Unsupervised Continual Learning via Self-Adaptive Deep Clustering Approach

Mahardhika Pratama∗, Andri Ashfahani1, Edwin Lughofer2

1SCSE, Nanyang Technological University, Singapore
2DKBMS, Johannes Kepler University, Linz, Austria
mpratama@ntu.edu.sg, andriash001@e.ntu.edu.sg, edwin.lughofer@jku.at

Abstract

Unsupervised continual learning remains a relatively uncharted territory in the existing literature because the vast majority of existing works call for unlimited access of ground truth incurring expensive labelling cost. Another issue lies in the problem of task boundaries and task IDs which must be known for model’s updates or model’s predictions hindering feasibility for real-time deployment. Knowledge Retention in Self-Adaptive Deep Continual Learner, (KIERA), is proposed in this paper. KIERA is developed from the notion of flexible deep clustering approach possessing an elastic network structure to cope with changing environments in the timely manner. The centroid-based experience replay is put forward to overcome the catastrophic forgetting problem. KIERA does not exploit any labelled samples for model updates while featuring a task-agnostic merit. The advantage of KIERA has been numerically validated in popular continual learning problems where it shows highly competitive performance compared to state-of-the-art approaches. Our implementation is available in https://github.com/ContinualAL/KIERA.

1 Introduction

Continual learning is a machine learning field studying a learning model which can handle a sequence of tasks \(T_1, T_2, T_3, \ldots, T_K\) [Parisi et al., 2018] where \(K\) labels the number of tasks. Unlike conventional learning problem with the i.i.d assumption, every task \(T_k\) suffers from non-stationary conditions where there exists drifting data distributions \(P(X, Y)_k \neq P(X, Y)_{k+1}\), emergence of new classes or combination between both conditions. The goal of general continual learning should be to build a model \(f(.)\) which is capable of identifying and adapting to changes without suffering from the catastrophic forgetting problem. It is done on the fly with the absence of samples from previous tasks \(T_{k-1}\).

There has been growing research interest in the continual learning domain in which the main goal is to resolve the issue of catastrophic forgetting thereby actualizing knowledge retention property of a model. These works can be divided into three categories: memory-based approach, regularization-based approach and structure-based approach. Memory-based Approach is designed to handle the continual learning problem with the use of external memory storing past samples. Past samples are replayed along with new samples of the current task such that the catastrophic forgetting problem can be addressed. Regularization-based approach is put forward with the use of additional regularization term in the cost function aiming at achieving tradeoff points between old task and new task. Structure-based approach is developed under the roof of dynamic structure. It adds new network components to cope with new tasks while isolating old parameters from new task to avoid the catastrophic forgetting problem.

The area of continual learning still deserves in-depth study because the vast majority of existing approaches are fully-supervised algorithms limiting their applications in the scarcity of labelled samples. Existing approaches rely on a very strong assumption of task boundaries and task IDs. This assumption is impractical because the point of changes is often unknown and the presence of task IDs for inference imply extra domain knowledge.

An unsupervised continual learning algorithm, namely Knowledge Retention in Self-Adaptive Deep Continual Learner (KIERA), is proposed in this paper. KIERA does not utilize any labelled samples for model updates. That is, labelled samples are only offered to establish clusters-to-classes associations required to perform classification tasks. KIERA is constructed from the idea of self-evolving deep clustering network making use of a different-depth network structure. Every hidden layer generates its own set of clusters and produces its own local outputs. This strategy enables an independent self-evolving clustering mechanism to be performed in different levels of deep embedding spaces. KIERA features an elastic network structure in which its hidden nodes, layers and clusters are self-generated from data streams in respect to varying data distributions.

The parameter learning mechanism of KIERA is devised to achieve a clustering-friendly latent space via simultaneous feature learning and clustering. It puts forward reconstruction loss and clustering loss minimized simultaneously in the framework of stacked autoencoder (SAE) to avoid trivial solutions. The clustering loss is formed as the K-Means
loss [Yang et al., 2017] inducing the clustering-friendly latent space where data samples are forced to be adjacent to the winning cluster, i.e., the most neighboring cluster to a data sample. The centroid-based experience replay strategy is put forward to address the catastrophic interference problem. That is, the sample selection mechanism is carried out in respect to focal-points, i.e., data samples triggering addition of clusters. The selective sampling mechanism is integrated to assure a bounded replay buffer. The advantage of KIERA is confirmed with rigorous numerical study in popular problems where it outperforms prominent algorithms.

