Putting the Utility of Match Tracking in Fuzzy ARTMAP Training to the Test

Georgios C. Anagnostopoulos[1] and Michael Georgiopoulos[2]

[1] ECE Department, Florida Institute of Technology
150 West University Boulevard, Melbourne, Florida 32901, USA
anagnostop@email.com
http://www.fit.edu/~georgio/
[2] School of EE & CS, University of Central Florida
4000 Central Florida Boulevard, Florida 32816, USA
michaelg@mail.ucf.edu

Abstract. An integral component of Fuzzy ARTMAP's training phase is the use of Match Tracking (MT), whose functionality is to search for an appropriate category that will correctly classify a presented training pattern in case this particular pattern was originally misclassified. In this paper we explain the MT's role in detail, why it actually works and finally we put its usefulness to the test by comparing it to the simpler, faster alternative of not using MT at all during training. Finally, we present a series of experimental results that eventually raise questions about the MT's utility. More specifically, we show that in the absence of MT the resulting, trained FAM networks are of reasonable size and exhibit better generalization performance.

1 Introduction

Fuzzy ARTMAP (FAM) [1] is a neural network architecture based on the principle of *adaptive resonance theory* developed in [2]. The network is capable of learning input-output domain associations in an on-line or an off-line fashion. As a special case, when the output domain consists of a collection of class labels, FAM can be used as a classifier. In the sequel, when we refer to FAM, we will actually be referring to the FAM classifier. FAM enjoys several desirable properties of learning including the dual support for off-line (batch) and on-line (incremental) learning as well as the property of *learning stability*: using *fast learning* its training phase completes in a finite number of *list presentations* (epochs). FAM follows an exemplar-based learning paradigm and crystallizes its acquired knowledge in the form of *categories*, whose geometric representations are hyper-boxes embedded into the input domain. Learning in the presence of new data evidence occurs when either existing categories are updated or new categories are created.

An integral part of FAM's training phase is the *Match Tracking* (MT) mechanism. When an already-existing, chosen category initially misclassifies a training pattern,

V. Palade, R.J. Howlett, and L.C. Jain (Eds.): KES 2003, LNAI 2774, pp. 1-6, 2003.

MT initiates a new search of the category pool with the hope of eventually finding an appropriate category that will correctly classify the presented pattern. In other words, MT attempts to avoid the unnecessary creation of new categories during the learning process and therefore controls the structural complexity of the resulting FAM network by inducing an extra computational cost.

In this paper we will first elucidate MT's role and explain how and why it actually works (see Section 3). In order to test the utility of MT we compare it to the alternative of immediately creating a new category, when the training pattern has been misclassified in the early stages of learning. Although this last strategy does not directly control the structural complexity of the resulting classifier as MT does, it is computationally simpler. We need to note here that the idea of removing and/or replacing MT with other approaches is not new (for example, see [3], [4] and [5] to a name a few). In Section 4 we present experimental results on both simulated and real data that raise questions about MT's usefulness in FAM training. Our results in all cases indicate that not using MT and immediately creating categories in the previously described scenario can be less computationally intensive, produces comparable-in-size architectures and, to our surprise, may improve generalization performance as well. In the next section we provide some background on how FAM's fast learning is being accomplished, when a training pattern is being presented.

2 Learning in Fuzzy ARTMAP

Let N be the set containing the indices of all already-formed categories, $\bar{\rho} \in [0,1]$ be the *baseline vigilance parameter* value, a>0 be the *choice parameter* value, $\mathbf{w}_{\mathrm{j}} \in [0,1]^{2M}$ be the *template* vector of category j, $L(j)$ be the class label associated with category j, $\mathbf{x} \in [0,1]^{M}$ be the presented training pattern, $\mathbf{x}^{c} \in [0,1]^{2M}$ be the training pattern in complement-coded form, $L(\mathbf{x})$ be the class label of pattern $\mathbf{x}$, $\mathbf{1} \in [0,1]^{M}$ be the all-ones row vector, $\wedge$ be the *fuzzy-min operator*, $\rho(\mathbf{w}_j|\mathbf{x})$ and $T(\mathbf{w}_j|\mathbf{x})$ be the *category match function* (CMF) value and the *category choice function* (CCF) value respectively of category j with respect to $\mathbf{x}$, and T_u be the CCF value corresponding to uncommitted F_2-layer nodes. The pseudo-code depicted on the next page shows how FAM learns a training pattern.

