A Lightweight Binary ART Model for Resource-Constrained Online Learning

Niklas M. Melton
Dept. of Computer Science
Missouri University of Science and Technology
Rolla, MO, USA
niklasmelton@mst.edu

Sasha Petrenko
Dept. of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, MO, USA
petrenkos@mst.edu

Jian Liu
Dept. of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, MO, USA
jliu@mst.edu

Jacob Schroll
Dept. of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, MO, USA
jsnph@mst.edu

Micah Renfrow
Dept. of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, MO, USA
marhnd@mst.edu

Iwan Sandjaja
Dept. of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, MO, USA
iskg3@mst.edu

Donald C. Wunsch II
Dept. of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, MO, USA
dwunsch@mst.edu

***Abstract*—Online learning models offer clear advantages over static alternatives when operating in dynamic settings. Unfortunately, this class of models is often too computationally demanding for the embedded and edge platforms where such adaptability is most valuable. This work introduces a new model architecture derived from the Adaptive Resonance Theory family that exploits binary data representations and exclusively hardware-efficient integer operations, eliminating the need for floating-point arithmetic or division. In contrast to existing ART architectures that rely on floating-point arithmetic, the proposed design enables fast, memory-efficient, and hardware-optimized online learning, making it suitable for deployment on resource-constrained devices. Through comparisons with analog counterparts, we demonstrate that the proposed model is capable of achieving faster training and inference while reducing memory usage by orders of magnitude, without any significant trade-offs in accuracy.**

***Index Terms*—online learning, continual learning, machine learning**

I. Introduction

Online learning methods are increasingly important in applications where data arrive sequentially and model updates must be performed in real time. Such settings arise in embedded sensing, adaptive control, human–machine interaction, and edge intelligence more broadly [1]. Despite these demands, many state-of-the-art online learning models rely on floating-point operations, dense parameterizations, or repeated gradient-based updates that impose substantial computational and memory costs. These requirements limit their deployment on resource-constrained platforms, where power, throughput, and memory budgets are tightly restricted [2].

Adaptive Resonance Theory (ART) provides a family of architectures designed to support stable incremental learning while mitigating catastrophic forgetting [3]–[6]. ART models have been successfully applied to clustering, classification, and multi-modal fusion, and their stability–plasticity framework offers clear benefits for nonstationary environments. These benefits are particularly relevant to online learning and edge computing, where model explainability is also valuable. Conventional ART variants, however, typically operate on real-valued feature vectors and rely on floating-point comparisons and updates [5]. Although effective, these operations can be slow on resource-constrained hardware and require memory footprints that exceed the capabilities of many embedded systems.

Mapping the continuous match and update operations of ART to binary representations provides a direct path toward reducing computational and memory demands. Bit-wise comparisons eliminate the need for floating-point arithmetic, and binary encodings allow category templates to be stored as compact bit arrays rather than real-valued vectors. These properties align closely with the constraints of embedded platforms, where integer and logical operations are significantly faster and memory is often the dominant limitation.

Although the earliest ART variant, ART-1, was formulated for binary inputs, its internal representations remain real-valued and do not fully exploit Boolean structure [3]. This highlights that merely operating on binary data is insufficient; the learning dynamics themselves must be reformulated to achieve meaningful hardware efficiency.

This paper develops a lightweight binary ART model, termed Binary Fuzzy ART (BFA), that addresses these constraints by reformulating core ART mechanisms around binary data representations and Boolean algebra. BFA implements a

variant of Fuzzy ART [7] in which real-valued feature vectors are replaced by binary encodings, and category matching and weight updates are computed using bit-wise operations. Despite these modifications, BFA preserves the essential dynamics of ART, including dynamic category creation and stable incremental/continual learning. At the same time, the binary formulation yields substantial improvements in speed and memory efficiency while completely eliminating the need for floating-point arithmetic or division. Furthermore, the model integrates seamlessly with compound ART architectures such as ARTMAP [8], [9], Fusion ART [10], and FALCON [11], enabling supervised learning, data fusion, and reinforcement learning within a unified binary framework.

A key enabler of this model is the use of binary encoding schemes that map continuous data to discrete representations. While various strategies exist, including standard base-2 binary and gray encoding, we focus primarily on thermometer encoding due to its monotonic structure and resulting interpretability [12]. We further compare thermometer encoding with alternative schemes to characterize their effects on classification accuracy, model compactness, and computational cost.

Through empirical evaluation on standard benchmarks, we show that BFA is capable of significantly accelerated training and inference and reductions of memory requirements by up to multiple orders of magnitude compared to their analog counterparts. These results demonstrate that BFA preserves the adaptability and stability of classical ART while making online learning feasible in settings that demand extreme efficiency.

To support reproducibility, the code and scripts used to run the experiments in this work are publicly available at https://github.com/NiklasMelton/BFA_experiments.