This paper presents four major contributions: 1) KIERA is proposed to handle unsupervised continual learning problems; 2) the flexible deep clustering approach is put forward in which hidden layers, nodes and clusters are self-evolved on the fly; 3) the different-depth network structure is designed making possible an independent clustering module to be carried out in different levels of deep embedding space; 4) the centroid-based experience replay method is put forward to address the catastrophic interference problem.

2 Related Works

Memory-based Approach: iCaRL [Rebuffi et al., 2017] exemplifies the memory-based approach in the continual learning using the exemplar set of each class. The classification decision is calculated from the similarity degree of a new sample to the exemplar set. Another example of the memory-based approach is Gradient Episodic Memory (GEM) [Lopez-Paz and Ranzato, 2017] where past samples are stored to calculate the forgetting case. A forgetting case can be examined from the angle between the gradient vector and the proposed update. This work is extended in [Chaudhry et al., 2019] and called Averaged GEM (AGEM). Its contribution lies in the modification of the loss function to expedite the model updates. Deep Generative Replay (DGR) [Shin et al., 2017] does not utilize external memory to address the catastrophic forgetting problem rather makes use of generative adversarial network (GAN) to create representation of old tasks. The catastrophic forgetting is addressed by generating pseudo samples for experience replay mechanism. Our work is framed under this category, because it can be executed without the presence of task IDs or task’s boundaries.

Regularization-based Approach: Elastic Weight Consolidation (EWC) [Kirkpatrick et al., 2016] is a prominent regularization-based approach using the L2-like regularization approach constraining the movement of important network parameters. It constructs Fisher Information Matrix (FIM) to signify the importance of network parameters. Synaptic Intelligence (SI) [Zenke et al., 2017] offers an alternative approach where it utilizes accumulated gradients to quantify importance of network importance instead of FIM incurring prohibitive computational burdens. Memory Aware Synapses (MAS) [Aljundi et al., 2018] is an EWC-like approach with modification of parameter importance matrix using an unsupervised and online criterion. EWC has been extended in [Schwarz et al., 2018] named onlineEWC where Laplace approximation is put forward to construct the parameter importance matrix. Learning Without Forgetting (LWF) is proposed in [Li and Hoiem, 2016] where it formulates a joint optimization problem between the current loss function and the knowledge distillation loss [Hinton et al., 2015]. This neuron-based regularization notion is presented in [Paik et al., 2020] where the regularization is performed by adjusting the learning rates of stochastic gradient descent. In [Mao et al., 2021], the inter-task synaptic mapping is proposed. Notwithstanding that the regularization-based approach is computationally efficient, this approach requires the task boundaries and task IDs to be known.

Structure-based Approach: the catastrophic forgetting is overcome in progressive neural networks (PNNs) [Rusu et al., 2016] by introduction of new column for every new task while freezing old network parameters. This approach is, however, not scalable for large-scale problem since the structural complexity linearly grows as the number of tasks. This drawback is addressed in [Lee et al., 2018] with the use of selective retraining approach. Learn-to-grow is proposed in [Li et al., 2019] where it utilizes the neural architecture search to find the best network structure of a given task. The structure-based approach imposes expensive complexities.

3 Problem Formulation

The continual learning problem aims to build a predictive model \(f(.)\) handling streaming tasks \(T_1, T_2, T_3, ..., T_K\) where \(K\) denotes the number of tasks. Unlike conventional problems assuming the i.i.d condition, each task \(T_k\) is influenced by non-stationary conditions. There exist two types of changes in the continual learning where the first one is known as the problem of changing data distributions while the second one is understood as the problem of changing target classes. The problem of changing data distributions or the concept drift problem is defined as a change of joint probability distribution \(P(X,Y)_k \neq P(X,Y)_{k+1}\). The problem of changing target classes refers to different target classes of each task. Suppose that \(L_k \) and \(L_{k'}\) stand for the label sets of the \(k-th\) task and the \(k'-th\) task, \(\forall k, k' \in \{1,2,3,...,K\} \) for \(k \neq k'\). This problem is also known as the incremental class problem. Each task normally consists of paired data samples \(T_k = \{x_n, y_n\}_{n=1}^{N_k}\) where \(N_k\) denotes the size of the \(k-th\) task. \(y_n\) is the \(n-th\) input image while \(x_n\) is the target vector formed as a one-hot vector. This assumption, however, does not apply in the unsupervised continual learning dealing with the scarcity of labelled samples. The access of true class labels are only provided for the initial batch of each task to associate the cluster’s centroids with classes \(B_k^0 = \{x_n, y_n\}_{n=1}^{N_k^0}\) while the remainder of data samples arrive with the absence of labelled samples \(B_k^j = \{x_n\}_{n=1}^{N_k^j}\). Note that \(N_k^0 + N_k^1 + N_k^2 + ... + N_k^j = N_k\).

