10-16-2023

Investigating Continual Learning Strategies in Neural Networks

Christopher Tam
Western University, ctam@uwo.ca

Luiz Fernando Capretz
University of Western Ontario, lcapretz@uwo.ca

Citation of this paper:
Christopher Tam and Luiz Fernando Capretz, Investigating Continual Learning Strategies in Neural Networks, IECON - 49th Annual Conference of the Industrial Electronics Society, Singapore, pp. 1-7, October 2023. DOI: https://doi.org/10.1109/IECON51785.2023.10311616.

Investigating Continual Learning Strategies in Neural Networks

Christopher Tam
Electrical and Computer Engineering
Western University
London, Canada
ctam86@uwo.ca

Luiz Fernando Capretz
Electrical and Computer Engineering
Western University
London, Canada
lcapretz@uwo.ca

***Abstract*—This paper explores the role of continual learning strategies when neural networks are confronted with learning tasks sequentially. We analyze the stability-plasticity dilemma with three factors in mind: the type of network architecture used, the continual learning scenario defined and the continual learning strategy implemented. Our results show that complementary learning systems and neural volume significantly contribute towards memory retrieval and consolidation in neural networks. Finally, we demonstrate how regularization strategies such as elastic weight consolidation are more well-suited for larger neural networks whereas rehearsal strategies such as gradient episodic memory are better suited for smaller neural networks.**

***Index Terms*—neural networks, continual learning, complementary learning systems**

I. Introduction

Humans have the distinct ability to continually acquire knowledge throughout their lifetime. This ability, aptly referred to as lifelong learning, is enabled by a rich set of neurocognitive mechanisms that allow us to consolidate new information without forgetting previously learnt concepts. This is to say that the process of learning new information does not significantly interfere with our ability to recall old information. In contrast, current machine learning algorithms are unable to process novel streams of data without forgetting previously learned patterns. This is referred to as *catastrophic forgetting*. The problem of artificial intelligence (AI) systems learning over time by accommodating new knowledge and retaining previously learned patterns is referred to as *continual learning*, and has been a long-standing challenge for machine learning and neural networks [1]. This paper investigates continual learning strategies to overcome catastrophic forgetting in neural networks with three factors in mind: the type of network architecture used, the continual learning scenario defined and the continual learning strategy implemented. Specifically, we will be comparing a multi-layer perceptron network to a convolutional neural network architecture in the task-IL, domain-IL and class-IL continual learning scenarios using regularization and memory-based continual learning strategies. The code used to produce this paper can be found here.

II. Background Work

A. Catastrophic Forgetting and Continual Learning

A well known constraint for artificial and biological neural systems is the stability-plasticity dilemma [2]. This dilemma expresses the trade-off between the integration of new knowledge and the stability required to prevent forgetting previously acquired knowledge in a neural system. On one hand, excessive plasticity will result in previously encoded information being overwritten as learning takes place whereas excessive stability will prevent the uptake of new information in the system. Between the two ends of the spectrum, McCloskey and Cohen showed that artificial neural networks (henceforth referred to as neural networks) lean heavily towards plasticity, producing a phenomenon known as catastrophic forgetting [3]. Catastrophic forgetting is defined as the complete or significant forgetting of previously learned information by a neural network trained to learn new information. Richardson and Thomas showed catastrophic forgetting to be present in a variety of neural networks, from standard back-propagation networks to unsupervised self-organizing map networks to connectionist models of sentence acquisition [4]. Catastrophic forgetting frequently occurs when neural networks are trained using a different distribution of data than the distribution the network had previously been trained on. In this case, new data instances differ significantly from previously encountered examples. New information causes the network to partially or completely overwrite the embedded representations from previously learned data, producing a 'catastrophic forgetting' effect of those previously learned patterns. This problem has stood in the way of building lifelong learning systems capable of learning from a continuous stream of information where new information becomes available over time and the number of tasks to be learned is not predefined [5]. In order for continual learning systems to succeed, it is critical that the accommodation of new information should not produce the problem of catastrophic forgetting.

