Message Passing Adaptive Resonance Theory for Online Active Semi-Supervised Learning

Taehyeong Kim,[1,2] **Injune Hwang,** [2] **Hyundo Lee** [2]
Hyunseo Kim,[2] **Won-Seok Choi,** [2] **Byoung-Tak Zhang** [2]
[1] AI Lab, CTO Division, LG Electronics, Seoul, Republic of Korea
[2] Seoul National University, Seoul, Republic of Korea
{thkim, ijhwang, hdlee, hskim, wchoi, btzhang}@bi.snu.ac.kr

Abstract

Active learning is widely used to reduce labeling effort and training time by repeatedly querying only the most beneficial samples from the unlabeled data. In real-world problems where data cannot be stored indefinitely due to limited storage or privacy issues, the query selection and the model update should be performed as soon as a new data sample is observed. Various online active learning methods have been studied to deal with these challenges; however, there are difficulties in selecting *representative* query samples and updating the model efficiently. In this study, we propose *Message Passing Adaptive Resonance Theory* (MPART) for online active semi-supervised learning. The proposed model learns the distribution and topology of the input data online. It then infers the class of unlabeled data and selects informative and representative samples through message passing between nodes on the topological graph. MPART queries the beneficial samples on-the-fly in stream-based selective sampling scenarios, and continuously improve the classification model using both labeled and unlabeled data. We evaluate our model on visual (MNIST, SVHN, CIFAR-10) and audio (NSynth) datasets with comparable query selection strategies and frequencies, showing that MPART significantly outperforms the competitive models in online active learning environments.

1 Introduction

The recent success of deep learning in the field of visual object recognition and speech recognition is largely attributed to massive amount of labeled data. However, most of the data in the real-world is not labeled, so it takes a lot of time and effort to label and use it for deep learning. This burden impedes the widespread use of deep learning in real-world applications such as medical AI, home appliances and robotics, and separates the best solutions from becoming the best real-world solutions.

From this point of view, active learning is a promising field of machine learning. It reduces the labeling cost and effort by repeatedly selecting the most beneficial data among unlabeled set and requesting the oracle to label it. In most traditional active learning algorithms, one or multiple samples are selected from a large pool of data for querying (Settles 2011). Once the queried data is labeled, it is accumulated in the training dataset and the entire model is trained again. This process is repeated, typically assuming that all the labeled and unlabeled data can be stored and the learner can access the data at any time again. However, in aforementioned real-world problems, it is impossible to store unlimited training data due to limited storage or privacy issues. In addition, it is highly inefficient to repeatedly train the entire model with large data and review all the unlabeled data for querying whenever a new label is acquired.

In contrast, the active learning paradigm in an online manner does not assume that data samples can be accessed repeatedly but instead, the input data is given as a continuous stream (Lughofer 2017). This entails that uncertainty estimation of input samples and the decision of whether to query or not should be made online. The labeling of sample and its training results affect the uncertainty of subsequent input samples. Therefore, the model retraining or update should be done on-the-fly so that the new uncertainty distribution is estimated for the next query. Moreover, because the labeling cost can be expensive, only a limited number of query is permitted. This online active learning process, while more suitable for real-world problems, is more challenging than off-line active learning.

In this regard, we propose Message Passing Adaptive Resonance Theory (MPART) for online active semi-supervised learning. MPART learns the distribution and the topology of the input data online based on the ART network (Grossberg 1987) which keeps the existing knowledge stable when learning new data. Then it utilizes a message passing method inspired by GNN (Zhou et al. 2018) to estimate the class label and uncertainty of the input sample in a semi-supervised manner. MPART gradually improves its performance as the data accumulates by online active learning. The proposed model can efficiently learn new data from continuous stream by updating only a part of the model without reviewing the data already learned. Figure 1 shows the overview of the proposed model.

The main contributions can be summarized as follows.

- We propose a novel method that utilizes *co-activated* nodes in order to continuously construct a weighted topological graph based on the ART network.
- We propose a message passing method for the graph to estimate the class labels in a semi-supervised manner and to select informative and representative samples for querying.
- We propose an online active learning task where the frequency of query is limited and the total number of classes