4 Network Structure

KIERA is structured as a SAE comprising two components: feature extraction layer and fully-connected layer. The feature extraction layer is built upon a convolutional layer or a nonlinear layer. It maps an input image \(x\) into a latent input
vector $Z \in \mathbb{R}^{u'}$ where $u'$ denotes the number of natural features. The fully connected layer performs the encoding and encoding steps across $L_{sae}$ layers producing a reconstructed latent input vector $\hat{Z}$. The reconstructed latent input vector $\hat{Z}$ is further fed to transposed convolutional or nonlinear layers generating the reconstructed input image $\hat{x}$.

Suppose that $h_{l-1} \in \mathbb{R}^{u}$ stands for the latent input vector of the $l-$th encoder where $h_{0} = Z$, the encoder projects $h_{l-1}$ to a lower dimensional latent space $h_{l} = r(W_{enc}^{l}h_{l-1} + b_{l})$ while the decoder reconstructs the latent input vector $h_{l} = r(W_{dec}^{l}+ c_{l})$. $r(.)$ is a ReLU activation function inducing a nonlinear mapping. $W_{enc}^{l} \in \mathbb{R}^{R_{l} \times u}$, $b_{l} \in \mathbb{R}^{R_{l}}$ are the connective weight and bias of the $l-$th encoder while $W_{dec}^{l} \in \mathbb{R}^{u \times R_{l}}$, $c_{l} \in \mathbb{R}^{u}$ are the connective weight and bias of the $l-$th decoder. $R_{l}, u_{l}$ are respectively the number of hidden nodes of the $l-$th layer and the number of input features of the $l-$th layer. The tied-weight constraint is applied here to avoid the over-training problem $W_{enc}^{l} = (W_{dec}^{l})^{T}$. The SAE carries out the non-linear dimension reduction step preventing the trivial solution as the case of the linear dimension reduction mechanism while the training process occurs in the greedy layer-wise fashion.

The unique facet of KIERA lies in the different-depth network structure in which the clustering mechanism takes place in every hidden layer of the fully connected layer $h_{l}(.)$ thereby producing its own local output. It distinguishes itself from the conventional deep clustering network where the clustering mechanism happens at the bottleneck layer only. The cluster’s allegiance [Smith et al., 2019] is expressed:

$$a_{l} = \frac{\exp(||C_{s,l} - h_{l}||)}{\max_{s=1,...,Cl_{s}} \exp(||C_{s,l} - h_{l}||)}$$ (1)

where $C_{s,l}$ denotes the centroid of the $s$–th cluster of the $l-$th layer and $Cl_{s}$ stands for the number of clusters of the $l-$th layer. The cluster’s allegiance is averaged across all labelled samples $B_{0}^{k} = \{x_{i}, y_{i}\}_{i=1}^{N_{label}^{k}}$ to indicate cluster’s tendency to a target class. Let $N_{label}^{k,o}$ be the number of initially labelled samples of the $k-$th task falling into the $o-$th target class where $N_{o,0}^{k} + N_{o,0}^{k} + ... + N_{o,m}^{k} = N_{o}^{k}$, the averaged cluster allegiance is written:

$$A_{l} = \frac{\sum_{o=1}^{N_{o}^{k}} a_{l}^{o}}{N_{o}^{k}}$$ (2)

(2) implies a low cluster allegiance of unclean clusters being populated by data samples of mixed classes. The score of the $o-$th target class of the $l-$th layer is computed by combining the cluster’s allegiance and the distance to a data sample $h_{l}$:

$$Score_{l}^{o} = \text{Softmax} \sum_{s=1}^{Cl_{s}} \exp(-||h_{l} - C_{s,l}||A_{l}^{o})$$ (3)

