ±3	76 ±5	81 ±2	62 ±4	88 ±2	79 ±1

When all classes are refined with each block, data structure changes are on a more global scale and the hyperparameters defined with D_1 ($\mathbf{h}_1$) proves to be inadequate for learning the rest of the data. Classification rate constantly decreases, while variance increases. Moreover, after training with all data, Table II shows that patterns from classes that were dominant with $\mathbf{h}_{std}$ (classes three, five, six and nine) now have large proportion of misclassified patterns. Compared to class refinement, Table II shows that, for each individual class, global changes make for lower, or similar, classification rate and larger confidence interval.

C. Hyperparameters optimized on all D_b ($\mathbf{h}_b$)

With $\mathbf{h}_b$, the hyperparameters continue to be adjusted for each new learning block. When one class is updated at a time, the classification rate starts at $70 \pm 1\%$, decreases to $52 \pm 4\%$ at D_4, and increases until it reaches $61 \pm 3\%$ at D_{12}. During class refinement, learning with $\mathbf{h}_b$ yields the largest confidence interval: 4 at D_2 and D_3, and, except 2 at D_7, 3 afterward.

During balanced refinement, classification rate obtained with $\mathbf{h}_b$ improves over learning blocks, while the confidence interval always remains inferior than 2. The largest confidence interval, observed at D_1, is 1.4. Classification rate starts at $68 \pm 2\%$ on D_1 and ends at $79 \pm 1\%$ on D_{10}. For both learning scenarios, fuzzy ARTMAP generates many F_2 nodes with compressions rate that are slightly higher than their initial values after D_1. Again the number of training epochs is mostly one, while sometimes reaching up to three.

When hyperparameters are optimized on all D_b, Fig. 4 shows that, for balanced refinement, $\mathbf{h}$ changes significantly during learning. After the first learning bloc, $\mathbf{h}_1$ = (α, β, ϵ, $\bar{\rho}$) = (46 ± 15, 0.69 ± 0.09, -0.28 ± 0.18, 0.53 ± 0.17). Even though all confidence intervals remains large, two significant variations are observed for all hyperparameters, except $\bar{\rho}$. While, α decreases to 21 ± 7 at D_6, and increases to 37 ± 15 at D_8, β drops to 0.52 ± 0.14 at D_6, and rises to 0.65 ± 0.13 at D_9, and ϵ changes to 0.10 ± 0.20 at D_4, and again to -0.27 ± 0.18. If compared to its starting value at D_1, no significant deviations of $\bar{\rho}$ are observed. However, once $\rho = 0.65 \pm 18$ at D_5, it significantly decreases to 0.40 ± 16 at D_8, followed by 0.67 ± 0.12 at D_9, and 0.41 ± 0.09 at D_{10}.

Results with $\mathbf{h}_b$ show that the data sets used to guide the particles during optimization (D_b^p) must contain a representative set of samples from all classes to optimize the network properly. If D_b^p is composed of only one class (when classes are refined one at a time), $\mathbf{h}$ is optimized only for that specific class and, even though fuzzy ARTMAP network structure changes locally, the overall classification rate decreases considerably. As an example, classification rate for one sequence of class refinement is shown in Table III. When a new block is presented to fuzzy ARTMAP, the vector $\mathbf{h}_b$ is adjusted using only classes found in D_b, resulting in a classification rate that is very high for that class, yet low for all other classes. Once all classes are learned in this way, Table II indicates that for a specific sequence of class refinement, the latest classes that were learned tend to dominate the ones learned in previous D_b. Classes three (D_{11}), seven (D_{12}), and nine (D_9) give classification rates significantly better than the other classes.

In contrast, if hyperparamters are optimized with regards to all classes, fuzzy ARTMAP is able to adapt itself to global changes in the input feature space. New information may be incorporated in the class models, and the classification rate increases with little variations over the 10 blocks of data.

V. Conclusion

In practical video-based face recognition applications, acquisition of new training data, after a classification system has been fielded, is common. As new training data becomes available, the recognition system should adapt to this information. In this paper, adaptation of the fuzzy ARTMAP neural network classifier during supervised incremental learning is studied using real-world video data. Performances of fuzzy ARTMAP are compared for different settings of hyperparameters and two class update scenarios. Results indicate that adaptation of hyperparameters allows a neural network classifier to learn new data efficiently as they become available. In both scenarios,

TABLE III

EXAMPLE OF THE AVERAGE CLASSIFICATION RATE ACHIEVED AFTER LEARNING BLOCK D_b FOR A SEQUENCE OF CLASS REFINEMENT (THE SAME SEQUENCE SHOWN IN TABLE II). THE CLASSIFICATION RATE SPECIFIC TO THE CLASS LEARNED AT D_b AND FOR THE REMAINING OF THE CLASSES ARE SHOWN IN PERCENTAGE WITH THEIR 90% CONFIDENCE INTERVAL.