II. Background

A. Adaptive Resonance Theory

ART provides a model framework for stable incremental learning in nonstationary environments [3]–[6]. ART models maintain a dynamic set of categories that are updated online through a match–tracking mechanism. This mechanism ensures that new inputs either refine an existing category or trigger the creation of a new one when the similarity between the input and all existing categories falls below a vigilance threshold. The result is a system capable of both plasticity—rapid adaptation to new data—and stability—retention of previously learned knowledge. Within this framework, different ART variants have been developed to accommodate specific data types and learning objectives. ART-1, the original ART model [3], was designed for binary inputs but nevertheless maintains internal category representations as floating-point vectors, limiting its efficiency and representational compactness. Fuzzy ART [7] extends ART to continuous inputs by replacing exact matching with a fuzzy intersection operator and introducing complement coding to reduce category proliferation. Compound models such as ARTMAP [8], [9], Fusion ART [10], and FALCON [11] build on these foundations to support supervised learning, multi-modal data fusion, and reinforcement learning, respectively.

Although these architectures are effective, their reliance on real-valued computations, vector norms, and floating-point internal states presents substantial overhead when deployed on embedded or low-power systems. The memory requirements scale linearly with both the input dimensionality and the number of learned categories, and the comparisons required for category matching dominate runtime in streaming applications. Despite efforts having been made to mitigate these drawbacks when combining ART algorithms with deep fixed-architecture neural networks [13], these factors are frequently the limiting principle of their efficacy in real applications.

B. Fuzzy ART

Fuzzy ART is defined by Eqs. 1–4, which specify the category choice function, match criterion, vigilance test, and weight update rule, respectively [7]. In these equations, $\mathbf{I}$ denotes the normalized, complement-coded input sample, $\mathbf{W}_c^t$ is the weight vector associated with category c at time t, and $\wedge_f$ represents the fuzzy AND (element-wise minimum) operator. The notation $|x|_1$ denotes the ℓ_1 norm. The parameters $\alpha \in [0, \infty)$, $\beta \in [0, 1]$, and $\rho \in [0, 1]$ correspond to the choice, learning-rate, and vigilance parameters, respectively.

For a given input sample, the activation of each category c is first computed using Eq. 1. Categories are then evaluated in descending order of activation. For each candidate category, the match criterion between the input and the category weight is computed using Eq. 2. If this value satisfies the vigilance test in Eq. 3, the input and category are said to resonate. Upon resonance, the category weight is updated to incorporate the input according to Eq. 4. If no existing category satisfies the vigilance constraint, or if no categories yet exist, a new category is created and its weight is initialized directly from the input sample $\mathbf{I}$.

$$T(\mathbf{I}, \mathbf{W}_c^t) = \frac{|\mathbf{I} \wedge_f \mathbf{W}_c^t|_1}{\alpha + |\mathbf{W}_c^t|_1} \tag{1}$$

$$M(\mathbf{I}, \mathbf{W}_c^t) = \frac{|\mathbf{I} \wedge_f \mathbf{W}_c^t|_1}{|\mathbf{I}|_1} \tag{2}$$

$$M(\mathbf{I}, \mathbf{W}_c^t) \geq \rho \tag{3}$$

$$\mathbf{W}_c^{t+1} = \beta \left(\mathbf{I} \wedge_f \mathbf{W}_c^t\right) + (1 - \beta)\mathbf{W}_c^t \tag{4}$$

C. Binary Encodings and Discrete Learning

Discrete encodings replace continuous inputs with structured binary representations. These encodings enable comparisons to be computed using Boolean operations, which are considerably faster on most embedded processors and require significantly less memory. Unary or thermometer encoding produces a monotonic, easily interpretable representation in which the number of active bits corresponds to the magnitude of the original value. This scheme is widely used due to its robustness and predictable structure [12].

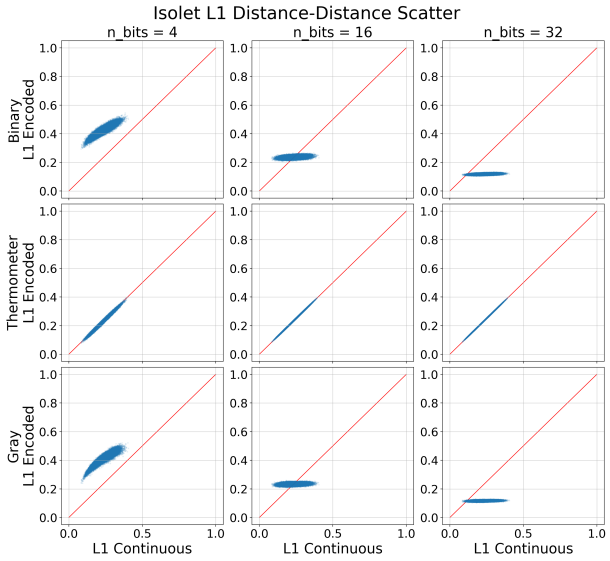

Fig. 1: Comparison of ℓ_1 distances between sample pairs in the continuous Isolet feature space and their corresponding binary-encoded representations using true binary, gray, and thermometer encodings. Distances are normalized by the number of features in the space, and the dataset is normalized prior to encoding. Thermometer encoding exhibits a strong linear relationship between continuous and encoded distances, whereas true binary and gray encodings show pronounced nonlinear behavior.