The local output of the $l-$th layer can be found by taking a maximum operation $\max_{o=1,...,m} Score_{l}^{o}$. The use of $\text{Softmax}(.)$ operation assures the partition of unity as well as the uniform range of $Score_{l}^{o}$ across $L_{sae}$ layers. The final output is produced by aggregating the local outputs using the summation operation as follows:

$$\hat{y} = \max_{o=1,...,m} \sum_{l=1}^{L_{sae}} Score_{l}^{o}$$ (4)

Labelled samples of each task $B_{0}^{k}$ are only exploited to associate a cluster to a specific class, i.e., the calculation of cluster allegiance (2) thereby satisfying the unsupervised learning requirement. Another typical characteristic of KIERA exists in its self-evolving property where $R_{l}, L_{sae}, Cl_{s}$ are not deterministic rather self-evolved from data streams.

### 4.2 Structural Learning Mechanism

**Evolution of Hidden Nodes:** KIERA features an elastic network width where new nodes are dynamically added while inactive nodes are pruned. This strategy is underpinned by the network significance (NS) method [Pratama et al., 2019] estimating the generalization power of a network based on the bias and variance decomposition approach. New nodes are introduced in the case of high bias to cope with the underfitting issue while inconsequential nodes are discarded in the case of high variance to overcome the overfitting situation. Because of the absence of any labelled samples for model’s updates, the network bias and variance are estimated from reconstructions losses of a layer $NS = Bias^{2} + Var = (E[h_{l}^{t}] - h_{l}^{t})^{2} + (E[h_{l}^{t}] - E[h_{l}^{t}]^{2})^{2}$. Note that KIERA adopts the greedy layer-wise training approach. The network bias and variance are formalized under a normal distribution $p(x) = N(\mu, \sigma^{2})$ and the ReLU activation function $r(.)$ where $\mu, \sigma^{2}$ are the mean and standard deviation of data samples. The modified statistical process control (SPC) method [Gama, 2010], a popular approach for anomaly detection, is utilized to signal the high bias condition and the high variance condition as follows:

$$\mu_{bias}^{n} + \sigma_{bias}^{n} \geq \mu_{bias}^{min} + (\ln(l) + 1) \times k_{1} \sigma_{bias}^{min}$$ (5)

$$\mu_{var}^{n} + \sigma_{var}^{n} \geq \mu_{var}^{min} + 2 \times (\ln(l) + 1) \times k_{2} \sigma_{var}^{min}$$ (6)

where the main modification of the conventional SPC lies in $k_{1} = 1.3 \exp(-Bias^{2}) + 0.7$ and $k_{2} = 1.3 \exp(-Var^{2}) + 0.7$ leading to dynamic confidence levels. This idea enables new nodes to be created in the case of high bias and inconsequential nodes to be removed in the case of high variance. The Xavier’s initialization approach is applied for the sake of initialization. Conversely, the pruning process targets an inactive node having the least statistical contribution $\text{min}_{l=1,...,R_{l}} E[h_{l}^{t}]$. The term $(\ln(l) + 1)$ is inserted to guarantee the nonlinear feature reduction step due to $R_{l} < R_{l-1}$. The node growing strategy becomes insensitive as the increase of network depth. The term 2 is applied in (6) to prevent a new node to be directly eliminated.

**Evolution of Hidden Layer:** KIERA characterizes a variable network depth where a drift detection technique is applied to expand the network depth $L_{sae}$. A new layer is incorporated if a drift is detected. Note that insertion of a new layer is capable of boosting the network’s capacity more significantly than that introduction of new nodes. Since no labelled samples are not available at all, the drift detection mechanism focuses on
the dynamic of latent input features $Z$ extracted by the feature extraction layer. Nevertheless, the characteristic of latent input features are insensitive to changing data distributions. We apply the drift detection method for the last two consecutive data batches $A = [Z^i_{t-1}, Z^i_t]$ here. The drift detection mechanism first finds a cutting point $c$ signifying the increase of population means $\tilde{A} + \epsilon_a \leq \tilde{B} + \epsilon_B$. $\tilde{A}, \tilde{B}$ are the statistics of two data matrices $A \in \mathbb{R}^{N \times u'}$ and $B \in \mathbb{R}^{N \times u'}$ respectively while $\epsilon_a, \epsilon_B = \sqrt{\frac{2 \times \text{size} \ln \frac{1}{\alpha}}{\lambda}}$ is their corresponding Hoeffding’s bounds and $\alpha$ is the significance level. $\text{size}$ is the size of data matrix of interest.

