Lifelong Learning Needs Sleep: Few-Shot Incremental Learning Enhanced by Sleep

Yuma Kishimoto and Koichiro Yamauchi[a]

Department of Computer Science, Chubu University, Kasugai, Japan

-

Keywords: Sleep, Lifelong Learning, Incremental Learning, Catastrophic Forgetting, Few-Shot Leanring, Incremental Few-Shot Learning.

Abstract: Catastrophic forgetting due to incremental learning in neural networks is a serious problem. We demonstrate that introducing a sleep period can address this issue from two perspectives. First, it provides a learning period for re-learning old memories. Second, it allows for time to process new learning. We applied a VAE, enhanced by an adapter, for incremental learning of new samples and generating valid samples from a few learning examples. These generated samples are used for the neural network's re-learning, which also contributes to improving its generalization ability. The experimental results suggest that this approach effectively realizes few-shot incremental learning.

1 INTRODUCTION

Artificial intelligence (AI) has rapidly advanced with improvements in computing power and storage capacity. Deep learning, a core AI technology, achieves high accuracy through offline learning by layering neural networks.

Some applications, however, require incremental learning to handle new tasks. Mixing past training data with new data demands substantial memory resources, making it impractical in many real-world scenarios. Lifelong learning models have been developed to address this by enabling incremental learning and recognition.

Lifelong learning models have evolved significantly. However, they often struggle with catastrophic forgetting, which is a critical issue in incremental learning. Existing lifelong learning models employ various techniques to mitigate this problem, such as replay mechanisms(e.g., replay-buffer, naive rehearsal(Hsu et al., 2018) , VQ based(Hayes et al., 2019), GAN(Shin et al., 2018)(Lesort et al., 2019), VAE(van de Ven et al., 2020), FearNet(Kemker and Kanan, 2018)), spontaneous firing(Golden et al., 2022), and weight consolidation strategies such as EWC(Kirkpatrick et al., 2017) (Rusu et al., 2016), Packnet (Mallya and Lazebnik, 2018).

Despite progress, no model has simultaneously achieved incremental end-to-end learning, knowledge reformulation, and immediate response in a short time. To address this, we propose a novel approach introducing sleep periods in lifelong learning. These periods enable relearning old memories and reorganizing existing knowledge.

Our model uses an adapter-enhanced VAE to record past samples and generate pseudosamples for data augmentation. The VAE encoder also serves as the backbone for classification tasks, ensuring model compactness and quick responses.

This method offers an integrated solution that enables effective incremental learning, enhances knowledge reformulation, and ensures rapid inference after observing new instances.

Although lifelong learning methods based on self-organization(e.g. (Parisi et al., 2017)) display properties similar to ours, they do not use knowledge reorganization. In our model, the sleep process realizes knowledge reorganization, which contributes to the acquisition of high generalization capability.

2 RELATED WORKS

This section compares the proposed method with existing methods related to few-shot learning, incremental learning, and sleep-based learning.

[a] https://orcid.org/0000-0003-1477-0391

Kishimoto, Y. and Yamauchi, K.
Lifelong Learning Needs Sleep: Few-Shot Incremental Learning Enhanced by Sleep.
DOI: 10.5220/0013170200003905

In *Proceedings of the 14th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2025)*, pages 243-250
ISBN: 978-989-758-730-6; ISSN: 2184-4313

2.1 Incremental Learning

Incremental or continual learning methods allow new samples to be learned by an already-trained network. However, this is always accompanied by catastrophic forgetting(French, 1999).Various methods have been proposed to solve this problem.

2.1.1 Replay Model

Naive Rehearsal (Hsu et al., 2018) and Remind(Hayes et al., 2019) address catastrophic forgetting by replaying past data alongside new data. Methods utilizing VQ improve efficiency by compressing data and reducing memory usage, but increasing past data raises memory consumption, making them unsuitable for resource-limited environments. To solve this problem, generative replay models were developped.(Shin et al., 2018)(Kemker and Kanan, 2018)(Lesort et al., 2019)(Liu et al., 2020)(van de Ven et al., 2020).