Learning block	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}
Class learned	5	2	11	1	8	6	4	9	10	3	7
Classification rate for the learned class	91 ±2	73 ±7	97 ±0	95 ±1	86 ±2	98 ±0	86 ±2	95 ±1	83 ±3	99 ±0	90 ±2
Classification rate the rest of the classes	64 ±2	65 ±3	59 ±3	56 ±4	47 ±5	57 ±2	60 ±3	52 ±3	54 ±3	58 ±4	58 ±4

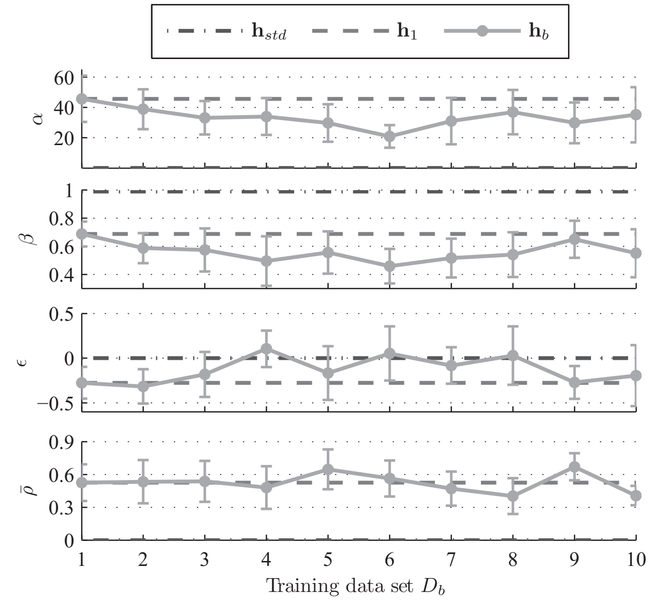


Fig. 4. Hyperparameters evolution during a balanced supervised incremental learning of all classes from ten learning blocks D_b. The mean of each hyperparameter is shown with its 90% confidence interval.

best classification rates where attained using hyperparameters optimized with data from all classes. It is the case with $\mathbf{h}_1$ for class refinement, and for $\mathbf{h}_b$ when refining all classes equally.

Moreover, results also show that an incremental learning algorithm, in addition of existing properties, should include: (1) adaptation of hyperparameters, and (2) access to previous learning data by the classifier to validate the model during training and properly optimize the hyperparameters. It is suggested that a long term memory, in the form of an external database composed of an equal number of samples from each class, should be used for the optimization process. Future works would then be to devise means for creating and updating such an external data base with regards of video-based face recognition challenges, and to apply it to both class update and enrollment learning scenario.

REFERENCES

[1] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, pp. 399–458, 2003.

[2] E. Granger, M. A. Rubin, S. Grossberg, and P. Lavoie, "A what-and-where fusion neural network for recognition and tracking of tultiple radar emitters," *Neural Networks*, vol. 14, pp. 325–344, 2001.

[3] R. Polikar, L. Udpa, S. S. Udpa, and V. Honavar, "Learn++ : An incremental learning algorithm for supervised neural networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 31, no. 4, pp. 497–508, 2001.

[4] G. A. Carpenter, S. Grossberg, and J. H. Reynolds, "Artmap: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, pp. 565–588, 1991.

[5] B. Fritzke, "Growing self-organizing networks - why?" in *Proc. European Symposium on Artificial Intelligence*, Brugge, Belgium, Apr. 1996, pp. 61–72.

[6] D. Chakraborty and N. R. Pal, "A novel training scheme for mlps to realize proper generalization and incremental learning," *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 1–4, 2003.

[7] K. Okamoto, S. Ozawa, and S. Abe, "A fast incremental learning algorithm with long-term memory," in *IEEE Int'l Joint Conf. on Neural Networks*, Portland, USA, Jul. 2003, pp. 102–107.

[8] S. Ruping, "Incremental learning with support vector machines," in *IEEE Int'l Conf. on Data Mining*, San Jose, USA, Nov. 2001, pp. 641–642.

[9] G. A. Carpenter, S. Grossberg, N. Markuzon, and J. H. R. D. B. Rosen, "Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 698–713, 1992.

[10] E. Granger, P. Henniges, L. S. Oliveira, and R. Sabourin, "Supervised learning of fuzzy artmap neural networks through particle swarm optimization," *Journal of Pattern Recognition Research*, vol. 2, no. 1, pp. 27–60, 2007.

[11] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *IEEE Int'l Joint Conf. on Neural Networks*, Perth, Australia, Nov. 1995, pp. 1942–1948.

[12] A. Canuto, G. Howells, and M. Fairhurst, "An investigation of the effects of variable vigilance within the repart neuro-fuzzy network," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 29, no. 4, pp. 317–334, 2000.

[13] A. Dubawski, "Stochastic validation for automated tuning of neural network's hyper-paramters," *Robotics and Autonomous Systems*, vol. 1997, pp. 83–93, 21.

[14] W. keung Fung and Y. hui Liu, "Adaptive categorization of art networks in robot behavior learning using game-theoretic formulation," *Neural Networks*, vol. 16, no. 10, pp. 1403–1420, 2003.

[15] M. Barry and E. Granger, "Comparison of artmap neural networks for classification for face recognition from video," in *IEEE Int'l Joint Conf. on Neural Networks*, Orlando, USA, 2007 Aug., pp. 2256–2261.

[16] E. Granger, J.-F. Connolly, and R. Sabourin, "A comparison of fuzzy artmap and gaussian artmap neural networks for incremental learning," in *IEEE Int'l Joint Conf. on Neural Networks*, Hong Kong, China, Jun. 2008, pp. 3304–3311.

[17] J.-F. Connolly, E. Granger, and R. Sabourin, "Supervised incremental learning with the fuzzy artmap neural network," in *Artificial Neural Networks in Pattern Recognition*, Paris, France, Jul. 2008, pp. 66–77.

[18] J. Kennedy, "Some issues and practices for particle swarms," in *IEEE Swarm Intelligence*, Honolulu, USA, Apr. 2007, pp. 162–169.

[19] D. O. Gorodnichy, "Video-based framework for face recognition in video," in *Second Workshop on Face Processing in Video in Proc. on Conf. on Computer and Robot Vision*, Victoria, Canada, May 2005, pp. 325–344.

[20] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Proc. Conf. Computer Vision and Pattern Recognition*, Kauai, USA, Dec. 2001, pp. 511–518.