Once finding the cutting point, the data matrix $A$ is divided into two matrices $B$ and $C^{(2N^k \times \text{cut}) \times u'}$. A drift is confirmed if $|B - C| \geq \epsilon_d$ whereas a warning is flagged by $\epsilon_d \geq |B - C| \geq \epsilon_w$. $\epsilon_d, \epsilon_w = (b - a)\sqrt{\frac{\text{size} \times \text{cut}}{2 \times \text{size} \times \text{cut}} \ln \frac{1}{\alpha_d, \alpha_w}}$ where $[a, b]$ represents the interval of the data matrix $A$ and $\alpha_d < \alpha_w$. The warning condition is a transition situation where the drift condition is to be confirmed by a next stream. A new layer is added if a drift condition is signalled. The number of nodes of the new layer is set at the half of that of the previous layer. Addition of a new layer does not impose the catastrophic forgetting because of the different-depth network structure, i.e., every layer to produce its local output.

**Evolution of Hidden Clusters:** the growing strategy of hidden clusters relies on the compatibility measure examining a distance between a latent sample $h^l$ to the nearest cluster. A new cluster is added if a data sample is deemed remote to the zone of influence of any existing clusters as follows:

$$\min_{s=1,...,C_{ts}} D(C_{s,l}, h^l) > \mu_{D_{s,l}} + k_3 * \sigma_{D_{s,l}}$$

where $\mu_{D_{s,l}}, \sigma_{D_{s,l}}$ are the mean and standard deviation of the distance measure $D(C_{s,l}, h^l)$ while $k_3 = 2 \exp -||h^l - C_{s,l}|| + 2$. (7) is perceived as the SPC method with the dynamic confidence degree assuring that a cluster is added if it is far from the coverage of existing clusters. (7) hints the presence of a new concept unseen in the previous observations. Hence, a data sample $h^l$ can be regarded as a focal point. A new cluster is integrated by setting the current sample as a centroid of a new cluster $C_{(s+1),l} = h^l$ while its cardinality is set as $C_{ar(s+1),l} = 1$. Note that the clustering process occurs in different levels of deep embedding space.

### 4.3 Parameter Learning Mechanism

**Network Parameters:** the parameter learning strategy of network parameters performs simultaneous feature learning and clustering in which the main goal is to establish clustering-friendly latent spaces [Yang et al., 2017]. The loss function comprises two terms: reconstruction loss and clustering loss written as follows:

$$L_{all} = \frac{\sum_{l=1}^{L_{all}} (L(h^l, \hat{h}^l) + \lambda \frac{1}{2} ||h^l - C_{win,l}||^2)}{L_t}$$

where $\lambda$ is a tradeoff constant controlling the influence of each loss function. (8) can be solved using the stochastic gradient descent optimizer where $L_1$ is executed in the end-to-end fashion while $L_2$ is carried out per layer, i.e., greedy layer wise fashion. $C_{win,l}$ stands for the centroid of the winning cluster of the $l$ - th layer, i.e., the closest cluster to a latent sample $\text{win} \leftarrow \min_{s=1,...,C_{ts}} ||h^l - C_{s,l}||$. The first and second terms $L(x, \hat{x}), L(h^l, \hat{h}^l)$ are formed as the mean squared error (MSE) loss function where $L(x, \hat{x})$ assures data samples to be mapped back to their original representations while $L(h^l, \hat{h}^l)$ is to extract meaningful latent features in every hidden layer of SAE. The last term $||h^l - C_{win,l}||^2$ is known as the distance loss or the K-means loss producing the K-means friendly latent space. That is, a latent sample is driven to be close to the centroid of the winning hidden cluster resulting in a high cluster probability, i.e., the assignment probability of a data sample to its nearest cluster is high. The multiple nonlinear mapping via the SAE also functions as nonlinear feature reduction addressing the trivial solution as often the case of linear mapping.