These methods, however, require training a neural network as a replay buffer. To prevent catastrophic forgetting, many past samples must be mixed during training, leading to high computational costs.

2.1.2 Weight Consolidation

Catastrophic forgetting occurs when modifying weight connections critical to past knowledge. Freezing these connections during new learning retains memories. EWC(Kirkpatrick et al., 2017) and synaptic intelligence(Zenke et al., 2017) minimize the impact of new tasks by restricting changes to important weights, while PackNet(Mallya and Lazebnik, 2018) partitions network capacity by masking critical weights for each task, enabling continuous learning. This method preserves past knowledge by fixing task-specific weights. However, these approaches are unsuitable for environments with incrementally provided datasets because fixed weights cannot be updated, limiting their adaptability.

2.2 Methods that Mimic Sleep

Many organisms have sleep periods, often remaining inactive during them. Inspired by this, several methods introduce periodic learning into machine learning, enabling continuous incremental learning.

2.2.1 Incremental Learning with Sleep (ILS)

The ILS model utilizes two types of radial basis functions (RBFs) (Yamauchi and Hayami, 2007). During daytime learning, one RBF incrementally learns new samples by adding RBFs, while the other prunes redundant RBFs using pseudo-samples. After nighttime learning, all daytime RBFs are replaced with parameters from the night-learned RBF. This cycle repeats daily. The system showed improved generalization after nighttime learning compared to daytime learning. However, its simple network structure limits its recognition accuracy.

2.2.2 Sleep Models Using Spiking Neuron Models

The spiking neuron model mimics brain sleep processes, reproducing synaptic firing to prevent catastrophic forgetting during continuous learning. The multilayer spiking neural network (Golden et al., 2022) uses spontaneous firing during sleep to form shared synaptic weights, ensuring existing tasks are preserved when new tasks are added. By integrating a pseudo-sleep process for re-learning, this approach reviews past tasks and mitigates catastrophic forgetting. While still in its early stages, this model has demonstrated memory retention during sleep, but the evolution of learning outcomes requires further analysis.

2.2.3 FEARNet

FEARNet(Kemker and Kanan, 2017) is a continuous learning approach that combines short- and long-term memory networks to adapt to new tasks while retaining prior knowledge. It uses ResNet as the backbone for feature extraction. As new data are introduced, features are pre-extracted, and knowledge accumulates in both memory networks, leveraging replay mechanisms for learning.

While ResNet provides strong feature extraction, FEARNet relies on pre-extracted features, making it less flexible for adapting to diverse datasets and tasks. This reliance on ResNet limits its scalability despite its strengths in continuous learning.

2.3 Positioning of the Proposed Method

As detailed in 3.1,this paper proposes a continual few-shot learning method using VAE and k-nearest neighbors (K-NN). Features are extracted by VAE, with an adapter layer mitigating catastrophic forgetting. Unlike FearNet, VAE handles feature extraction flexibly, enabling efficient task adaptation.

The method employs a two-stage learning process: daytime learning for new tasks and nighttime learning for reviewing and reconstructing knowledge. This approach efficiently adapts to continual tasks while preserving past knowledge

Table 1 compares the proposed method with major few-shot continual learning methods, highlight-

Table 1: Comparison of Methods.

Method	Catastrophic Forgetting Prevention	Scalability	Knowledge re-organization	Computation Resource Efficiency	Response time
Generative Replay models	○	○	○	△	△
Naive Rehearsal	○	△	○	△	△
VQ-based Methods	○	△	△	△	△
Multi-layer Spiking Neural Network	△	△	○	△	△
EWC	△	△	×	△	△
PackNet	○	△	×	△	△
FEARNet	○	×	△	△	○
Proposed Method	○	○	○	○	○

ing their characteristics. These methods are evaluated based on network size efficiency, suppression of catastrophic forgetting, scalability, and computational resource efficiency. The proposed method excels in scalability, knowledge re-organization, and computational efficiency in a continual learning environment. In the table, ○ indicates superiority, △ average performance, and × poor performance.