Alternative encodings–such as standard base-2 binary or gray encoding–provide more compact representations but introduce non-monotonic transitions that can affect similarity computations and learning dynamics. Gray encoding reduces bit transitions but does not necessarily preserve the ℓ_1 distance structure used by ART.

III. Encoding Analysis

We explicitly compare three common schemes for encoding continuous-valued features as binary vectors in the context of ℓ_1-based models such as Fuzzy ART: standard base-2 binary, gray code, and thermometer code. Using the normalized Isolet dataset as a benchmark [14], we evaluate each encoding at bit depths of 4, 16, and 32. We focus on thermometer encoding due to its monotonicity and interpretability.

For evaluation, we randomly sample 60,000 pairs of encoded data points for each encoding type and bit depth. We compute the ℓ_1 distance between each pair in the encoded space and compare it to the ℓ_1 distance between the corresponding samples in the original continuous space. All distances are normalized by the number of features in the corresponding space. The resulting distances are visualized as scatter plots, with continuous-space distances on the x-axis and encoded-space distances on the y-axis. A $45°$ reference line is included in each subplot to indicate ideal distance preservation.

Figure 1 demonstrates that thermometer encoding yields an approximately linear relationship between distances in the continuous and encoded spaces across all evaluated bit depths. In contrast, both true binary and gray encodings exhibit substantial non-linearity, indicating poor preservation of ℓ_1 distances. This behavior is consistent with the strictly monotonic structure of thermometer codes, which ensures proportional changes in encoded distance with respect to changes in the underlying continuous values. Based on these results, the remainder of this work will leverage thermometer encoding for binary conversion.

IV. Binary Fuzzy ART

A. Definition

Binary Fuzzy ART adapts Eqs. 1–4 to operate on discrete-valued data. In this variant, both the input vector $\mathbf{I}$ and the category weight vector $\mathbf{W}_c^t$ are binary. The operator $H(\cdot)$ denotes the Hamming weight of a binary vector, i.e., the population count, while the operator $\wedge_b$ denotes the element-wise logical AND operator applied to binary vectors. The vigilance parameter $\bar{\rho} \in [0, \mathcal{D}]$ is an integer threshold, where $\mathcal{D}$ (equivalent to $|\mathbf{I}|_1$) is the dimensionality of the input prior to complement coding.

The category choice function is given by Eq. 6, where the denominator is simply the Hamming weight of the category weight vector. The match criterion in Eq. 5 measures the overlap between the input and the category weight using a simple bit-wise AND followed by a population count. A category is said to resonate with the input if the match criterion satisfies the vigilance constraint in Eq. 7, in which case the category weight is updated according to the binary update rule in Eq. 8.

$$M(\mathbf{I}, \mathbf{W}_c^t) = H\left(\mathbf{I} \wedge_b \mathbf{W}_c^t\right) \tag{5}$$

$$T(\mathbf{I}, \mathbf{W}_c^t) = \begin{cases} \frac{M(\mathbf{I}, \mathbf{W}_c^t)}{H(\mathbf{W}_c^t)}, & H(\mathbf{W}_c^t) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$M(\mathbf{I}, \mathbf{W}_c^t) \geq \bar{\rho} \tag{7}$$

$$\mathbf{W}_c^{t+1} = \mathbf{I} \wedge_b \mathbf{W}_c^t \tag{8}$$

It is important to note that Eq. 6 lacks the α parameter from the original Fuzzy ART implementation. However, both $M(\mathbf{I}, \mathbf{W}_c^t)$ and $H(\mathbf{W}_c^t)$ are non-negative integer values and, due to their nature, we can state that $0 \leq M(\mathbf{I}, \mathbf{W}_c^t) \leq H(\mathbf{W}_c^t)$. Therefore, if $H(\mathbf{W}_c^t) = 0$, $M(\mathbf{I}, \mathbf{W}_c^t)$ must also equal 0. This necessary condition allows us to mimic the stabilizing effect of the α parameter with the normalizing conditional statement in Eq. 6. Additionally, it is important to note that Eq. 8 lacks the β parameter from the original Fuzzy ART implementation. Removing β is required to maintain binary weight vectors, but the functional result of this is that BFA exclusively learns under the Fuzzy ART equivalent of Fast-Learning.

In BFA, floating-point operations are confined to Eq. 6, where a single division is required to control category proliferation. In practice, this division can be fully eliminated using cross-multiplication as $T(\mathbf{I}, \mathbf{W}_c^t)$ is only needed for comparison operations.