The reader may notice that the provided pseudo-code differs from the one described in [1], but he/she should be assured that it reflects FAM's correct operation. The pseudo-code was re-written in a form that is more suitable for a software implementation of the training procedure. For example, it uses expressions (1) and (2) instead of $\rho := \rho(\mathbf{w}_J|\mathbf{x})+\varepsilon$ and $S:=S-\{j \in S \mid \rho(\mathbf{w}_j|\mathbf{x})<\rho\}$ respectively to avoid the involvement of the arbitrary, small, positive value ε mentioned in [1]. Also, CMF values are calculated prior to CCF values, which contrasts the description in [1], but is computationally more efficient as is shown in [6]. The interested reader can refer to [1] and [7] for more details on the involved concepts and other details regarding FAM's training phase.

```
Set $S$:=$N$ and $\rho := \overline{\rho}$
If $S=\emptyset$, set $J$:=$none$; otherwise
   Compute CMF values $\rho(\mathbf{w}_j|\mathbf{x})$ $\forall j\in S$
   Perform Vigilance Test: $S:=S-\{j\in S|\rho(\mathbf{w}_j|\mathbf{x})<\rho\}$
If $S=\emptyset$, set $J$:=$none$; otherwise
   Compute CCF values $T(\mathbf{w}_j|\mathbf{x})$ $\forall j\in S$
   Select wining category $J := \inf \arg\max_{j\in S} T(\mathbf{w}_j \mid \mathbf{x})$
   Perform Commitment Test: If $T(\mathbf{w}_J|\mathbf{x})<T_u$, set $J$:=$none$
While $J\neq none$ do
   Perform Prediction Test:
   If $L(J)=L(\mathbf{x})$, set $\mathbf{w}_J:=\mathbf{x}^c\wedge\mathbf{w}_J$ and exit the while-loop.
   If $L(J)\neq L(\mathbf{x})$
      Perform Match Tracking: Set $\rho:=\rho(\mathbf{w}_J|\mathbf{x})$              (1)
      Perform Vigilance Test: $S:=S-\{j\in S|\rho(\mathbf{w}_j|\mathbf{x})\leq\rho\}$   (2)
      If $S=\emptyset$, set $J$:=$none$; otherwise
         Select wining category $J := \inf \arg\max_{j\in S} T(\mathbf{w}_j \mid \mathbf{x})$
         Perform Commitment Test:
         If $T(\mathbf{w}_J|\mathbf{x})<T_u$, set $J$:=$none$
If $J=none$
   Create a new category $K$ with $\mathbf{w}_K$:=[$\mathbf{x}$ $\mathbf{1}$-$\mathbf{x}$]
   Set $L(K)$:=$L(\mathbf{x})$ and $N:=N\cup\{K\}$
```

3 The Role of Match Tracking

FAM is designed to support both off-line and on-line (incremental) learning. In order to accommodate the latter learning mode, FAM's training phase has been designed to adhere to the following principle:

FAM's Incremental Learning Principal (ILP): *Assume that a training pattern* **x** *has been presented and has been learnt by a FAM network during its training phase. If we present again the same pattern* **x** *immediately after it has been learnt, the network will classify it correctly.*

Assume that a training pattern **x** is being presented and category J is being selected, where $L(J)\neq L(\mathbf{x})$. In other words, FAM initially misclassifies **x**. In this case there would be two possibilities: a) Try to search for an already existing category I with $L(I)=L(\mathbf{x})$ such that the ILP holds. Only if the search fails, then create a new category K with $L(K)=L(\mathbf{x})$. This approach, although being computationally more involved, avoids increasing the structural complexity of the classifier, whenever this is possible, and is being followed in FAM training via the use of MT. b) Immediately create a new category K with $L(K)=L(\mathbf{x})$. This approach, of course, although putting no effort to control the increase in structural complexity, is faster in a computational sense than the former one. Here we need to note that the creation of a new category during training adheres to the ILP, but we omit a more detailed explanation.