B. Three Continual Learning Scenarios

Continual learning research has garnered plenty of attention and has resulted in a wide variety of experimental protocols being used. This has led to confusion, as some methods are

shown to perform well in certain experimental settings but dramatically fail in others. For example, the elastic weight consolidation algorithm presented by Kirkpatrick et al. [6] claims state-of-the-art performance in it's paper but was reported to show significant performance issues compared to the brain-inspired rehearsal approach of ven de Ven et al. [7]. In order to better compare methods for reducing catastrophic forgetting, this report will use a framework consisting of three distinct continual learning scenarios proposed by Van de Ven and Tolias [8]. This framework focuses on the problem in which a single neural network sequentially learns a series of tasks. Each continual learning scenario is distinguished by a task requirement and the amount of data available at test time. Figure 1 presents the three continual learning scenarios in order of increasing difficulty.

Scenario	*Required at test time*
Task-IL	Solve tasks so far, task-ID provided
Domain-IL	Solve tasks so far, task-ID not provided
Class-IL	Solve tasks so far *and* infer task-ID

Fig. 1. Overview of the three continual learning scenarios proposed by [8].

1) Task-Incremental Learning (Task-IL): In the task-incremental learning scenario, after learning a set of tasks the network is always informed about which of the learned tasks needs to be performed at test time. Given the availability of task identifiers, it is possible to train a network with task-specific components. This enables architectures such as a multi-headed output layer, where the network shares learning resources in the hidden layers but uses task specific output units at test time.

2) Domain-Incremental Learning (Domain-IL): In the domain-incremental learning scenario, the network is not informed about which task needs to be performed at test time. The network needs to perform the proposed task correctly despite not having information about the task identifier, but does not need to correctly infer the task identifier. This scenario is representative of problems where the structure of the tasks is always the same, but the input distribution is changing.

3) Class-Incremental Learning (Class-IL): In the class-incremental learning scenario, the network must be able to solve the learned tasks as well as infer which class the task belongs to at test time. This scenario represents the most difficult problem and reflects the most common real-world problem of incrementally acquiring new knowledge.

C. Strategies for Continual Learning

Continual learning algorithms for neural networks are heavily inspired by our understanding of learning in biological neural systems. In [9], McClelland et al. proposed a theory for complementary learning systems in biological connectionist models which ended up becoming the basis for a computational learning framework for memory consolidation and retrieval. At the heart of this framework is the interplay between episodic and semantic memory which has since provided important insights into the mechanisms of memory consolidation in neural networks. Many learning systems have taken inspiration from this interplay in order to address the problem of catastrophic forgetting in neural networks. We refer the reader to Lesort et al. [10] for a comprehensive overview of continual learning strategies.

1) Regularization Approaches: Several approaches introduce a regularization term into the loss function in order to mitigate the effect of catastrophic forgetting. One way of doing this has been to regularize network parameters during training on each new task to constrain the movement of weights in a way that minimizes the amount of forgetting. This strategy is used in the Elastic Weight Consolidation (EWC) [6] and Synaptic Intelligence [11] algorithms. In both of these methods, estimates for the importance of parameters relevant to previously learned tasks are calculated and parameters are regularized proportional to their relative importance. This has the effect of slowing down learning for parts of the network which are important for previous tasks. Another class of regularization techniques is aimed at preventing activation drift primarily through means of knowledge distillation. One instance of this strategy is the Learning without Forgetting [12] algorithm which computes the output (probabilities) of old tasks for every piece of data in the new task and treats the response as a "pseudo label". The network is then trained again to optimize all the tasks using the pseudo labels generated for the old tasks and the real labels for the new task. The goal here is to prevent the representations of previous data from drifting too far away while learning new tasks.

2) Rehearsal Approaches: Another approach to catastrophic forgetting is to store data from previous tasks. Rehearsal methods keep a small number of "exemplars" or generate synthetic representations of the data previously encountered in order to prevent the forgetting of previous tasks. This approach largely draws on inspiration from the generative role of the hippocampus for the rehearsal of previously encoded experiences. Shin et al. [13] proposed a dual-model architecture consisting of a deep generative model and a task solver. Modelling the rehearsal process of the hippocampus, their architecture sampled training data from previously learned experiences to generate pseudo-data to be interleaved with the data from new tasks. In this way, there was no requirement to explicitly revisit old training samples for experience rehearsal and therefore reduced the cost requirements of working memory. More recently, Lorez-Paz and Ranzato [14] proposed Gradient Episodic Memory (GEM), a rehearsal method which yields positive knowledge transfer to previous tasks. GEM features an episodic memory storing a subset of previously seen examples from a given task. GEM requires far more memory than typical regularization approaches but has produced better results in single pass settings.