To allow for cross multiplication, we must redefine the normalization seen in Eq. 6 using Eq. 9.

$$\widehat{H_c^W} = \begin{cases} H(\mathbf{W}_c^t), & \text{if } H(\mathbf{W}_c^t) > 0, \\ 1, & \text{otherwise} \end{cases} \tag{9}$$

$$\begin{aligned} T(\mathbf{I}, \mathbf{W}_{c_1}^t) > T(\mathbf{I}, \mathbf{W}_{c_2}^t) \Leftrightarrow \\ M(\mathbf{I}, \mathbf{W}_{c_1}^t) \cdot \widehat{H_{c_2}^W} > M(\mathbf{I}, \mathbf{W}_{c_2}^t) \cdot \widehat{H_{c_1}^W} \end{aligned} \tag{10}$$

This comparison definition naturally lends itself to efficient sorting, which can be leveraged for the search operation which iterates over each category in descending order of activation until either one category satisfies Eq. 7 or no categories remain. Practically, this should be implemented as a two-phase sort where the first phase orders categories in descending order of $H(\mathbf{W}_c^t)$ and the second phase applies a stable sort using the cross-multiplication comparator. The first phase mimics the effect of an infinitesimal alpha parameter by biasing results toward smaller categories.

As a result, **no floating point operations nor divisions are required for BFA**. The only required arithmetic operations are the bit-wise AND, population count, integer multiplication, and logical comparison.

B. Memory Optimization of Binary Fuzzy ART

In BFA we leverage thermometer encoded representations of sample data, which then results in thermometer encoded weight vectors. While this encoding greatly benefits BFA (as we will show in the following sections), it comes at great cost to memory, particularly when combined with complement coding. However, the thermometer code representation of samples and the resulting weights is only required at compute time. When samples or weights are not in immediate use, the thermometer encoded representations can easily be stored in base-2 binary format. For weights, however, we prescribe the following steps:

BFA weights can be decomposed into multiple feature-level components:

$$\mathbf{W}_c = \Big\Vert_{f \in \mathcal{F}} \mathbf{W}_{c,f} \tag{11}$$

Due to complement coding, these feature-level components can further be decomposed into upper- and lower-bound components of the feature:

$$\mathbf{W}_{c,f} = \left[\mathbf{W}_{c,f}^{lower} \,\middle\Vert\, 1 - \mathbf{W}_{c,f}^{upper} \right] \tag{12}$$

These upper- and lower-bound components, each representing an integer value in thermometer code, can then be easily converted to base-2 binary representation using the method outlined in [12].

Algorithm 1 Binary Fuzzy ART (BFA)

Require: Continuous dataset $X = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$
Require: Vigilance parameter $\bar{\rho} \in [0, \mathcal{D}]$, thermometer resolution n_{bits}
Ensure: Set of category weight vectors $\{\mathbf{W}_c\}$

$X \leftarrow \text{NormalizeToUnitInterval}(X)$
$X \leftarrow \text{ThermometerEncode}(X, n_{\text{bits}})$ ▷ $\mathbf{I} \in \{0,1\}^{\mathcal{D}}$
$\mathcal{W} \leftarrow \emptyset$ ▷ set of category weight vectors
$\widehat{H^W} \leftarrow \emptyset$ ▷ normalized Hamming weights
for each input vector $\mathbf{I} \in X$ **do**
  if $\mathcal{W} = \emptyset$ **then**
    create new category c
    $\mathbf{W}_c \leftarrow \mathbf{I}$
    $\widehat{H_c^W} \leftarrow \max(1, H(\mathbf{W}_c))$
    add c to $\mathcal{W}$
    continue
  end if
  for each category $c \in \mathcal{W}$ **do**
    $M_c \leftarrow H(\mathbf{I} \wedge_b \mathbf{W}_c)$ ▷ Eq. 5
  end for
  $\mathcal{C} \leftarrow \text{SortCategories}(\mathcal{W})$ ▷ descending using Eq. 10
  for each category $c \in \mathcal{C}$ **do**
    if $M_c \geq \bar{\rho}$ **then** ▷ Eq. 7
      $\mathbf{W}_c \leftarrow \mathbf{I} \wedge_b \mathbf{W}_c$ ▷ Eq. 8
      $\widehat{H_c^W} \leftarrow \max(1, H(\mathbf{W}_c))$
      break
    end if
  end for
end for
return $\mathcal{W}$

By following these steps, we can reduce the memory requirements of both samples and weights in complement coded, n_T-bit thermometer encoding with $n_{\mathcal{F}}$ features from $2n_T n_{\mathcal{F}}$ to $2\lceil log_2(n_T + 1) \rceil n_{\mathcal{F}}$. Thus resulting in a memory efficient representation which can be leveraged for storage and transfer.

We intentionally leave the frequency and timing of these format conversions as a hardware-dependent design choice for engineers.

C. Pseudo Code

Algorithm 1 presents a pseudo-code implementation of Binary Fuzzy ART (BFA). The algorithm directly realizes the category choice, vigilance test, and weight update rules defined in Eqs. 6–8, while avoiding floating-point operations and divisions, making it well suited for hardware implementation.