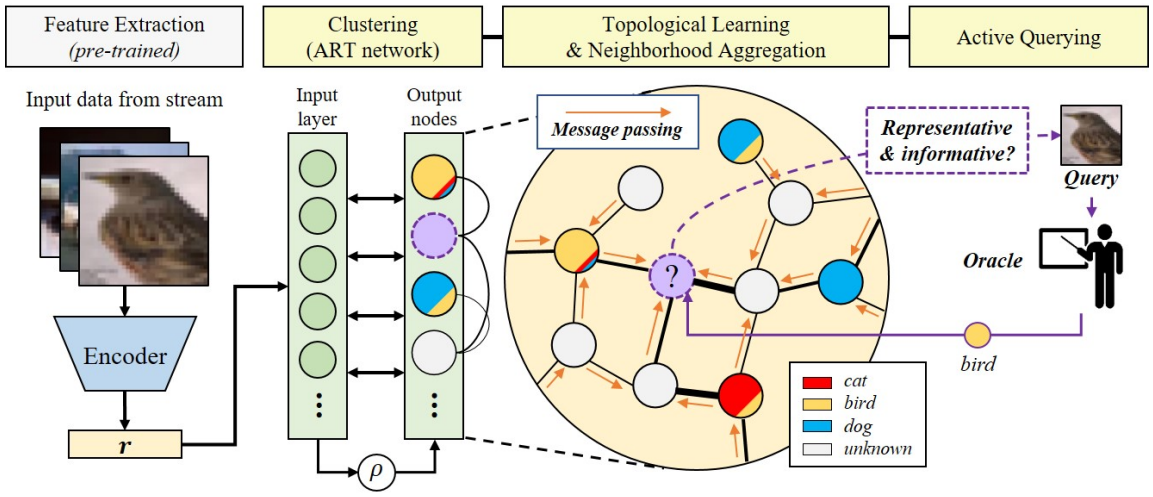

Figure 1: Overview of Message Passing Adaptive Resonance Theory (MPART). The feature is extracted from the input sample, and then MPART continuously constructs a weighted graph based on ART network by learning distributions and topology of the input data. It uses a message passing method to infer the class label and estimate uncertainty of the input sample for querying.

is unknown. We validate the proposed model with various query selection strategies and datasets. The experiment results show that our model significantly outperforms the competitive models in online active learning environments.

2 Related Work

Topology Learning Topology learning aims to learn the structural properties and topological relations underlying the input data. Additional knowledge such as class label or uncertainty can be obtained from the structural relationship using the topological information. SOM (Kohonen 1990), a type of artificial neural network, learns the topological properties of the input data using neighborhood function and competitive Hebbian learning (CHL). This method is useful for dimension reduction of high-dimensional input data, but has a disadvantage in that it requires a priori selection of a subspace. GNG (Fritzke 1995) and its derived models such as SOINN (Shen and Hasegawa 2006) and E-SOINN (Shen, Ogura, and Hasegawa 2007) are incremental network models which can learn the topological relations of input data using the CHL. There are also variants of ART networks which integrates new knowledge into its entire knowledge base so that what has already been learned is not forgotten by the new learning. Fuzzy ART-GL (Isawa et al. 2007) uses Group Learning that creates connections between similar categories. TopoART (Tscherepanow 2010) combines incremental clustering with topology learning, which enables stable online clustering of both stationary and non-stationary input data.

Semi-supervised Learning Semi-supervised learning uses a small amount of labeled data with a large amount of unlabeled data to improve the performance of the model. (Iscen et al. 2019) and (Douze et al. 2018) use the label propagation method to assign labels for unlabeled data using the nearby labeled data. GAM (Stretcu et al. 2019) uses an agreement model that calculates the probability of two nodes sharing the same label on the graph. EGNN (Kim et al. 2019) adapts a deep neural network to predict the edge-labels rather than the node-labels on the graph. However, most of these methods are not suitable for online learning, because they need predefined topological information or whole training data repeatedly. LPART (Kim et al. 2020) uses online label propagation on the ART network trained in semi-supervised manner to overcome this issue, but the conveyed label information is limited between the nodes due to the weak topology.

Online Active Learning Online active learning is used to reduce annotation costs. OASIS (Goldberg et al. 2011) is a Bayesian model using particle filtering to estimate the posterior. It is successful in binary classification, but the fixed number of particles makes it hard to adapt to increasing complexity of data. Other methods (Loy et al. 2012; Weigl et al. 2016) use ensemble models to compute the uncertainty of inputs and sample the model parameters to update the classifier. But the sampling in a complex task is difficult and inefficient, so these algorithms require the underlying model to be simple. To avoid this, we use a deterministic approach to calculate the uncertainty of data for active learning. The most closely related work to ours is (Shen et al. 2011), which addresses a very similar problem using A-SOINN. It shows the capability of online active semi-supervised learning, however, the generated query might be ambiguous since it is not the input itself but the prototype of inputs. Moreover, the model usually stores only a single label in a cluster of multiple nodes, so it cannot handle the possible mixture of classes within the cluster. Our model solves these problems by querying the input instantly and using the label density of each node which is updated via message passing.