III. Task Protocol

The core aim of this paper is to differentiate between three continual learning scenarios and to provide a comprehensive analysis of the performance of the discussed continual learning strategies. We adopted the widely recognized permuted MNIST task protocol as our reference dataset. Permuted MNIST, an altered version of the traditional MNIST dataset, contains 70,000 handwritten digits (0-9), split into 60,000 training and 10,000 testing images. Distinctively, permuted MNIST allows the generation of new tasks by random pixel permutation of each digit. Within the continual learning landscape, tasks are spawned by randomly reordering the 784 pixels of the original 28x28 grey-scale images, which were subsequently tensor-transformed and normalized to a mean of 0.1307 and standard deviation of 0.3081. Thus, every fresh task poses a ten-way classification challenge.

IV. Model Architectures

A multi-layer perceptron (MLP) network and a convolutional neural network (CNN) were used as the network architectures for comparison. For each network architecture, hyperparameters were tuned using a grid-search. The highest average accuracy across all tasks from the grid-search of each network architecture were reported in the final evaluation results.

A. Multi-layer Perceptron Network

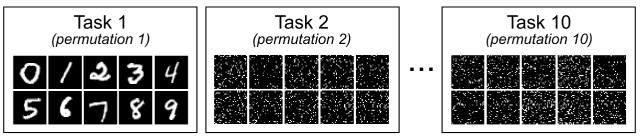
The multi-layer perceptron network consisted of a hidden layer of 512 nodes as the first layer. This was followed by a ReLU non-linearity layer and then by a dropout layer with a probability of 0.5. The network was optimized using stochastic gradient descent and was minimized using the multi-class cross entropy loss function. A summary of the multi-layer perceptron network used is given Table I(a).

B. Convolutional Neural Network

A convolutional neural network with two convolutional layers and two fully connected layers was used. For each convolutional layer, a kernel of size 5 and a ReLU non-linearity was used. The two convolutional layers produced 6 and 16 output channels, respectively. Each convolutional layer was followed by a max pooling layer. In the two fully connected layers, 120 and 105 hidden units were used. The network was optimized using stochastic gradient descent and minimized the multi-class cross entropy loss function. A summary of the convolutional neural network is given in Table I(b).

V. Continual Learning Scenarios

As discussed in Section II-B, we are interested in investigating three continual learning scenarios. For each scenario, both network architectures are evaluated using 3-tasks and 10-tasks. Our goal is to observe how each strategy generalizes to more tasks. In this section we define what each of the continual learning scenarios looks like in the context of permuted MNIST. An overview of the scenarios under permuted MNIST is provided in Figure 2.



Task-IL	Given permutation *X*, which digit?
Domain-IL	With permutation unknown, which digit?
Class-IL	Which digit *and* which permutation?

Fig. 2. Permuted MNIST continual learning scenario proposed by [8].

A. Task-Incremental Learning (Task-IL)

Under Task-IL, task identities are always known and shown at test time. Given an image and a task identity, the network must be able to correctly identify which digit the image represents. Setting up Task-IL involves assigning a progressive task identity to each of the permutations starting from 0. A network is then consecutively trained on all tasks starting from task 0 to task 2 or 9. After training on each task identity is complete, an evaluation loop is run to test the networks ability to recall all previously seen tasks. Since the output of the network was always an integer between 0 and 9 inclusive, 10 units were used as output layer for architectures in the Task-IL scenario.

B. Domain-Incremental Learning (Domain-IL)

In Domain-IL, task identities are irrelevant at test time. Task identities are neither given nor expected to be inferred by the network. Given only an image, the network must be able to correctly identify which digit the image represents. Under the permuted MNIST task protocol, a network will alway predict an integer between 0 and 9 inclusive for this scenario. The challenge is that for each evaluation step, a random bag of digits from all previously seen permutations will be presented. As such 10 units were used as the output layer for architectures under this scenario.

C. Class-Incremental Learning (Class-IL)

In Class-IL, task identities exist but are not provided at test time. Given an image, the network must be able to correctly identify which digit the image represents as well as correctly infer which permutation the image belongs to. This additional requirement prompted an adjustment to the labelling process of input and output classes. For each permutation, class labels were generated starting from the integer where the previous permutation ended. For instance, permutation 1 classes were labelled [0...9], permutations 2 classes were labelled [10...19], etc. At test time, these labels can be viewed as representing the m-th digit (ones column) from the n-th permutation (tens column). As a result, 30 and 100 units were used in the output layer for 3- and 10-task scenarios compared to the previous two scenarios.