During MT the value of ρ is being increased and a new search for a winning category is initiated. The search involves filtering out those categories that do not pass the Vigilance Test and then selecting the category of highest CCF value (if there is more than one, then the one featuring the lowest category index) is being chosen. Assuming that the Commitment Test has been passed, a new application of the Prediction Test will determine if MT has to be applied once again or if an appropriate category has been found. Let J be the category that was initially chosen with $L(J)\neq L(\mathbf{x})$ and I be the category found after MT has been invoked with $L(I)=L(\mathbf{x})$. Due to MT it will hold that $\rho(\mathbf{w}_I|\mathbf{x}) > \rho(\mathbf{w}_J|\mathbf{x})$. In order to preserve the ILP we must have that $T(\mathbf{x}^c\wedge\mathbf{w}_I|\mathbf{x}) > T(\mathbf{w}_J|\mathbf{x})$. It can be shown that MT achieves exactly this goal:

Proposition: *For any $a>0$ and any input pattern $\mathbf{x}$, if it holds that $\rho(\mathbf{w}_I|\mathbf{x})>\rho(\mathbf{w}_J|\mathbf{x})$, where I and J are any two categories of a FAM network, then it also holds that $T(\mathbf{x}^c\wedge\mathbf{w}_I|\mathbf{x})>T(\mathbf{w}_J|\mathbf{x})$.*

Proof: It can be shown that, if $\rho(\mathbf{w}_I|\mathbf{x})>\rho(\mathbf{w}_J|\mathbf{x})$ and $a>0$, then $T(\mathbf{x}^c\wedge\mathbf{w}_I|\mathbf{x})>T(\mathbf{x}^c\wedge\mathbf{w}_J|\mathbf{x})$. The result follows immediately from the fact that $T(\mathbf{x}^c\wedge\mathbf{w}_J|\mathbf{x})\geq T(\mathbf{w}_J|\mathbf{x})$, which holds for any $\mathbf{x}$ and any category J.

The above proof implicitly assumes the usage of the *Weber Law* CCF, although the proposition holds also for the *Choice-By-Difference* CCF (see [8]). Having explained what MT's role is and why it actually works, we return back to the two approaches (a) and (b) we've mentioned earlier to elaborate further. Approach (a) that subscribes to the use of MT tries to avoid creating new categories, whenever this is possible, by paying an extra computational cost. From the provided pseudo code it is easy to see that the computational complexity of FAM's training phase is $O(N^2)$ per presented pattern, where N stands for the number of existing categories when a pattern is presented. On the other hand, we have approach (b), where no MT is being practiced and a new category is immediately created. It is easy to show (but, again, the details are omitted) that this behavior is equivalent to performing MT by setting $\rho:=1$ instead of $\rho:=\rho(\mathbf{w}_J|\mathbf{x})$ in the pseudo-code. It can also be shown that approach (b) would result in a computational complexity of $O(N)$ per presented pattern.

In order to assess the effectiveness or the utility of MT, we need to first find out how does the presence or absence of MT affect the i) overall computational cost of FAM's training phase ii) the total number of categories created after training has completed (FAM's structural complexity) and iii) FAM's generalization performance. In the following section we make an attempt to provide an answer for these questions.