Each related work shows a high potential in a variety of research fields to solve real-world problems, and motivates our idea of online active semi-supervised learning. In the

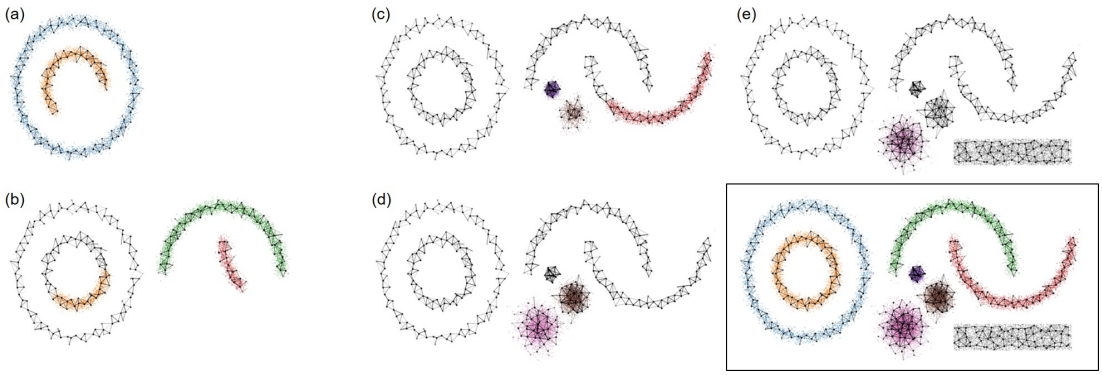

Figure 2: An example result of topology learning. MPART continuously learns the distribution and topology structure of sequential input data without catastrophic forgetting. (a) $\sim$ (e) is the process of topology learning, and the lower right figure is the final learning result. The colored scattered points refer to newly inputted data and the intensity of the topological graph represents the density of nodes and edges. All data samples was inputted only once, one by one.

following section, a novel topology learning method is presented, followed by the class inference method using message passing, and the active querying process.

3 Methods

3.1 Topology Learning

We first encode the input data using the pre-trained encoder of a modified VAE (Kingma and Welling 2013) (see Section 4.1 for details). As in Fuzzy ART (Carpenter, Grossberg, and Rosen 1991), the encoded representation r is complement coded to $I = [r, \vec{1} - r]$ in order to avoid proliferation of prototypes. With this, we can measure the similarity between the input and the category node j using the match function M_j and the choice function T_j.

$$M_j(I) = \frac{\|I \wedge w_j\|_1}{\|I\|_1}, \qquad T_j(I) = \frac{\|I \wedge w_j\|_1}{\alpha + \|w_j\|_1} \tag{1}$$

Here, $\wedge$ is the element-wise minimum operator, $\|\cdot\|_1$ is the L1 norm, and $\alpha > 0$ is a choice parameter. A node j becomes *activated* by I if $M_j(I)$ is greater than or equal to a vigilance parameter ρ. If none is activated, a new node i is created with I as an initial weight w_i and a winning count d_i of 1. When there are multiple activated nodes, one with the highest $T_j(I)$ is chosen as a *winner* denoted by J. We update w_J with a learning rate $0 \leq \beta \leq 1$, and increase d_J by 1.

$$\begin{aligned} w_J^{new} &= \beta(I \wedge w_J^{old}) + (1-\beta) w_J^{old} \\ d_J^{new} &= d_J^{old} + 1 \end{aligned} \tag{2}$$

In addition to this update, edges between the co-activated nodes are developed. Unlike GNG, SOINN and TopoART in which only the winner and the runner-up node are connected, we connect the winner to every co-activated node to better represent more complex topology of multiple nodes. For an edge connecting nodes i and j, its count c_{ij} is set as the number of times i and j have been co-activated. The edge weight e_{ij} is defined as a ratio of c_{ij} to the sum of winning counts of incident nodes i and j, i.e., $e_{ij} = c_{ij}/(d_i + d_j)$. While the edge count can grow prohibitively large, the edge weight remains low if any of the incident nodes was mostly co-activated with other nodes or activated solely. The value of e_{ij} approaches to 1, which is the upper bound, when the two nodes have always been co-activated whenever either one of them was the winner. Therefore, the edge weight e_{ij} is a good indication of the relationship between the two nodes.

Introducing the development of edges does not impede original ART properties. As a result, this process does not need to pre-define the structure and size of the network, and exploits the self-organizing property to learn the topology of data with irregular and complex distributions. It can also be operated online without catastrophic forgetting and has low computational costs. The example results of the topology learning are shown in Figure 2.

3.2 Message Passing

MPART performs message passing between nodes through a gradually formed topological graph. By using this method, Graph Neural Networks (GNN) could update the representation of the target node according to the neighboring nodes (Hamilton, Ying, and Leskovec 2017). This makes it possible to represent each node's information more accurately based on the overall graph. In this study, message passing for MPART is defined as Equation 3 for node classification and uncertainty estimation.

$$X_i^{(l+1)} = X_i^{(l)} + \delta \sum_{j \in \mathcal{N}_i} e_{ij} X_j^{(l)} \tag{3}$$

Here, X_i and X_j are information vectors of interest, such as class probability, on the target and neighboring nodes, and $\mathcal{N}_i$ is a set of all neighbors of node i. δ is a constant between 0 and 1 to determine the propagation rate. By performing this process on all node information $X_i^{(l)}$ of the graph, the updated node information $X_i^{(l+1)}$ for the next layer $(l+1)$ is obtained. The base value $X_i^{(0)}$ for layer 0 is the node's own value. The more repeatedly this method is performed on multiple layers, the broader aggregation is possible for information on the nodes that are further away from the

Algorithm 1: The MPART algorithm.