**Cluster Parameters:** the parameter learning of hidden clusters is executed if (7) is violated meaning that a latent sample $h^l$ is sufficiently adjacent to existing clusters. This condition only calls for association of the current latent sample to the winning cluster, the nearest cluster, fine-tuning the centroid of the winning cluster as follows:

$$C_{win,l} = C_{win,l} - \frac{(C_{win,l} - h^l)}{C_{ar_{win,l}} + 1}; C_{ar_{win,l}} = C_{ar_{win,l}} + 1$$

The tuning process improves the coverage of the winning cluster to the current sample and thus enhances the cluster’s posterior probability $P(C_{win,l}|x)$. The intensity of the tuning process in ((9)) decreases as the increase of cluster’s cardinality thereby expecting convergence. Only the winning cluster is adjusted here to avoid the issue of cluster’s overlapping. The alternate optimization strategy is implemented in KIERA where the cluster’s parameters are fixed while adjusting the network parameters and vice versa.

#### 4.4 Centroid-based Experience Replay

KIERA adopts the centroid-based experience replay strategy to address the catastrophic forgetting problem. This mechanism stores focal-points of previous tasks $T_1, T_2, ..., T_{k-1}$ into an episodic memory interleaved along with samples of the current task. $T_1$ to overcome the catastrophic interference issue. Note that unlabelled images are retained in the episodic memory. The sample selection mechanism is driven by the cluster growing strategy in (7). That is, an image is considered as a focal-point $x^*$ thus being stored in the episodic memory $M_{k_{em}} = M_{k_{em}} \cup x^*$ provided that (7) is observed. The sample selection strategy is necessary to control the size of memory as well as to substantiate the efficacy of experience replay mechanism making sure conservation of important images, focal points. Focal points represent varieties of concepts seen thus far.

The size of memory grows as the increase of tasks making the experience replay mechanism intractable for a large problem. On the other hand, the network parameters and the cluster parameters are adjusted with recent samples thereby paving possibility of the catastrophic forgetting issue. A selective sampling strategy is implemented in the centroid-based experience replay method. The goal of the selec-
tive sampling strategy is to identify the most forgotten focal points in the episodic memory for the sake of experience replay while ignoring other focal-points thereby expediting the model’s updates. The most forgotten samples are those focal points which do not receive sufficient coverage of existing clusters. That is, the cluster posterior probabilities $P(C_{s,l}|x^*) = \exp \frac{-||C_{s,l} - h^l_x||}{N_{em}}$ of the most forgotten samples are below a midpoint $mid$ defined:

$$mid = \frac{\sum_{n=1}^{N_{em}} \max_{s=1,...,C_{ls}} P(C_{s,l}|x^*)}{N_{em}}$$

(10)

where $N_{em}$ denotes the size of episodic memory. The midpoint indicates the average level of coverage to all focal-points in the episodic memory. A focal-point is included into the replay buffer $B^* = B^* \cup x^*$ to be replayed along with the current concept if it is not sufficiently covered $P(C_{s,l}|x^*) < mid$. Finally, the centroid-based experience replay strategy is executed by interleaving images of current data batch $B^k$ and replay buffer $B^*$ for the training process $B^k \cup B^*$. The most forgotten focal points are evaluated by checking the current situation of network parameters and cluster parameters portrayed by the cluster posterior probability.

5 Proof of Concepts

5.1 Datasets

The performance of KIERA is numerically validated using four popular continual learning problems: Permuted MNIST (PMNIST), Rotated MNIST (RMNIST), Split MNIST (SMNIST) and Split CIFAR10 (SCIFAR10). PMNIST is constructed by applying four random permutations to the original MNIST problem thus leading to four tasks $K = 4$. RMNIST applies rotations with random angles to the original MNIST problem: [0, 30], [31, 60], [61, 90], [91, 120] thus creating four tasks $K = 4$ in total. The two problems characterize the concept drift problem in each task. The SMNIST presents the incremental class problem of five tasks where each task presents two mutually exclusive classes, (0/1, 2/3, 4/5, 6/7, 8/9). As with the SMNIST problem, the SCIFAR10 also features the incremental class problem of five tasks where each task possesses two non-overlapping target classes.

5.2 Implementation Notes of KIERA

KIERA applies an iterative training strategy for the initial training process of each task and when a new layer is added. This mechanism utilizes $N^{k}_{n_{ini}}$ unlabelled samples to be iterated across $E$ number of epochs where $N^{k}_{n_{ini}}$ and $E$ are respective selected as 1000 and 50 respectively. The training process completely runs in the single-pass training mode afterward. $B^0_k = \{x_i, y_i\}_{i=1}^{N^k_0}$ labelled samples are revealed for each task to associate a cluster with a target class in which for every class 100 labelled samples are offered. Hence, $N^k_0$ is 200 for the SMNIST problem and the SCIFAR10 while $N^k_0$ is 1000 for RMNIST and PMNIST.