3 PROPOSED METHOD

3.1 Overview

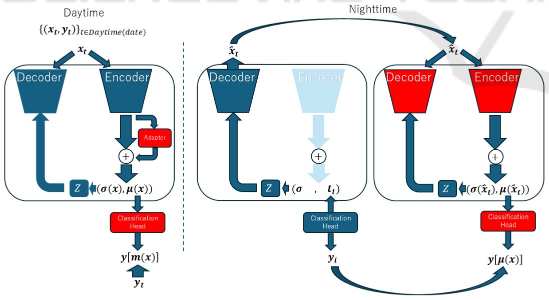

Figure 1: Learning Stages and Flow.

Environments in which learning systems are used, require that known samples be recognized while learning new samples incrementally. When new samples arise, rapid incremental learning is needed to update the recognition system and adapt it to subsequent tasks. Few-shot incremental learning faces the dual challenges of catastrophic forgetting and overfitting. Combining data augmentation with incremental learning can mitigate these risks.

Figure 1 outlines the proposed method, which alternates between two stages: daytime (learning new tasks) and nighttime (reviewing past tasks and reconstructing knowledge). This model combines a VAE with NNs as recognition heads, with the VAE encoder serving as the backbone. During daytime, NNs learn few-shot samples as feature-label pairs from the backbone. The Adapter, introduced by (Houlsby et al., 2019), is attached to the encoder's output to adapt and reconstruct new samples, mitigating catastrophic forgetting without updating the original VAE parameters.

However, at night, copies of the NNs and decoder from the original VAE are used to generate learning samples by producing previously memorized samples, and the VAE is retrained. In this retraining process, contrastive loss was used to train the metric.

In the following sections, we explain the recognition head (NNs and VAE) with the attached Adapter, followed by a detailed description of the two learning periods, daytime and nighttime.

3.2 VAE with an Adapter

The model uses a variational autoencoder (VAE)(Kingma and Welling, 2014) for feature extraction and data expansion in both day and night phases. The VAE, composed of a CNN-based encoder and decoder, outputs a latent variable z from input x, assumed to follow the following normal distribution.

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma^2_\phi(x)) \qquad (1)$$

Here, ϕ denotes the parameter vector of the encoder. By contrast, the decoder reconstructs the input vector from z. That is,

$$p_\theta(x|z) = \mathcal{N}(\mu_\theta(z), \sigma^2_\theta(z)) \qquad (2)$$

Here, θ denotes the parameter vector of the decoder. The parameter vectors are optimized through learning by minimizing the following Kullback-Leibler divergence (KLD) loss function:

$$\begin{aligned}\mathcal{L}(\theta,\phi;x) &= -D_{KL}(q_\phi(z|x)||p(z)) \\ &+ \mathbb{E}q_\phi(z|x)[\log p_\theta(x|z)] \qquad (3)\end{aligned}$$

The VAE originally minimizes Eq 3 using a fixed dataset. During daytime, incremental learning is needed as new samples are provided. Minimizing Eq 3 with only new samples risks catastrophic forgetting, losing prior information.