$V \leftarrow \{\}, C \leftarrow \{\}$ // initialize complete graph $G(V)$ and known class set C
for x_i **in** *input data stream* **do**
  $r_i \leftarrow \texttt{Encoding}(x_i)$ // see Section 4.1 for details
  $I_i \leftarrow [r_i, \overrightarrow{1} - r_i]$ // complement coding
  $A \leftarrow \{\}$
  for j **in** $1, \ldots, |V|$ **do**
    $M_j \leftarrow \|I_i \wedge w_j\|_1 \,/\, \|I_i\|_1$ // compute match function
    $T_j \leftarrow \|I_i \wedge w_j\|_1 \,/\, (\alpha + \|w_j\|_1)$ // compute choice function
    if $M_j \geq \rho$ **then** // find co-activated nodes
      $A \leftarrow A \cup \{j\}$
  if *A is empty* **then**
    $J \leftarrow |V| + 1, \quad V \leftarrow V \cup \{J\}$ // create new node
    $c_{Jv} \leftarrow 0,\ c_{vJ} \leftarrow 0 \quad \forall v \in V - \{J\}$ // initialize edge counts
    $q_J^{(0)}(y) \leftarrow 0 \quad \forall y \in C$ // initialize label density function
    $w_J \leftarrow I_i, \quad d_J \leftarrow 1$ // initialize weight and winning count
  else
    $J \leftarrow \arg\max_{j \in A}(T_j)$ // find winner
    $c_{Jv} \leftarrow c_{Jv} + 1,\ c_{vJ} \leftarrow c_{vJ} + 1 \quad \forall v \in A - \{J\}$
    $w_J \leftarrow \beta(I_i \wedge w_J) + (1 - \beta) w_J$ // update weight
    $d_J \leftarrow d_J + 1$ // update winning count
  $q_J^{(L)} \leftarrow \texttt{MessagePassing}(J, c, d, q^{(0)})$ // see Section 3.2 for details
  $p_J, \hat{y} \leftarrow \texttt{NodeClassification}(q_J^{(L)})$ // see Section 3.2 for details
  $u_J \leftarrow \texttt{UncertaintyEstimation}(p_J, q_J^{(L)})$ // see Section 3.3 for details
  if u_J *satisfies query condition* **then** // see Section 4.2 for details
    $y_i \leftarrow \texttt{QueryLabel}(x_i)$
    $q_J^{(0)}(y_i) \leftarrow q_J^{(0)}(y_i) + 1$ // initial $q_J^{(0)}(y_i)$ value is 0
    $C \leftarrow C \cup \{y_i\}$ // add new label to known class set C

reference node. Finally, we can use the node information $X_i^{(L)}$ of the final layer L to perform the task we want.

The calculation of all the information vectors $X^{(l+1)}$ at layer $(l+1)$ referring to layer l can be performed in parallel. In addition, it allows efficient computation as it only updates a partial area that needs to be newly calculated according to the number of layers and input sample.

Node Classification Node classification is a process of inferring a class of an input sample by a consensus among the winner and its neighboring nodes. The class of a node can be evaluated not only by the labels of the node which the sample belongs to, but also the labels of the surrounding nodes even in the case where a rare label is given. Therefore, we perform L-layer message passing for label density function $q_J^{(0)}(y)$ which is a vector containing the cumulative value of the node's label (see Section 3.3 for details). Then, the class probability distribution of input x_i is obtained using the aggregated label density function as shown in Equation 4.

$$q_i^{(l+1)}(y) = q_i^{(l)}(y) + \delta \sum_{j \in \mathcal{N}_i} e_{ij} q_j^{(l)}(y),\ \forall i \in \mathcal{N}_J(L-l)$$
$$p_J(y) = q_J^{(L)}(y) / \sum_{y' \in C} q_J^{(L)}(y') \tag{4}$$

J is a winning node with given input x_i and $\mathcal{N}_J(L-l)$ is $(L-l)$-hop neighborhood of node J. $p_J(y)$ is the probability that this node belongs to class y in last layer L. We infer the class $\hat{y}$ of x_i as the label with the highest class probability.