V. Experimental Methodology

Our experiments center around comparing the performance of BFA to both the ancestral Fuzzy ART as well as to ART1. As BFA inherits all the core properties of Fuzzy ART, we primarily focus our experiments on evaluating the comparative time and space complexity of the models. We also address the reduced acuity of the data that results from the binary

encoding from the continuous space by evaluating Fuzzy ART on both the binary encoded and continuous forms of the data. For datasets, we selected Human Activity Recognition Using Smartphones (HAR) [15], Spambase [16], Pen-Based Recognition of Handwritten Digits (Pendigits) [17], MNIST [18], and FashionMNIST [19]. These datasets offer a diverse array of real-world, continuous-domain, multi-class classification problems. In all experiments, a 50/50 train test split was utilized with 5 different random seeds for shuffling.

All experiments were conducted on an HPC cluster using 4-CPU Intel Skylake 64-bit nodes with 40GB of memory. All ART-based models utilized the C++/pybind11 accelerated implementations found in [20].

A. Clustering Experiments

As BFA is, like other elementary ART models, an unsupervised model, we first evaluate performance on clustering tasks. Using the datasets described above, we use BFA, Fuzzy ART, and ART1 to find naturally occurring clusters in the data. Fuzzy ART models were parameterized with $alpha = 1e-10$ and $\beta = 1.0$ while ART1 was parameterized with $L = 1.0$. The datasets were encoded in both 1-bit and 16-bit thermometer code. Both BFA and ART1 exclusively use the binary encoded data; however, Fuzzy ART was tested on both the binary encoded data as well as the original continuous form. Two representative ρ values, 0.5 and 0.75, were tested for the ART-based models. The vigilance parameter ρ regulates the size of clusters by an inverse relationship, and thus increasing ρ tends to result in additional, small clusters being formed. Each valid combination of model, dataset, data-type, bit-depth, and ρ-value was tested 5 times with different random seeds. During these tests, the adjusted mutual information score (AMI) and memory in bits were recorded; this memory is calculated as the sum of the lengths of the category weight matrices, factoring in complement coded inputs and encoding bit depth. Any buffers are excluded from this calculation. For BFA models, the compressed memory format described in Section IV-B was used.

B. Supervised Experiments

As Binary Fuzzy ART is naturally compatible with the ARTMAP family of supervised learning algorithms, we establish a series of experiments to compare performance with related and common supervised classification models. Using Simplified ARTMAP [21] as a backbone, we create Binary Fuzzy ARTMAP, ART1MAP, and Fuzzy ARTMAP models for supervised classification. Additionally, we selected Multinomial Naive Bayes (NB) and Stochastic Gradient Descent classifiers (SGD) as baselines using the implementations provided in [22]. Fuzzy ARTMAP models were parameterized with $alpha = 1e-10$ and $\beta = 1.0$, ART1MAP was parameterized with $L = 1.0$, NB was parameterized with $\alpha = 1.0$, and SGD models were parameterized with log-loss and $\alpha = 1e-4$. All datasets are naturally in the continuous domain and thus must be encoded using the thermometer encoding process for the binary-exclusive models (ART1MAP and Binary Fuzzy ARTMAP). The datasets were encoded in both 1-bit and 16-bit thermometer code, and, to provide additional baseline comparisons, Fuzzy ARTMAP, NB, and SGD were tested on both the continuous and binary encoded versions of these datasets. Two representative ρ values, 0.0 and 0.9, were tested for the ART-based models. $\rho = 0$ allows the classification side to drive memory/cluster creation, while $\rho = 0.9$ generates more natural clusters. Unless noted, each configuration is evaluated over five random seeds. During these tests, accuracy, memory in bits, fit time, and predict time were recorded. We finally calculated the mean of each of these values across all random seeds. Timing excludes thermometer encoding and was limited to only the fit/predict times for each model. Memory calculations include all learned parameters of each model. For the ARTMAP variants, this includes the thermometer+complement coded weights as well as the class-to-cluster map. Any buffers are excluded from this calculation. For BFA models, the compressed memory format described in Section IV-B was used.

C. Continual Learning Experiments

Although BFA inherits the continual learning properties of Fuzzy ART, we empirically verify this behavior using a controlled class-incremental experiment. We train a BFA model –parameterized with $\rho = 0.8$ and $n_{\text{bits}} = 8$– incrementally on the MNIST dataset using a class-ordered training protocol. The dataset is split with an even 50/50 training and testing split, and training samples are presented in batches of 1,000 instances. Classes are introduced sequentially, with all samples from class 0 presented first, followed by class 1, and so on until the full training set has been processed. After each batch update, we evaluate per-class accuracy on the fixed test set, enabling us to track performance on newly introduced classes while monitoring retention on previously learned classes. Timing results reflect model fit/predict computation and exclude encoding overhead, while BFA memory results use the compressed representation described in Section IV-B.