To address this, an Adapter(Houlsby et al., 2019) is added to the encoder. The adapter, a single linear layer, is inserted as a bypass in the output of the encoder's final layer.

$$\mu_\phi(x) = \mu_\phi(x) + f_{Adapter,\mu}[W^T_{a\mu}\mu_\phi(x) + b_\mu] \qquad (4)$$

$$\sigma_\phi(x) = \sigma_\phi(x) + f_{Adapter,\sigma}[W^T_{a\sigma}\sigma_\phi(x) + b_\sigma] \qquad (5)$$

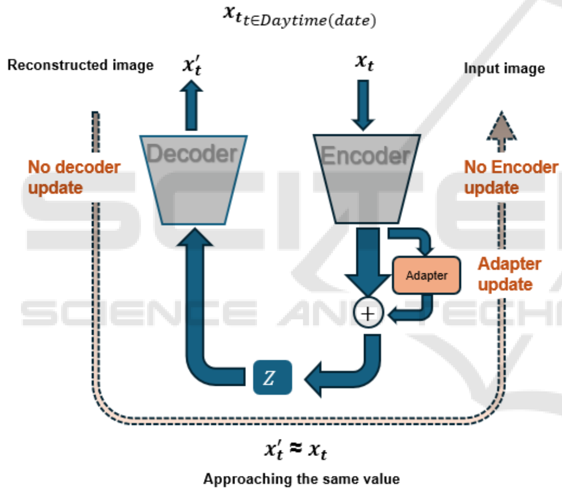

Figure 2: Adapter Update Flow.

Eq 3 is calculated as follows: The VAE mainly learns during nighttime. Let an input image from a set of new data be $x_i \in D_{new}(date)$, the latent variable obtained through the encoder be z_i, and the image reconstructed through the decoder $\mu_\theta(z_i)$ be $x^{'}_i$.

The reconstruction error representing the difference between x_i and $x^{'}_i$ is calculated using the binary cross-entropy(Kingma and Welling, 2014) in Eq 6.

$$\begin{aligned}&L_{\text{recon}}(D_{new}(date)) = \\ &-\sum_{i \in D_{new}(date)} \left(x_i \log x^{'}_i + (1-x_i)\log(1-x^{'}_i)\right) \qquad (6)\end{aligned}$$

Additionally, the KL divergence between latent variable z and its prior distribution is calculated using Eq 7 and included in the loss function.

$$\begin{aligned}&L_{\text{KL}}(D_{new}(date)) \\ &= -\frac{1}{2}\sum_{i \in D_{new}(date)} \left(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2\right) \qquad (7)\end{aligned}$$

where σ_i is the variance of the input data and μ_i is the mean of the input data. From Eqs 6 and 7 , the total loss of the VAE is expressed as shown in Eq 8.

$$\begin{aligned}&L_{\text{total}}(D_{new}(date)) = \\ &L_{\text{recon}}(D_{new}(date)) + L_{\text{KL}}(D_{new}(date)) \qquad (8)\end{aligned}$$

When updating the Adapter, the parameters of the encoder and decoder are fixed, and updating is performed using Eq 9 only for the set of new samples from that day, as shown in Figure 2. $D_{new}(date) \equiv \{(x_t, y_t)\}_{t \in Daytime(date)}$.

$$\Delta\theta_a \propto -\nabla_{\theta_a} L_{\text{total}}(D_{new}(date)), \qquad (9)$$

where $\theta_a = [W_{a\mu}, w_{a\sigma}, b_\mu, b_\sigma]^T$.

In this way, by training only the Adapter layer, the weights of the existing encoder and decoder remain fixed, preserving the information from previously learned tasks, thereby preventing catastrophic forgetting.

3.3 Recognition Head

Nearest neighbors perform both learning and recognition as recognition heads. The NN is composed of feature vectors $\mu_\phi(x_i)$ and variance vectors $\sigma_\phi(x_j)$obtained from the encoder in Eq 4 and corresponding labels y_i. This set is referred to as the support set S.

$$S \equiv \{(\mu_\phi(x_j), \sigma_\phi(x_j), y_j) | j = 1, \cdots, M\} \qquad (10)$$

During recognition, the label vector y_{i^*} of the template vector closest to the mean vector $\mu_\phi(x)$ is output. That is

$$i^* = \arg\max_j \frac{\mu_\phi(x)^T \mu_\phi(x_j)}{||\mu_\phi(x)|| \, ||\mu_\phi(x_j)||} \qquad (11)$$