3.3 Active Querying

Uncertainty Estimation We use two different metrics to measure the uncertainty of the nodes as in (Kim et al. 2020). The first uncertainty, $u_{J,a}$, can be seen as an aleatoric uncertainty which is measured by the entropy of the classification probability as shown in Equation 5. This uncertainty tends to have a high value near the decision boundary, which can help in selecting samples near the boundary.

$$u_{J,a} = \begin{cases} \frac{-\sum_{y \in C} p_J(y) \log p_J(y)}{\log(|C|)}, & \text{if } |C| > 1 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The second uncertainty $u_{J,e}$ which can be seen as an epistemic uncertainty is based on the quantitative information of label density function $q_J^{(L)}$ of winning node as shown in Equation 6. Here, k is a positive constant for sensitivity. This uncertainty has a high value in the region lacking labels in

the input data distribution.

$$u_{J,e} = 1 - tanh(k \sum_{y \in C} q_J^{(L)}(y)) \quad (6)$$

Both uncertainties $u_{J,a}$ and $u_{J,e}$ are considered representative because they are derived from label density function $q_J^{(L)}$ aggregated from surrounding nodes. This is a desired property of criteria for selecting beneficial queries (Huang, Jin, and Zhou 2010), and the two uncertainties are complementary for query selection strategies. Therefore, we put these two uncertainty measures together with a weight τ between 0 and 1 to define a query selection score u_J as in Equation 7.

$$u_J = \tau u_{J,a} + (1 - \tau) u_{J,e} \quad (7)$$

If the label density of nodes is high enough, $u_{J,e}$ converges to 0, so u_J changes according to $u_{J,a}$ and more data near the decision boundary is selected as a query target.

Online Query Selection The message passing is performed around the winning node based on the topological graph, and estimates the label and uncertainty of an input sample. If the query selection score u_J of the input satisfies the condition according to the query selection strategy (see Section 4.2 for details), the model immediately queries the oracle to get a label y_i and increases the corresponding value of the winner's label density function $q_J^{(0)}(y_i)$ by 1. When a new node is created, $q_J^{(0)}(y)$ is initialized to zero for all $y \in C$ before updating $q_J^{(0)}(y_i)$. At this time, if the label of the new class is observed, the known class set C is also updated. The number of known classes C increases according to the label of the new class received from oracle, so if new class label y_{new} is added, $q_i^{(0)}(y_{new})$ for all nodes $i \in V$ is initialized to zero before updating $q_J^{(0)}(y_{new})$. The whole process of MPART algorithm is described in Algorithm 1, where we used a complete graph for simplicity of notations.

4 Experiments

We designed a new task for evaluating online active semi-supervised learning models. The basic goal is to correctly classify the given data, but the number of possible classes is not known in advance. The whole training dataset is initially unlabeled and the model can inquire the oracle for ground truth labels. Unlike typical active learning scenarios, the query frequency is limited in this task to imitate real-world scenarios where the labeling is highly costly or the oracle is often unavailable. Specifically, we let the model query only once in every T consecutive samples, where T is one of the three fixed periods: 100, 500 and 1000. All input samples are provided in an online manner, i.e., one after another, and the model has to discard a sample before it gets the next one.

We used four kinds of datasets, MNIST (LeCun et al. 1998), SVHN (Netzer et al. 2011), CIFAR-10 (Krizhevsky, Hinton et al. 2009), and NSynth (Engel et al. 2017). The NSynth audio data were converted to 64×64 spectrogram images. Additional pre-processing was applied including removal of silence-only data and limiting the pitch range (see Appendix for details). To train the model in each experiment, we sampled 10,000 data from the training split randomly per trial, except when using the NSynth dataset, where the whole validation split with 9,216 samples was used as a training set. This is to push the situation to an extreme; with 1/1000 query frequency restriction, the model will obtain only a single label per each of 10 classes by average. Therefore, the selection of representative samples becomes more crucial.

Our experiments have two main purposes. First, we validate the effect of message passing in the proposed model. For this, we also measured the performance of the model without message passing. The performance was evaluated on different combinations of task settings: three query selection frequencies $1/T$ ($T = 1000, 500, 100$) and three query selection strategies ('Random', 'Memory', 'Explorer') which are described in Section 4.2. Second, we compared the performance of the proposed model to that of the competitive models. We performed evaluation according to the query selection frequencies and different model settings ($L = 1, 3, 5$), while only the 'Explorer' strategy was used for query selection. Refer to the appendix for all other model parameters. In all experiments, query selection was restricted to be performed only once in every query period T. We repeated each experiment 30 times to obtain statistical significance.

4.1 Feature Extraction

In order for MPART to properly form clusters and generate a topological graph for high-dimensional input data, we used a pre-trained feature extractor for dimension reduction. Any typical feature extractor can be applied, and here we employed an encoder of a variant of VAE (Kingma and Welling 2013), which is adapted to weakly-supervised training as in (Kim et al. 2020). This feature extractor is trained with weak labels indicating whether or not the class of the current sample is the same as that of the previous one. A part of the encoded vector is called class-embedded representation, denoted by μ_d, that is considered to contain the class information. Since our interest is to classify the given data, we used this class-embedded representation as an input to MPART.