TABLE I
MODEL SUMMARIES

(a) MLP			(b) CNN		
Layer (type)	Output shape	Param #	Layer (type)	Output shape	Param #
Linear-1	[-1, 512]	401,920	Conv2d-1	[-1, 6, 24, 24]	156
ReLU-2	[-1, 512]	0	Conv2d-1	[-1, 16, 19, 19]	2,416
Dropout-3	[-1, 512]	0	Linear-3	[-1, 120]	155,640
Linear-4	[-1, 10]	5,130	Linear-4	[-1, 105]	12,705
			Linear-5	[-1, 30]	3,180
Total trainable params		**407,050**			**174,097**

VI. CONTINUAL LEARNING STRATEGIES

A. Naive Strategy

In the naive strategy, a network is sequentially trained on all tasks. With each new task it encounters, the network fine-tunes it's weights to fit the new data. This is where the most amount of catastrophic forgetting occurs and will be used as a lower bound in our comparison.

B. Elastic Weight Consolidation

The regularization approach selected was Elastic Weight Consolidation (EWC) [6]. In this strategy, a regularization term is added to the loss function which penalizes changes to parameters estimated to be important to previously learned tasks. The regularization strength of the loss function is controlled by a hyperparameter λ, where

$$L_{total} = L_{current} + \lambda L_{regularization}$$

We used grid search to find the optimal value of λ.

C. Average Gradient Episodic Memory

For the rehearsal approach, we used the Average Gradient Episodic Memory (AGEM) [15] algorithm. AGEM is similar to the GEM algorithm discussed in Section II-C2 but requires less memory by storing the average gradient vector computed from the individual gradients of task loss for all previously seen tasks at each weight update. This is in contrast to GEM, where each task specific gradient vector has to be stored. The AGEM algorithm was tuned using two hyperparameters which varied the number of patterns to store in memory per experience, and number of patterns from each memory sample to consider when computing the reference gradient. The optimal values of these hyperparameters were found using grid search as well.

VII. MODEL TRAINING AND HYPERPARAMETER TUNING

Each continual learning strategy (naive, regularization and rehearsal) was trained and evaluated on each of the continual learning scenarios (Task-IL, Domain-IL and Class-IL) for each network architecture (MLP and CNN) using 3- and 10-classes. Each configuration trained using 2,000 iterations per task to minimize the multi-class cross entropy loss funtion using the SGD-optimizer with a learning rate of 0.01 and momentum of 0.9. The hyperparameter search for each of the methods is given in Table II.

VIII. RESULTS

A tabular summary of the evaluation results is given in Table III. Our results suggest that complementary learning systems do indeed contribute to improving memory consolidation and retrieval in neural networks. In Figures 3 and 4 we find that continual learning strategies consistently outperform the naive strategy in all experimental conditions. Furthermore, the MLP network outperformed the CNN in all continual learning scenarios and for all strategies. The network architectures had been initialized so that the MLP had just over double the number of trainable parameters of the CNN, as shown in Table I. This suggests that larger networks may have a higher capacity to continually learn than smaller networks. It's possible that MLPs are simply more effective than CNNs for continual learning, but both architectures scored similarly in the 3-Class scenario before departing significantly in the 10-Class scenario. This leads us to believe the CNN is capable of continually learning and that the inability to scale to a higher number of classes is due to the limited number of trainable parameters. Finally, the MLP results show that EWC and AGEM performed similarly in the Task-IL and Domain-IL scenarios but EWC dominated in the Class-IL (hardest) scenario. Similar results were produced in the CNN architecture but in reverse, with AGEM dominating EWC in the most difficult scenario. This suggests that regularization techniques may not be as effective as replay approaches in smaller neural networks with fewer parameters, and indicates a trade-off between the strengths of these two strategies. Since EWC learns new tasks while trying to avoid changes to parameters which are sensitive to previous tasks, it is very likely that this algorithm suffers in networks with fewer parameters to regularize. Because AGEM keeps a memory of each of the previous tasks by storing the gradients computed from the previously seen tasks loss functions, AGEM is designed to be more performant in architectures with fewer internal parameters but with more memory-bandwidth. These results also provide a more granular view of the effectiveness of the two learning algorithms in the context of the continual learning scenario at hand. It may be such that the desired use-case for the neural network falls into the 3-Class Domain-IL scenario, in which case a smaller network such as the CNN may be satisfactory.