The $\sigma(x_j)$ will be used for data augmentation as explained in 3.4.1.During daytime learning, the Adapter is configured for the new samples, and the feature and variance outputs are adjusted for the new samples. The adjusted feature outputs are then used to add a new support set to the existing support S. That is,

$$S = S \cup \{(\mu_\phi(x_j), \sigma_\phi(x_j), y_j) | j \in D_{\text{new}}(date)\} \qquad (12)$$

After daytime learning, recognition results are determined by nearest neighbors. Inputs include known and newly learned samples, so cosine similarities of feature outputs are compared, with and without the adapter, to select the most probable labels. During nighttime learning, the VAE and support sets are rebuilt because the encoder's feature outputs change.

$$S_{\text{new}} \equiv \{(\mu_{\phi'}(x_j), y_j) | j \in \chi\} \tag{13}$$

where $\phi^{'}$ denotes the re-constructed VAE during nighttime learning. Therefore, the next day, the learning process is stated after initialization as follows:

$$S = S_{\text{new}}, \;\; \phi = \phi^{'} \tag{14}$$

3.4 Entire Learning Process

Here, the entire learning process is briefly explained. The learning process is divided into two categories: daytime and nighttime learning.. Note that initially, VAE initial learning must be performed to successfully realize data augmentation. Two VAEs are prepared for each learning phase. Therefore, BaseVAE:$(\phi, \theta)]$ is used for daytime learning, and SubVAE:$(\phi^{'}, \theta^{'})$ for nighttime learning.

3.4.1 Daytime Learning

During daytime learning, the BaseVAE learns from the new samples $D_{new}(date) \equiv \{(x_t, y_t) | t \in Daytime(date)\}$.

First, the Adapter learns using Eq 9. Subsequently, NN appends new prototypes using Eq 12.

3.4.2 Nighttime Learning (Sleeping)

During nighttime learning, data augmentation and the training of SubVAE are executed simultaneously. In this process, latent vectors z are generated using the templates in the NN.

$$z \sim \mathcal{N}(\mu_\phi(x_j), \sigma(x_j)^2) \tag{15}$$

The proposed system generates M samples for each template vector.The decoder generates input vector from z as follows. In this paper, let us denote the set of generated input vectors as R.

$$\mathrm{R} = \mathrm{R} \cup (\mu_\theta(z), y_j), \tag{16}$$

The relearning process uses SubVAE$(\phi^{'}, \theta^{'})$, and parameters of SubVAE are initialized using the parameters of BaseVAE

During nighttime learning, the SubVAE attempts to minimize not only the standard loss defined in Eq 3 but also a contrastive loss for the decoder. The contrastive loss is defined by

$$L_{\text{contrast}} = \frac{1}{N} \sum_{(i,j) \in \text{RBuffer}, i \neq j} \Big(y_i \cdot \|\mu_{\phi'}(x_i) - \mu_{\phi'}(x_j)\|^2 + (1 - y_i) \cdot \max(0, m - \|\mu_{\phi'}(x_i) - \mu_{\phi'}(x_j)\|^2) \Big) \tag{17}$$

Therefore, the SubVAE performs learning using the following loss function during nighttime learning.

$$L_{\text{nightime}} = L_{\text{total}} + L_{\text{contrast}}, \tag{18}$$

where, L_{total} is defined by Eq 8.

After learning the SubVAE, the nearest neighbors are rebuilt from scratch. Therefore, the template vectors are generated from the encoder in the SubVAE without its adapter: $\mu_{\phi'}(x_j)$, where $x_j \in$ RBuffer and the label for each template is set to the corresponding label $y_j \in$ R.