4.2 Query Selection Strategy

Random A random query is selected from a sequence of inputs. This is a baseline and works as an ablated version of active selection strategies. In this case, MPART and other models operate in an online semi-supervised learning manner.

Memory 'Memory' assumes that the learning agent has memory that can store only one sample. During each query period, one sample with the maximum query selection score u_J is stored and inquired whenever the period ends. By using this strategy, we can select the most beneficial data sample in the input stream.

Explorer 'Explorer' assumes the most stringent situation where the learning agent cannot store any input sample. In this situation, the learning agent must select one sample online for each query period T, and cannot change the sample

Query Selection		MNIST ($\rho = 0.97$)		SVHN ($\rho = 0.95$)		CIFAR-10 ($\rho = 0.95$)		NSynth ($\rho = 0.97$)	
Frequency	Strategy	w/o MP	w/ MP	w/o MP	w/ MP	w/o MP	w/ MP	w/o MP	w/ MP
1 / 1000	Random	22.6±5.2	64.9±11.5	15.8±4.3	56.7±6.3	10.5±0.2	32.0±4.2	22.3±0.9	66.5±9.6
	Memory	23.3±5.0	**96.1**±3.3	16.8±3.5	**70.2**±4.9	10.5±0.2	31.8±4.3	22.6±1.0	**79.9**±3.9
	Explorer	23.6±5.3	91.9±5.9	16.1±3.9	**70.2**±5.7	10.9±0.3	**34.3**±2.8	22.2±1.1	79.0±7.0
1 / 500	Random	30.3±3.2	87.0±9.0	23.0±4.2	71.5±6.5	10.9±0.2	**37.3**±2.6	24.9±1.3	76.7±6.2
	Memory	31.1±4.2	97.3±0.7	23.9±3.4	**77.0**±3.0	10.9±0.3	34.9±2.9	25.4±1.6	**85.7**±2.3
	Explorer	32.6±2.2	**97.4**±0.2	22.4±4.7	**77.0**±4.8	10.4±0.1	36.4±2.7	25.4±1.4	85.1±2.4
1 / 100	Random	55.8±3.2	97.4±0.2	41.6±3.3	**81.3**±0.5	14.2±0.6	**41.6**±1.3	41.6±2.2	87.8±1.2
	Memory	71.3±2.2	97.5±0.2	52.1±3.2	80.9±0.6	14.5±0.7	40.7±1.7	48.0±1.7	**88.9**±0.8
	Explorer	70.7±2.5	**97.6**±0.2	51.2±2.9	80.9±0.6	14.3±0.4	40.6±1.7	47.7±1.6	88.7±1.0

Table 1: The classification accuracy (mean ± std) of our model (MPART) depending on whether the message passing (MP) method is applied. The number of layer $L = 5$ was used. (unit : %)

once selected. Therefore, the chance of selecting an informative sample decreases as the exploration gets longer because of the fixed query selection period. In this strategy, the uncertainty distribution of the data explored so far is continuously updated using Equation 8 to solve the exploration-exploitation dilemma. The uncertainty distribution is assumed to follow the normal distribution $u_t \sim N(\mu_t, \sigma_t^2)$, where t is the number of samples examined within this period. The agent uses this distribution to determine whether to query the sample or not.

$$\begin{aligned} \mu_t &= (1 - t^{-1})\mu_{t-1} + t^{-1}u_J(x_t) \\ \sigma_t^2 &= (1 - t^{-1})\sigma_{t-1}^2 + t^{-1}(\mu_t - u_J(x_t))^2 \end{aligned} \tag{8}$$

Here, $u_J(x_t)$ is the uncertainty of the input x_t. For an input sample, if its uncertainty is highly likely to be greater than that of most unseen samples, we consider it to be informative. To be more concrete, we select the t-th sample if $1 - F_{u_t}(u_J(x_t))^{(T-t)} < 0.5$ unless σ_t is 0, where $F_{u_t}(\cdot)$ is the cumulative density function of the uncertainty distribution u_t. If σ_t is 0, we reject x_t. The agent selects the last sample if all samples are rejected. Whenever a new query is selected and a new period is started, we reset t to 0. With the 'Explorer' strategy, the model can efficiently and effectively choose the useful samples by predicting the informativeness of the unseen ones.