TABLE II
HYPERPARAMETER VALUES

EWC		AGEM	
Hyperparameter	Value	Hyperparameter	Value
λ	[0.001, 0.01, 0.1]	Patterns per experience	[10, 30]
		Sample size	[50, 250, 500]

TABLE III
AVERAGE ACCURACY OF EACH CONFIGURATION AFTER COMPLETING ALL TASKS.

MLP							
		Task-IL		Domain-IL		Class-IL	
Approach	Method	3-Class	10-Class	3-Class	10-Class	3-Class	10-Class
Baseline	Naive	93.89%	85.16%	93.80%	91.30%	93.49%	91.30%
Regularization	EWC	97.24%	96.22%	97.22%	96.89%	96.62%	95.26%
Replay	AGEM	98.12%	97.36%	97.51%	97.33%	97.17%	91.15%
CNN							
		Task-IL		Domain-IL		Class-IL	
Approach	Method	3-Class	10-Class	3-Class	10-Class	3-Class	10-Class
Baseline	Naive	55.74%	28.90%	52.92%	31.29%	31.40%	9.38%
Regularization	EWC	89.07%	52.12%	89.74%	52.18%	82.86%	19.90%
Replay	AGEM	98.32%	52.45%	91.81%	56.40%	83.25%	37.18%

IX. DISCUSSION

Convergence in neural networks signifies the network's ability to arrive at a stable solution during the learning phase. It's an essential attribute because a network that fails to converge might not generalize well or might even fail to learn. In the context of our results and the continual learning strategies under study, the following section dives deep into analyzing any convergence-related issues.

A. Variability in Convergence Due to Network Size

An interesting observation arises when comparing the MLP and CNN architectures. The MLP, with its larger parameter set, appears to exhibit faster convergence in many scenarios compared to the CNN. This suggests that larger networks might possess an inherent ability to converge more rapidly, perhaps due to their expansive search space in parameter tuning. However, this could also render them more susceptible to local minima, which may not represent the global optimal solution.

B. Task Complexity and Convergence

As the task's complexity increases from 3-Class to 10-Class scenarios, there's an evident strain on the network's convergence ability. In particular, the CNN, with its limited trainable parameters, struggles significantly in the 10-Class scenario, suggesting that task complexity can heavily influence the rate and stability of convergence.

C. Replay Techniques and Stable Convergence

Replay strategies like AGEM, by design, hold promise in offering a more stable convergence. By storing gradients from previously seen task loss functions, they aim to ensure that the network does not stray too far from its previous learning. However, the potential downside could be a slower convergence rate as the network continuously tries to balance its learning from multiple tasks.

X. CONCLUSION

In this paper we investigate the role of continual learning strategies to overcome catastrophic forgetting in neural networks with three factors in mind: the type of network architecture used, the continual learning scenario defined and the continual learning strategy implemented. Our results show that neural networks with a higher number of trainable parameters are more successful at continually learning than are neural networks with fewer trainable parameters, suggesting that neural volume may play a role in a networks ability to consecutively learn new tasks. However when working with networks with fewer trainable parameters, we see an advantage in using memory-based continual learning strategies which rely on storing representations from previous tasks in memory. As the number of trainable parameters grows, regularization-based strategies begin to dominate indicating a trade-off between the availability of trainable parameters and dependency on external memory. This supports the work proposed by McClelland et al. [9] in the role of complementary learning systems as a basis for improving memory consolidation and retrieval and contributes an analysis of which types of complementary learning systems benefit from certain network configurations.

XI. FUTURE WORK

In Section VIII we raised the possibility that MLP network architectures may be better suited for continual learning than CNN architectures. This work may be extended to investigate the strengths of various continual learning strategies between network architectures by holding the number of trainable

Fig. 3. MLP: Average accuracy computed after training on each task under different learning scenarios.