4 EXPERIMENTS

This section presents preliminary benchmark results using the MNIST dataset to evaluate the proposed system. Ablation tests were conducted to assess the impact of each technique introduced in the method.

4.1 Initial Setting

This experiment used the MNIST dataset, consisting of grayscale images of handwritten digits (0–9) with 60,000 training and 10,000 test samples, each 28 × 28 pixels. Data were divided for incremental learning: classes [0–4] for the 1st daytime learning and [5–9] for the 2nd. The experiment followed two phases: ”daytime” and ”nighttime.” In daytime learning, the model learns from new data (30 batches of [0–4] for the 1st and [5–9] for the 2nd). During nighttime learning, the VAE reconstructs data by decoding feature representations from both newly learned and previously learned data.

4.2 RESULTS

4.2.1 Entire Behavior of the Proposed Model

Table 2 presents the accuracies after the 1st daytime learning, 1st sleep learning, 2nd daytime learning, and 2nd sleep learning. We can see from the table that accuracy after daytime learning increased following successful sleep learning.

```
Data: BaseVAE:$(\phi,\theta)$, SubVAE:$(\phi^{'},\theta^{'})$,
      Adapter: $\theta_a$, $M$ , $\chi_{\text{init}}$, $D_{\text{new}}$
initialize BaseVAE:$(\phi,\theta)$ by using $L_{\text{total}}(\chi_{\text{init}})$
while True do
    receive new samples
      $D_{new} = \{(x_p, y_p)\}_{p=1}^{n}$
    optimize Adapter: $\theta_a$ to minimize
      $L_{\text{total}}(D_{new})$ with
    freezing BaseVAE:$(\phi,\theta)$.
    Append prototypes to the Nearest
    Neighbors (NNs).
    for $(x_t, y_t) \in D_{new}$ do
        $f_{meant} = \mu_\phi(x)$
        $S = S \cup \{(\mu_\phi(x_t), \sigma_\phi(x_t))\}$
    end
    Expanding $D_{\text{new}}$ and old memories.
    for each $\mu_\phi(x_i), \sigma_\phi(x_i), y_i$ where $i \in S$ do
        for $n = 0$ to $M$ do
            R =
              $\text{R} \cup \{\mu_\theta(z(f_{\mu_\phi}(x_i)), \sigma_\phi(x_i), y_i)\}$
        end
    end
    Initialize new VAE parameters
       $\phi^{'} = \phi$, $\theta^{'} = \theta$
    optimize $\phi^{'}, \theta^{withoutAdapter.}$
      by using $L_{total}(\text{R})$
    Rebuild the nearest neighbors
    $S = \Phi$
    for each $(x_j, y_j) \in R$ do
        $f_{meanj} = (\mu_{\phi'}(x_j), \sigma_{\phi'}(x_j))$
        $S = S \cup \{(f_{meanj}, y_j)\}$
    end
    $\phi = \phi^{'}$, $\theta = \theta^{'}$
end
```

Algorithm 1: Learning Flow Algorithm.

4.2.2 Ablation Test

An ablation study evaluated the proposed method by removing individual components. Removing the VAE Adapter reduced recognition accuracy by 5%, while removing contrastive loss reduced it by 7%. These results highlight the importance of each module for learning performance.

Ablation Test: Effect of Removing the Adapter. The Adapter layer enables the VAE to retain past knowledge while adapting to new data efficiently during daytime, without significant parameter updates. Its purpose is to prevent catastrophic forgetting and support learning new knowledge alongside existing knowledge.

Experimental results show that removing the Adapter layer reduced recognition accuracy by 5% (Table 3). This highlights significant catastrophic forgetting, where previously learned knowledge was rapidly lost. The Adapter improves long-term performance by balancing knowledge retention and updates with new data.