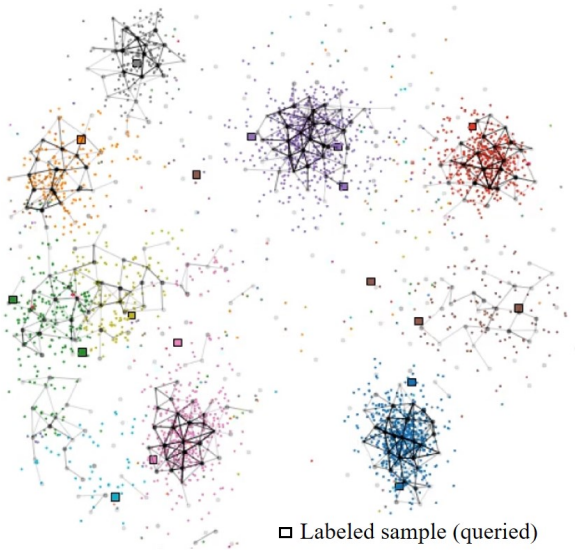

Figure 3: The training results of the NSynth dataset at the 1/500 query frequency and $L = 3$ using 'Explorer' strategy. The color of the dots visualize the classification results. The distribution of the queried data is spread evenly around the boundary of the classes according to the query selection score.

5 Results and Discussion

An example of training results using the NSynth dataset is visualized in Figure 3. Since we only store the representative values in a graph, the number of nodes and edges created is significantly less than the number of training data. For the MNIST dataset, the averaged numbers of nodes over 10 experiments are 264, 377, and 459 when trained with 10k, 30k and 60k samples respectively.

Table 1 summarizes the results of performance evaluation of our model on four datasets according to query selection frequencies and strategies. In all experimental settings, the model that applies message passing showed significantly higher performance than the one without. The accuracies decrease as the frequency of query decreases, but the gap is not significant when the message passing method is applied. When comparing the performance with respect to the query selection strategy, the accuracy of the 'Random' strategy was generally low, and the 'Memory' strategy and the 'Explorer' strategy showed almost similar performance. This is because the 'Explorer' strategy properly estimates the uncertainty distribution of input data and efficiently selects the representative samples based on the remaining query opportunities. In addition, the lower the query frequency, the greater the performance difference according to the query selection strategy, indicating that the query selection strategy can have a significant impact on performance in a situation where the labeled data is extremely scarce.

Query Selection Frequency	Model	MNIST ($\rho = 0.97$)	SVHN ($\rho = 0.95$)	CIFAR-10 ($\rho = 0.95$)	NSynth ($\rho = 0.97$)
Fully Supervised	MLP	98.9±0.0	83.4±2.9	48.8±0.2	77.9±0.1
1 / 1000	A-SOINN (Shen et al. 2011)	62.7±12.1	47.0±9.3	32.0±4.8	56.2±9.9
	LPART (Kim et al. 2020)	62.3±9.8	55.3±5.8	28.6±3.4	58.4±8.3
	MPART ($L = 1$)	61.5±6.8	62.9±6.1	18.5±2.8	45.3±5.2
	MPART ($L = 3$)	**94.6**±4.3	**73.7**±4.2	32.7±3.3	77.1±6.7
	MPART ($L = 5$)	91.9±5.9	70.2±5.7	**34.3**±2.8	**79.0**±7.0
1 / 500	A-SOINN (Shen et al. 2011)	79.1±12.1	53.4±11.3	35.6±4.7	70.5±5.4
	LPART (Kim et al. 2020)	82.6±7.7	68.0±6.0	32.9±3.2	72.8±6.0
	MPART ($L = 1$)	86.1±2.3	73.6±3.2	23.8±2.0	60.8±3.6
	MPART ($L = 3$)	97.3±0.3	**78.5**±2.8	35.9±2.8	84.9±1.6
	MPART ($L = 5$)	**97.4**±0.2	77.0±4.8	**36.4**±2.7	**85.1**±2.4
1 / 100	A-SOINN (Shen et al. 2011)	85.3±10.8	55.6±13.9	35.3±5.3	70.0±7.6
	LPART (Kim et al. 2020)	97.0±0.5	80.1±0.7	38.7±1.3	84.2±1.6
	MPART ($L = 1$)	97.4±0.3	78.5±1.9	35.8±1.2	85.4±1.0
	MPART ($L = 3$)	97.5±0.2	80.6±0.7	39.7±1.4	**88.7**±0.7
	MPART ($L = 5$)	**97.6**±0.2	**80.9**±0.6	**40.6**±1.7	**88.7**±1.0

Table 2: Comparison of classification accuracy (mean ± std) between our model (MPART) and the competitive models. The 'Explorer' strategy was used for query selection of MPART. (unit : %)

The performance of the competitive models such as A-SOINN (Shen et al. 2011) and LPART (Kim et al. 2020) was compared to MPART using various message passing layers L and 'Explorer' strategy in Table 2. The multi-layer perceptron (MLP) model was used as a reference of fully supervised learning, which was trained using all labeled data for each dataset. The same ART-related parameter values were used in MPART and LPART, while A-SOINN parameters were adjusted to achieve its best performance for fair comparison. A-SOINN needs to query the prototype of the most dense node, which is a weighted sum of encoded representations and does not correspond to any input samples. Therefore, in A-SOINN, the prototype of the node was not directly queried, but the most recent input sample that activates the node was queried. In all experimental settings except MLP, the MPART model showed the highest accuracy, and even outperformed the MLP on the NSynth dataset. It was confirmed that the class distribution of the NSynth dataset is imbalanced, and the number of training data was insufficient, so it showed low performance in the MLP. The proposed method is more useful in environments where the number and distribution of classes are not known in advance, as in online learning. As the number of message passing layers increased, the classification performance generally increased, but the results of using 3 and 5 layers were almost the same.