VI. Results & Discussion

A. Clustering Results

The clustering experiments, plotted in Fig. 2 primarily reveal that BFA achieves nearly identical AMI results on all datasets to Fuzzy ART with binary encoded data. This behavior is expected as BFA is designed to perfectly mimic the behavior of Fuzzy ART with small α values. Minor differences do arise but we attribute these primarily to floating point errors. However, BFA achieves between 1 and 5 orders of magnitude in memory reduction (depending on the dataset and bit-depth) compared to Fuzzy ART with binary encodings. We do see that performance differs more significantly with Fuzzy ART on the continuous data but the AMI differences appear to resolve as n_{bits} increases. ART1 performs far worse in terms of both AMI and memory requirements in all experiments. Indeed, the ART1 MNIST and HAR experiments with $n_{bit} = 16$ exceeded 40Gb of memory and, due to machine limitations, those results were omitted from this study.

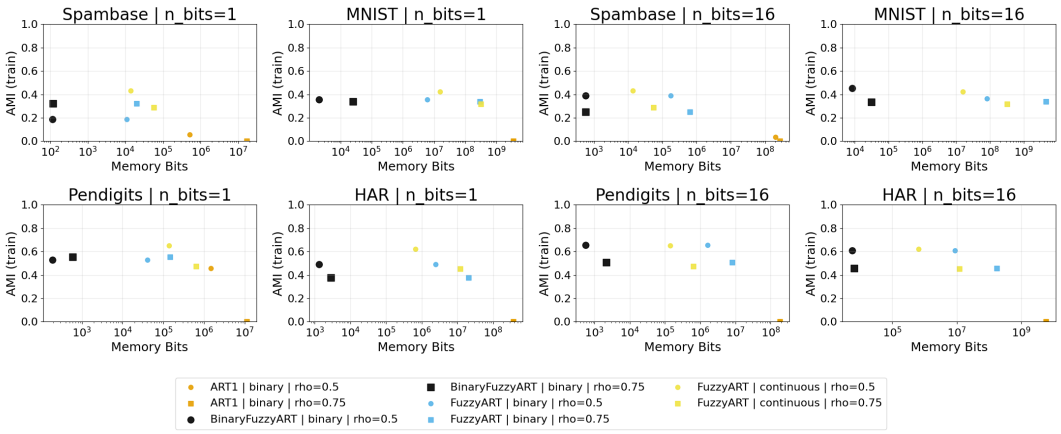

Fig. 2: Results of fitting Binary Fuzzy ART along with various baseline comparison models on four datasets for both 1-bit and 16-bit encoding levels, resulting in eight sub-plots. The performance of each model for the dataset and bit-depth pairing is plotted as a point, where the mean adjusted mutual information (AMI) score of the model is represented on the y-axis and the mean memory size (in bits) is represented on the x-axis. For the ρ-dependent ART models, we provide evaluation points for $\rho = 0.5$ and $\rho = 0.75$. For models capable of working with continuous data directly, we provide a reference evaluation point for the model trained on the original continuous data in addition to the evaluation point for the model trained on the binary encoded data.

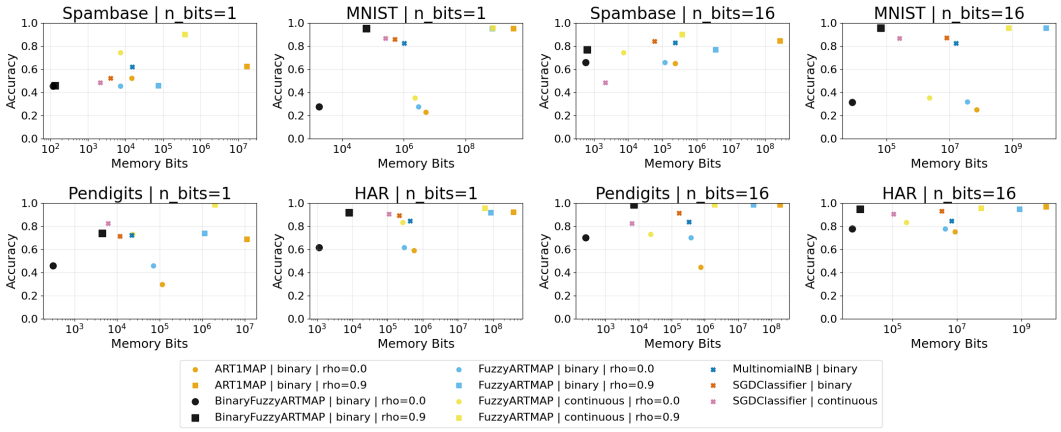

Fig. 3: Results of training Binary Fuzzy ARTMAP along with various baseline comparison models on four datasets for both 1-bit and 16-bit encoding levels, resulting in eight sub-plots. The performance of each model for the dataset and bit-depth pairing is plotted as a point, where the mean accuracy of the model is represented on the y-axis and the mean memory size (in bits) is represented on the x-axis. For the ρ-dependent ART models, we provide evaluation points for $\rho = 0$ and $\rho = 0.9$. For models capable of working with continuous data directly, we provide a reference evaluation point for the model trained on the original continuous data in addition to the evaluation point for the model trained on the binary encoded data.