5.3 Network Structure

The feature extraction layer of KIERA is realized as the convolutional neural network (CNN) where the encoder part utilizes two convolutional layers with 16 and 4 filters respectively and max-pooling layer in between. The decoder part applies transposed convolutional layers with 4 and 16 filters respectively. For PMNIST problem, the feature extraction layer is formed as a multilayer perceptron (MLP) network with two hidden layers where the number of nodes are selected as [1000, 500]. Note that the random permutation of PMNIST requires a model to consider all image pixels as done in MLP. The fully connected layer is initialized as a single hidden layer $L_{sar} = 1$ with 96 hidden nodes $R_1 = 96$. The ReLU activation function is applied as the hidden nodes and the sigmoid function is implemented in the decoder output to produce normalized reconstructed images.

5.4 Hyper-parameters

The hyper-parameters of KIERA are fixed across the four problems to demonstrate non ad-hoc characteristic. The learning rate, momentum coefficient and weight decay strength of the SGD method are selected as 0.01, 0.95, $5 \times 10^{-5}$ while the significant levels of the drift detector are set as $\alpha = 0.001, \alpha_0 = 0.001, \alpha_0 = 0.005$. The trade-off constant is chosen as $\lambda = 0.01$.

5.5 Baseline Algorithms

KIERA is compared against Deep Clustering Networks (DCN) [Yang et al., 2017], AE+KMeans and STAM [Smith et al., 2019]. DCN adopts simultaneous feature learning and clustering where the cost function is akin to KIERA (8). Nevertheless, it adopts a static network structure. AE+KMeans performs the feature learning first using the reconstruction loss while the KMeans clustering process is carried out afterward. STAM [Smith et al., 2019] relies on an irregular feature extraction layer based on the concept of patches while having a self-clustering mechanism as with KIERA. DCN and AE+KMeans are fitted with the Learning Without Forgetting (LWF) method [Li and Hoiem, 2016] and Synaptic Intelligence (SI) method [Zenke et al., 2017] to overcome the catastrophic forgetting problem. DCN and AE+KMeans make use of the same network structure as KIERA to ensure fair comparison. The regularization strength of LWF is set as $\beta = 5$ while it is allocated as 0.2 for the first task and 0.8 for the remaining task in SI method.

The hyper-parameters of STAM are selected as per their original values but hand-tuned if its performance is surprisingly poor. The baseline algorithms are executed in the same computational environments using their published codes. The performance of consolidated algorithms are examined using four evaluation metrics: prequential accuracy (PreAcc), task accuracy (Task Acc), backward transfer (BWT) and forward transfer (FWT). PreAcc measures the classification performance of the current task while Task Acc evaluates the classification performance of all tasks after completing the learning process. BWT and FWT are put forward in [Lopez-Paz and Ranzato, 2017] where FWT indicates knowledge transfer across task while BWT reveals knowledge retention of a model after learning a new task. BWT and FWT ranges in $-\infty$ and $+\infty$ with a positive high value being the best value. All consolidated algorithms are run five times where the averages are reported in Table 1.
5.6 Numerical Results

Referring to Table 1, KIERA delivers the highest Preq accuracy and FWT in the rotated MNIST problem with statistically significant margin while being on par with STAM in the context of Task Acc. KIERA outperforms other algorithms in the PMNISt problem in which it obtains the highest Task Acc and Preq Acc with substantial differences to its competitors. Although the FWT of DCN+LWF is higher than KIERA, its Task Acc and its Preq Acc are poor. Similar finding is observed in the SCIFAR10 problem, where KIERA outperforms other algorithms in the Preq Acc, Task Acc and FWT with statistically significant differences. KIERA does not perform well only on the SMNIST problem but it is still much better than DCN and AE+KM.