6 Conclusions

We propose Message Passing Adaptive Resonance Theory (MPART) for online active semi-supervised learning, which learns the distribution and topology of the input data online, infers the class of unlabeled data, and selects the informative and representative samples through message passing between nodes on the topological graph. By evaluating our method on datasets including CIFAR-10 and NSynth, we show that it outperforms the competitive models in online active learning environments. This approach reduces the need to create large datasets in advance in order to apply machine learning to various industries. We believe MPART offers new opportunities for machine learning technique to be widely used in real-world applications.

Acknowledgments

This work was partly supported by the Korea government (2019-0-01367-BabyMind, 2015-0-00310-SW.StarLab, 2017-0-01772-VTT, 2018-0-00622-RMI, P0006720-GENKO).

References

Carpenter, G. A.; Grossberg, S.; and Rosen, D. B. 1991. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural networks* 4(6): 759–771.

Douze, M.; Szlam, A.; Hariharan, B.; and Jégou, H. 2018. Low-shot learning with large-scale diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3349–3358.

Engel, J.; Resnick, C.; Roberts, A.; Dieleman, S.; Norouzi, M.; Eck, D.; and Simonyan, K. 2017. Neural audio synthesis of musical notes with wavenet autoencoders. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1068–1077. JMLR. org.

Fritzke, B. 1995. A growing neural gas network learns topologies. In *Advances in neural information processing systems*, 625–632.

Goldberg, A.; Zhu, X.; Furger, A.; and Xu, J.-M. 2011. OASIS: Online Active Semi-Supervised Learning. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, 362–367.

Grossberg, S. 1987. Competitive learning: From interactive activation to adaptive resonance. *Cognitive science* 11(1): 23–63.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* .

Huang, S.-J.; Jin, R.; and Zhou, Z.-H. 2010. Active learning by querying informative and representative examples. In *Advances in neural information processing systems*, 892–900.

Isawa, H.; Tomita, M.; Matsushita, H.; and Nishio, Y. 2007. Fuzzy adaptive resonance theory with group learning and its applications. In *Proc. of International Symposium on Nonlinear Theory and its Applications*, 292–295.

Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2019. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5070–5079.

Kim, J.; Kim, T.; Kim, S.; and Yoo, C. D. 2019. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11–20.

Kim, T.; Hwang, I.; Kang, G.-C.; Choi, W.-S.; Kim, H.; and Zhang, B.-T. 2020. Label Propagation Adaptive Resonance Theory for Semi-Supervised Continuous Learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4012–4016. IEEE.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* .

Kohonen, T. 1990. The self-organizing map. *Proceedings of the IEEE* 78(9): 1464–1480.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278–2324.

Loy, C. C.; Hospedales, T. M.; Xiang, T.; and Gong, S. 2012. Stream-based joint exploration-exploitation active learning. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 1560–1567. IEEE.

Lughofer, E. 2017. On-line active learning: A new paradigm to improve practical useability of data stream modeling methods. *Information Sciences* 415: 356–376.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*.

Settles, B. 2011. From theories to queries: Active learning in practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, 1–18.

Shen, F.; and Hasegawa, O. 2006. An incremental network for on-line unsupervised classification and topology learning. *Neural networks* 19(1): 90–106.

Shen, F.; Ogura, T.; and Hasegawa, O. 2007. An enhanced self-organizing incremental neural network for online unsupervised learning. *Neural Networks* 20(8): 893–903.

Shen, F.; Yu, H.; Sakurai, K.; and Hasegawa, O. 2011. An incremental online semi-supervised active learning algorithm based on self-organizing incremental neural network. *Neural Computing and Applications* 20(7): 1061–1074.

Stretcu, O.; Viswanathan, K.; Movshovitz-Attias, D.; Platanios, E.; Ravi, S.; and Tomkins, A. 2019. Graph Agreement Models for Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, 8710–8720.

Tscherepanow, M. 2010. TopoART: A topology learning hierarchical ART network. In *International Conference on Artificial Neural Networks*, 157–167. Springer.

Weigl, E.; Heidl, W.; Lughofer, E.; Radauer, T.; and Eitzinger, C. 2016. On improving performance of surface inspection systems by online active learning and flexible classifier updates. *Machine Vision and Applications* 27(1): 103–127.

Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434* .