B. Supervised Results

In Fig. 3, we see that supervised BFA models consistently match or exceed the prediction accuracy of competing models across most datasets while requiring substantially less memory. In contrast, all other ARTMAP variants demand more than two orders of magnitude greater memory, and in many cases considerably more. ART1MAP is especially memory intensive due to its reliance on both top–down and bottom–up weights. Among non-ART approaches, SGD classifiers are the most competitive in terms of accuracy and memory efficiency; however, they are poorly suited for continuous learning scenarios and generally lack interpretability.

The Spambase dataset provides an additional point of comparison, as nearly all models trained with $n_{bits} = 1$ perform significantly worse, in terms of accuracy, than their $n_{bits} = 16$ counterparts.

In Fig. 4, we see that with respect to computational time, BFA ARTMAP variants are consistently comparable to, or faster than, other ART-based models in both training and inference. The continuous Fuzzy ARTMAP variant surpasses BFA primarily in the $n_{bits} = 16$ setting. Although BFA does not outperform SGD or NB classifiers in raw timing

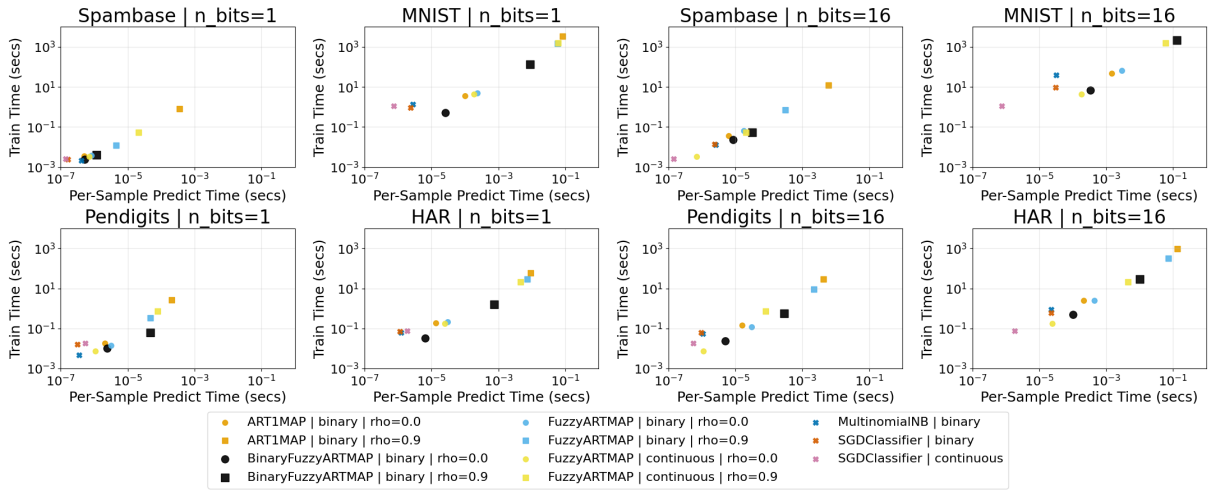

Fig. 4: Results of training Binary Fuzzy ARTMAP along with various baseline comparison models on four datasets for both 1-bit and 16-bit encoding levels, resulting in eight sub-plots. The performance of each model for the dataset and bit-depth pairing is plotted as a point, where the mean fit/training time (in seconds) of the model is represented on the y-axis and the mean per-sample prediction time (in seconds) is represented on the x-axis. For the ρ-dependent ART models, we provide evaluation points for $\rho = 0$ and $\rho = 0.9$. For models capable of working with continuous data directly, we provide a reference evaluation point for the model trained on the original continuous data in addition to the evaluation point for the model trained on the binary encoded data.

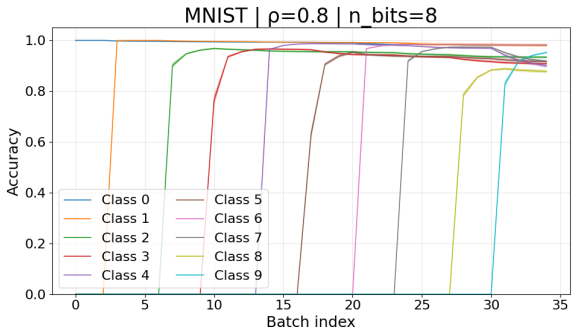

Fig. 5: Per-class accuracy of the MNIST dataset as a BFA model is trained on batches in a class-incremental fashion. The figure demonstrates that per-class accuracy for previously seen classes remains relatively steady throughout the entire training procedure.

performance, these alternatives do not support continuous learning and lack the level of explainability that distinguishes ART-based models.

C. Continual Learning Results

The continual learning experiment, visualized in Fig. 5, demonstrates clearly that the stability-plasticity characteristics of Fuzzy ART are preserved in BFA. We see that, as new classes are presented, the per-class accuracy of previously seen classes remains relatively stable, with minor accuracy degradation resulting from overlap of classes in the feature space [23].