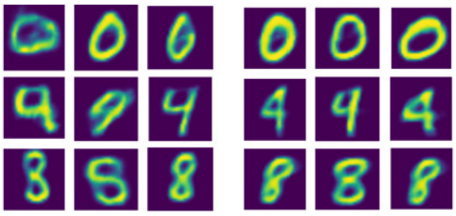

Figure 3: Left: without adapter, Right: with adapter.

Subjective evaluation assessed the visual quality of generated images, focusing on classes [0, 4, 8]. Without the Adapter, results were poor. As shown in Figure 3, images generated with the Adapter were clearer, more stable, and less noisy. This aligns with the decline in label prediction accuracy, highlighting the Adapter's role in improving model learning and image generation.

These findings confirm the Adapter's effectiveness in maintaining flexibility and preventing catastrophic forgetting, crucial for balancing past and new data.

Ablation Test: Effect of Removing the Contrastive Loss. Contrastive loss enhances the separation of data points in latent space, improving interclass discriminability. It brings same-class points closer and separates different-class points by a distance, increasing intraclass density of latent vectors and easing interclass discrimination. Additionally, it improves template quality, boosting nearest-neighbor prototype accuracy.

Removing contrastive loss reduced recognition accuracy by 7% (Table 4), indicating insufficient class separation in latent space and degraded discrimination performance. With contrastive loss, class distances in latent space are maintained, improving classification accuracy.

In conclusion, contrastive loss is essential for enhancing the proposed method's discriminative ability, enabling clear class separation and improved performance.

4.2.3 Sleep Learning Effectiveness

Nighttime learning was introduced to prevent accuracy decline after daytime learning, and its effectiveness was verified. As shown in Table 2 "Day2,"

Table 2: Recognition Accuracy by Class After Daytime and Nighttime Learning.

Class	Day 1 (Daytime)	Day 1 (Nighttime)	Day 2 (Daytime)	Day 2 (Nighttime)
0	99.18	98.88	98.37	89.08
1	99.47	99.47	99.47	92.51
2	82.85	81.20	80.72	67.54
3	89.90	85.64	85.15	74.52
4	89.92	90.22	89.21	67.68
5	0.00	0.00	46.86	77.91
6	0.00	0.00	62.73	91.96
7	0.00	0.00	50.97	81.32
8	0.00	0.00	35.73	84.29
9	0.00	0.00	16.55	78.30

Table 3: Accuracy Comparison Between Proposed Method and Without Adapter after the 2nd sleep learning.

Class	Proposed Method(%)	Without Adapter(%)
0	89.08	77.86
1	92.51	93.30
2	67.54	78.88
3	74.52	74.75
4	67.68	39.51
5	77.91	76.91
6	91.96	92.59
7	81.32	82.49
8	84.29	67.66
9	78.30	70.17
Average Accuracy	80.51	75.41

Table 4: Accuracy Comparison Between Proposed Method and Without Contrastive Loss after the 2nd sleep learning.

Class	Proposed Method(%)	Without Contrastive Loss(%)
0	89.08	80.61
1	92.51	96.04
2	67.54	78.10
3	74.52	76.04
4	67.68	57.23
5	77.91	70.18
6	91.96	68.48
7	81.32	76.26
8	84.29	67.56
9	78.30	63.83
Average Accuracy	80.51	73.43

recognition accuracy for classes 5–9 significantly decreased after daytime learning but improved remarkably after nighttime learning. This suggests that focusing on new data during daytime learning degrades past knowledge, lowering accuracy.

Nighttime learning uses both newly acquired data and reconstructed past data, reinforcing prior knowledge and preventing catastrophic forgetting. The VAE's reconstructed data supplements unseen patterns, balancing past memory retention with new knowledge acquisition.

5 DISCUSSION

Ablation studies verified the contribution of each module in the proposed method to learning performance. Removing elements like the Adapter and contrastive loss demonstrated their quantitative roles in the method's core mechanisms.

The proposed framework introduces a novel perspective by achieving efficient learning with limited data and preventing catastrophic forgetting through daytime and nighttime learning cycles. This highlights its potential for sequential data processing in real-world applications.