Despite its competitive performance, STAM takes advantage of a non-parametric feature extraction layer making it robust against the issue of catastrophic forgetting. This module, however, hinders its execution under the GPU environments thus slowing down its execution time. The advantage of clustering-based approach is seen in the FWT aspect. Although STAM adopts the clustering approach, its network parameters are trained with the absence of clustering loss. The self-evolving network structure plays vital role here where Task Acc and Preq Acc of KIERA and STAM outperforms DCN and AE+KM. Especially, the execution of centroid-based experience replay is proposed to address the catastrophic forgetting problem. Our numerical study with four popular continual learning problems confirms the efficacy of KIERA in attaining high Preq Acc, high Task Acc and high FWT. KIERA delivers higher BWT than those using SI and LWF. The advantage of structural learning mechanism is also demonstrated in our numerical study where it produces significantly better performance than those static network structure. KIERA is capable of learning and predicting without the presence of Task ID and Task’s boundaries. Our memory analysis deduces the effectiveness of the centroid-based experience replay in which the size of replay buffer is bounded. The number of forgotten samples is high in the beginning of each task but reduces as the execution of centroid-based experience replay.

Table 2 reports the network structures of KIERA and the episodic memory. The structural learning of KIERA generates a compact and bounded network structure where the number of hidden nodes, hidden layer and hidden clusters are much less than the number of data points. The hidden layer evolution is seen in the PMNISt problem where additional layers are inserted. The centroid-based experience replay excludes \( N_{init} \) unlabelled samples for the pre-training phase of each task. As a result, the focalpoints of the episodic memory are less than those the number of clusters.

Fig. 1 exhibits the evolution of episodic memory across different learning algorithms. It is shown that the episodic memory grows in much faster rate than the replay buffer size. This trend becomes obvious as the increase of the tasks. This finding is reasonable because each task possesses concept changes inducing addition of new clusters and thus focal points of the episodic memory. The selective sampling is capable of finding the most forgotten samples thus leading to bounded size of the replay buffer. The number of forgotten samples is high in the beginning of each task but reduces as the execution of centroid-based experience replay.
References

[Aljundi et al., 2018] Rahaf Aljundi, F. Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. ArXiv, abs/1711.09601, 2018.

[Chaudhry et al., 2019] Arslan Chaudhry, Marc’Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.

[Gama, 2010] Joao Gama. Knowledge Discovery from Data Streams. Chapman & Hall/CRC, 1st edition, 2010.

[Hinton et al., 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[Kirkpatrick et al., 2016] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. 2016.

[Lee et al., 2018] Jeongtae Lee, Jaehong Yoon, E. Yang, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. ArXiv, abs/1708.01547, 2018.

[Li and Hoiem, 2016] Zhizhong Li and Derek Hoiem. Learning without forgetting. 2016.

[Li et al., 2019] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. volume 97 of Proceedings of Machine Learning Research, pages 3925–3934, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[Lopez-Paz and Ranzato, 2017] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17, page 6470–6479, Red Hook, NY, USA, 2017. Curran Associates Inc.

[Mao et al., 2021] Fubing Mao, Weiwei Weng, Mahardhika Pratama, and Edward Yapp Kien Yee. Continual learning via inter-task synaptic mapping. Knowledge-Based Systems, 222:106947, 2021.

[Paik et al., 2020] Inyoung Paik, Sangjun Oh, Taeyeong Kwak, and Injung Kim. Overcoming catastrophic forgetting by neuron-level plasticity control. In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 5339–5346. AAAI Press, 2020.

[Parisi et al., 2018] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. 2018.

[Pratama et al., 2019] Mahardhika Pratama, Choiru Za’in, Andri Ashfahani, Yew Soon Ong, and Weiping Ding. Automatic construction of multi-layer perceptron network from streaming examples. 2019.

[Rebuffi et al., 2017] Sylvestre-Alvise Rebuffi, A. Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5533–5542, 2017.

[Rusu et al., 2016] Andrei A. Rusu, Neil C. Rabinowitz, G. Desjardins, Hubert Soyer, James Kirkpatrick, K. Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. ArXiv, abs/1606.04671, 2016.

[Schwarz et al., 2018] Jonathan Schwarz, Jelena Luketina, Wojciech M. Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. 2018.

[Shin et al., 2017] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. 2017.

[Smith et al., 2019] James Smith, Seth Baer, Zsolt Kira, and Constantine Dovrolis. Unsupervised continual learning and self-taught associative memory hierarchies. In 2019 International Conference on Learning Representations Workshops, 2019.

[Yang et al., 2017] Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In Doina Precup and Yee Whye Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 3861–3870. PMLR, 06–11 Aug 2017.

[Zenke et al., 2017] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. 2017.