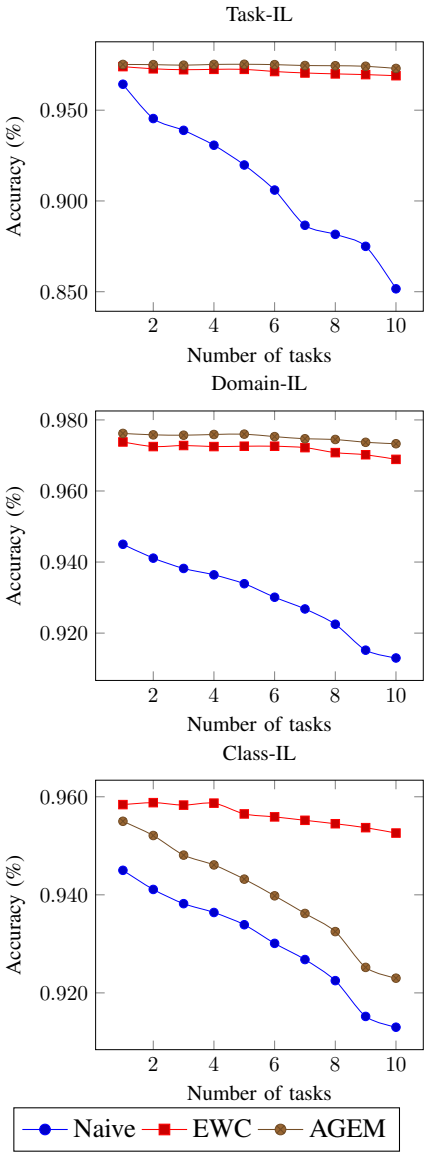
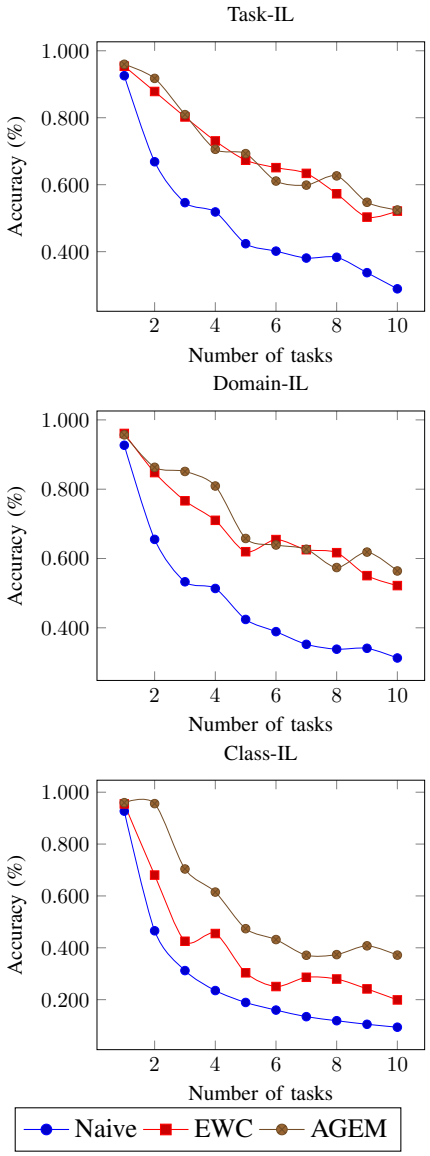


Fig. 4. CNN: Average accuracy computed after training on each task under different learning scenarios.



parameters constant. Similarly, a comparison of the strengths of various continual learning strategies for a given network architecture would benefit applications where the network architecture is pre-determined or fixed. One may also extend this methodology to other classes of continual learning strategies such as generative replay and architecture-based approaches, and other types of network architectures for a more comprehensive survey.

REFERENCES

[1] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired artificial intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, 2017.

[2] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers in psychology*, vol. 4, p. 504, 2013.

[3] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[4] F. M. Richardson and M. S. Thomas, "Critical periods and catastrophic interference effects in the development of self-organizing feature maps," *Developmental science*, vol. 11, no. 3, pp. 371–389, 2008.

[5] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[6] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[7] G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature communications*, vol. 11, no. 1, pp. 1–14, 2020.

[8] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv:1904.07734*, 2019.

[9] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex:

insights from the successes and failures of connectionist models of learning and memory." *Psychological review*, vol. 102, no. 3, p. 419, 1995.

[10] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Information fusion*, vol. 58, pp. 52–68, 2020.

[11] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3987–3995.

[12] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[13] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *arXiv preprint arXiv:1705.08690*, 2017.

[14] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *arXiv preprint arXiv:1706.08840*, 2017.

[15] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," *arXiv preprint arXiv:1812.00420*, 2018.