VII. Conclusion

We introduced BFA as a fast, memory-efficient model for resource-constrained online learning. BFA preserves the core properties of Adaptive Resonance Theory, including continual learning, stability–plasticity balance, and model interpretability, while achieving orders-of-magnitude reductions in memory usage and, in many cases, faster training and inference. By reformulating category choice, matching, and updates around binary representations using bitwise AND and population count, and by eliminating floating-point operations and division through cross-multiplication comparisons, BFA is particularly well suited for embedded learning applications.

A key enabler of this approach is the encoding analysis showing that thermometer encoding better preserves ℓ_1-distance structure than standard binary or gray encodings, aligning naturally with ART-style similarity computations. Across clustering and supervised benchmarks, BFA closely matches the performance of its analog counterparts while dramatically reducing memory footprint, and the class-incremental MNIST study further demonstrates that the stability–plasticity behavior of Fuzzy ART is retained in BFA. We also showed that thermometer-coded templates can be stored compactly in base-2 form when not in immediate use, enabling more memory-efficient storage and transfer without changing the compute-time operations. Exploratory sweeps further suggested that performance generally improves with higher thermometer bit depth, although the best configuration remains dataset-dependent.

Future work will extend evaluation to truly streaming and concept-drift settings, such as prequential protocols, and study

robustness to noise and quantization. We also plan to benchmark end-to-end latency and energy on embedded platforms, including the overhead of encoding and format conversion, and to evaluate performance under explicit resource budgets, such as fixed memory limits, for a more deployment-faithful comparison.

ACKNOWLEDGMENT

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-22-2-0209. This research was supported by NSF grant 2420248 and by the Kummer Institute, Mary Finley Endowment, and Intelligent Systems Center of the Missouri University of Science and Technology.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

The computation for this work was performed on the high-performance computing infrastructure provided by Research Support Solutions at Missouri University of Science and Technology https://doi.org/10.71674/PH64-N397

REFERENCES

[1] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[2] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[3] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer vision, graphics, and image processing*, vol. 37, no. 1, pp. 54–115, 1987.

[4] S. Grossberg, "How does a brain build a cognitive code?" *Psychological review*, vol. 87, no. 1, p. 1, 1980.

[5] L. E. B. Da Silva, I. Elnabarawy, and D. C. Wunsch II, "A survey of adaptive resonance theory neural network models for engineering applications," *Neural Networks*, vol. 120, pp. 167–203, 2019.

[6] S. Grossberg, *Conscious mind, resonant brain: How each brain makes a mind*. Oxford University Press, 2021.

[7] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural networks*, vol. 4, no. 6, pp. 759–771, 1991.

[8] 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, no. 5, pp. 565–588, 1991.

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

[10] Y. Asfour, G. Carpenter, S. Grossberg, and G. Lesher, "Fusion artmap: an adaptive fuzzy network for multi-channel classification," in *Third International Conference on Industrial Fuzzy Control and Intelligent Systems*, 1993, pp. 155–160.

[11] A.-H. Tan, "Falcon: A fusion architecture for learning, cognition, and navigation," in *2004 IEEE international joint conference on neural networks (IEEE cat. no. 04CH37541)*, vol. 4. IEEE, 2004, pp. 3297–3302.

[12] B. Razavi, *Principles of Data Conversion System Design*. IEEE Press, 1994.

[13] S. Petrenko, L. E. Brito da Silva, and D. C. Wunsch, "Deepart: Deep gradient-free local learning with adaptive resonance," *Neural Networks*, vol. 190, p. 107580, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608025004605

[14] R. Cole and M. Fanty, "ISOLET," UCI Machine Learning Repository, 1991, DOI: https://doi.org/10.24432/C51G69.

[15] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, and X. Parra, "Human Activity Recognition Using Smartphones," UCI Machine Learning Repository, 2013, DOI: https://doi.org/10.24432/C54S4K.

[16] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, "Spambase," UCI Machine Learning Repository, 1999, DOI: https://doi.org/10.24432/C53G6X.

[17] E. Alpaydin and F. Alimoglu, "Pen-Based Recognition of Handwritten Digits," UCI Machine Learning Repository, 1996, DOI: https://doi.org/10.24432/C5MG6K.

[18] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[19] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1708.07747

[20] N. M. Melton, D. Tanksley, and D. C. Wunsch, "Adaptive resonance lib: A python package for adaptive resonance theory (art) models," *Journal of Open Source Software*, vol. 10, no. 114, p. 7764, 2025.

[21] T. Kasuba, "Simplified Fuzzy ARTMAP," *AI Expert*, vol. 8, pp. 18–25, 1993.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] N. M. Melton, S. Petrenko, L. E. Brito da Silva, and D. C. Wunsch II, "The Need for More Nuanced Metrics of Catastrophic Forgetting," in *Proceedings of the International Conference on Soft Computing & Machine Intelligence*, 2025.