4 Experimental Results & Conclusions

Our experimentations dealt with 4 sets of data, the first 3 being artificially produced by sampling with equal prior probabilities from mixtures of 4 2-dimensional, isotropic and equidistant Gaussian distributions of Bayes error rates 10%, 25% and 40% respectively. The last data set we used was the Abalone database from the UCI Machine Learning Repository [9]. In order to assess the generalization performance of the two approaches (usage of MT versus not using it) as they apply to FAM learning, for each data set and for each of the two approaches we trained to completion FAM classifiers using off-line fast learning, 100 different presentation orders of training patterns, 40 values for ρ (0.0, 0.025, …, 0.975), 4 values of a (0.001, 0.01, 0.1, 1.0) and $w_u \rightarrow \infty$ (which implies that $T_u \rightarrow 2/(4+a)$). This resulted in a total of 100×40×4=1600 FAM architectures. For each data set and each approach (FAM with and without MT), out of the 16000 networks we selected via cross-validation the best 100 performing classifiers, which we then tested using a separate collection of test patterns. In specific, for the artificially generated data sets we used 500 training patterns, 5000 patterns for cross-validation and 5000 patterns for testing. On the other hand, for the Abalone database we used 1000 training patterns, 2133 patterns for cross-validation and 1044 patterns for testing. The collections of patterns for cross-validation and testing were chosen large so that the classification performance comparisons would yield statistically significant results.

Due to the lack of space, only selected results are depicted in Tables 1 and 2. Similar results to the ones illustrated in Table 1 were also found for the other two Gaussian mixture data sets. *PCC* stands for percent correct classification, *Categories* for the number of categories employed in the trained networks and *FLOPs* stands for the number of floating point operations that were performed during the training phase.

From the tables we observe that refraining from MT and immediately creating new categories to correct initial pattern misclassifications during training may be far superior to actually employing MT in terms of generalization performance. The best networks trained without the use of MT are approximately by 5% to 6% better on the collection of test patterns, while being of somewhat larger size (a difference of 60 to 70 categories). Taking into account that more than 1000 patterns were used in testing, these differences in performance are statistically significant at a significance level of 0.05. Furthermore, these "best" classifiers required less computational effort (measured in FLOPs) to be trained than their homologues (that were trained using MT) despite their larger size.

Table 1. Test results for the Gaussian Mixture (Bayes error 40%) data set

	Using MT		*Not using MT*	
	Best	Average	Best	Average
PCC	54.90%	49.37%	60.24%	51.47%
Categories	179	230	220	284
FLOPs	5.6×10^6	6.73×10^6	2.3×10^6	6.5×10^6

Table 2. Test results for the Abalone data set

	Using MT		*Not using MT*	
	Best	Average	Best	Average
PCC	49.56%	45.39%	56.87%	45.10%
Categories	367	230	541	569
FLOPs	63.6×10^6	61×10^6	40.2×10^6	96×10^6

On balance, we have demonstrated with our experimental results that the usefulness of MT may be questionable. We have presented indications that MT may hinder a FAM classifier to achieve higher correct classification rates, while requiring in some cases more computations during training to control the classifier's size.

Acknowledgements

The authors would like to thank the 3 anonymous referees that reviewed this paper and provided valuable comments and suggestions.

References

[1] 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 Transaction on Neural Networks. 3:5 (1992) 698-713

[2] Grossberg, S.: Adaptive Pattern Recognition and Universal Encoding II: Feedback, Expectation, Olfaction, and Illusions. Biological Cybernetics. 23 (1976) 187-202

[3] Mariot, S., Harrison, R.F.: A Modified Fuzzy ARTMAP Architecture for the Approximation of Noisy Mappings. Neural Networks. 8:4 (1995) 619-641

[4] Gomez Sanchez, E., Dimitriadis, Y.A., Cano Izquierdo, J.M., Lopez Coronado, J.: MicroARTMAP: Use of Mutual Information for Category Reduction in Fuzzy ARTMAP**.** Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. 6 (2000) 47-52

[5] Verzi, S.J., Heileman, G.L., Georgiopoulos, M., Healy, M.J.:Boosted ARTMAP. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. 1 (1998) 396-401

[6] Burwick, T., Joublin, F.: Optimal Algorithmic Complexity of Fuzzy ART. Neural Processing Letters. 7 (1998) 37-41

[7] Anagnostopoulos, G.C., Georgiopoulos, M.: Category Regions as New Geometric Concepts in Fuzzy ART and Fuzzy ARTMAP. Neural Networks. 15:10 (2002) 1205-1221

[8] Carpenter, G.A., Gjaja, M.N.: Fuzzy ART choice functions. Proceedings of the World Congress on Neural Networks. 5 (1994) 133-142

[9] Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Department of Information and Computer Science, University of California, Irvine, California. (1998)