In the future, to further broaden the scope of the proposed method, comparative experiments should be conducted with other state-of-the-art methods to confirm its relative superiority. However, at this stage, the significance lies in the fact that the proposed method offers a new direction.

6 CONCLUSIONS

This study proposed a VAE-based method with an Adapter layer and Contrastive Loss to tackle catastrophic forgetting in sequential learning. A two-stage learning process adapts to new data during daytime and integrates knowledge at night, enabling efficient learning while retaining past knowledge.

Ablation studies showed that removing the Adapter and Contrastive Loss reduced accuracy by 5% and 7%, respectively, highlighting their roles in class separation and knowledge retention.

The Adapter, analogous to the hippocampus in the biological brain, plays a crucial role in learning new samples, reflecting the benefits of mimicking biological brain strategies in AI development.

Future work includes applying this method to complex datasets, optimizing hyperparameters, and comparing it with state-of-the-art sequential learning methods to validate its effectiveness.

ACKNOWLEDGMENTS

I am grateful to Mr Naoki Kanemoto for participating in the discussions that contributed to this paper. I also thank the reviewers for their constructive feedback.

REFERENCES

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *TRENDS in Cognitive Sciences*, 3(4):128–135.

Golden, R., Delanois, J. E., Sanda, P., and Bazhenov, M. (2022). Sleep prevents catastrophic forgetting in spiking neural networks by forming a joint synaptic weight representation. *PLoS Comput Biol*, 18(11):e1010628.

Hayes, T. L., Kafle, K., Shrestha, R., Acharya, M., and Kanan, C. (2019). Remind your neural network to prevent catastrophic forgetting. *CoRR*.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning (ICML), 2019*, page 2790–2799.

Hsu, Y.-C., Liu, Y.-C., Ramasamy, A., and Kira, Z. (2018). Re-evaluating continual learning scenarios: A categorization and case for strong baselines. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc.

Kemker, R. and Kanan, C. (2017). Fearnet: Brain-inspired model for incremental learning. *ArXiv*, abs/1711.10563.

Kemker, R. and Kanan, C. (2018). Fearnet: Brain-inspired model for incremental learning. In *International Conference on Learning Representations ICLR2018*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *CoRR*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., and Ramalho, T. (2017). Overcoming catastrophic forgetting in neural networks. *Proceeding of the National Acacemy of United States of America*, 114(13):3521–3526.

Lesort, T. L., Gepperth, A., Stoian, A., and Filliat, D. (2019). Marginal replay vs conditional replay for continual learning. In Tetko, I. V., Kůrková, V., Karpov, P., and Theis, F., editors, *Artificial Neural Networks and Machine Learning – ICANN 2019*, volume 11728 of *Lecture Notes in Computer Science*, pages 466–480, Munich, Germany. Springer.

Liu, X., Wu, C., Menta, M., Herranz, L., Raducanu, B., Bagdanov, A. D., Jui, S., and van de Weijer, J. (2020). Generative feature replay for class-incremental learning. *IEEE*, pages 915–924.

Mallya, A. and Lazebnik, S. (2018). Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7765–7773.

Parisi, G. I., Tani, J., Weber, C., and Wermter, S. (2017). Lifelong learning of human actions with deep neural network. *Neural Networks*, 96:137–149.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *CoRR*.

Shin, H., Lee, J. K., Kim, J., and Kim, J. (2018). Continual learning with deep generative replay. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc.

van de Ven, G. M., Siegelmann, H. T., and Tolias, A. S. (2020). Brain-inspired replay for continual learning-with artificial neural networks. *Nature communications*, 11(4069):1–14.

Yamauchi, K. and Hayami, J. (2007). Incremental learning and model selection for radial basis function network through sleep. *IEICE TRANSACTIONS on Information and Systems*, E90-D(4):722–735.

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017*.