Continual Learning: Essential for AI Sustenance

Manil Shrestha

Candidacy Document
Department of Computer Science
Drexel University, College of Computing and Informatics
October 2023

Contents

1 Abstract

The rapid advancement of artificial intelligence (AI) and machine learning technologies has given rise to sophisticated models capable of a wide range of tasks. However, most current models are limited by their inability to adapt to new information after initial training, often suffering from "catastrophic forgetting" when exposed to new data or tasks. This poses a significant challenge to the sustained growth and applicability of AI models in dynamic, real-world settings. The field of continual learning emerges as a cornerstone for achieving truly intelligent systems that can adapt over time without requiring a full retraining process. The importance of continual learning is multi-faceted. First, it enables the development of flexible systems that can efficiently navigate dynamic settings. These models can incorporate new learnings without requiring full-scale retraining, thus conserving computational resources and time. Second, in domains where immediate adaptability has life-critical implications, such as healthcare, autonomous driving, and real-time language translation, continual learning is not just advantageous but often mandatory for ensuring safety and efficacy. Lastly, it empowers these systems to provide a customised experience for each user, thereby increasing engagement and usefulness. Ongoing research in this domain aims to improve long-term learning and memory retention, which is instrumental in advancing the capabilities and ensuring the sustenance of future AI systems.

2 Introduction

The 2020s may be heralded as the beginning of a new industrial revolution, driven by Artificial Intelligence (AI). The term "Artificial Intelligence" was first coined by John McCarthy in a proposal for the Dartmouth conference in the summer of 1956. However, the concept of equipping machines with human-like abilities had been broached earlier. In 1945, Vannevar Bush laid the foundational ideas for augmenting human cognition through machines in his paper "As We May Think".[1] Similarly, Alan Turing, who is considered a pioneer in theoretical computer science and AI, had discussed the potential of machines to perform intelligent tasks in a paper he wrote in 1950. [30] [26]

2.1 Neural networks and deep learning through time

In 1944, the foundational concept of neural networks was introduced by Warren McCullough and Walter Pitts, researchers from the University of Chicago. Perceptrons, the simplest form of neural network, gained traction in the realms of psychology and early computer science until 1959, when Marvin Minsky and Seymour Papert released their book, "Perceptrons". This work revealed the limitations of Perceptrons in performing certain types of computations such as solving linearly inseparable problems efficiently.[18][6] After experiencing a revival in the 1980s, interest in neural networks waned in the early 2000s. However, it has made a strong comeback recently, mainly due to the explosion of available data and significant improvements in hardware technologies.

Deep learning, a subfield of machine learning, revolves around neural networks comprising three or more layers. Today's cutting-edge machine learning models predominantly harness deep architectures, spanning applications from smartphone speech recognition to autonomous vehicles. The progression from a few layers in perceptrons to deep neural structures with billions of parameters has been swift. Remarkably, in less than a century, we've arrived at chatbots competent enough to tackle bar exams.[9] LeCun et al., in 1998, published a seminal paper [13] in the field of deep learning. Their proposed LeNet architecture remains one of the initial successful implementations of convolutional neural networks (CNNs). The paper demonstrated the effective deployment of gradient-based learning techniques for complex neural configurations, asserting that backpropagation could train CNNs. This replaced the manual feature extraction inherent in traditional computer vision methods. Essentially, this was a cornerstone, guiding subsequent CNN innovations applied to tasks ranging from image identification to video analytics and even beyond visual assignments.

Fast-forward to 2012: Krizhevsky et al.[12] presented the AlexNet architecture, integrating deep convolutional layers along with novel elements such as the ReLU (Rectified Linear Unit) activation functions, dropout methods, and data augmentation. Remarkably, this model eclipsed contemporaneous top-performing models in the prestigious ImageNet Large Scale Visual Recognition Challenge (ILSVRC). AlexNet's triumph ignited a shift in the machine learning realm, steering research predominantly towards deep learning, with CNNs gaining traction for a broad spectrum of tasks, not just limited to image categorization.

However, CNN's reign was not everlasting. Come 2017, Vaswani et al.[31] unveiled their groundbreaking paper titled "Attention is All You Need", spotlighting the transformer—a fresh neural network design purposed for sequence-to-sequence tasks, bypassing the need for

recurrent or convolutional layers. Outclassing then-leading NLP models in machine translation benchmarks, the transformer's essence soon permeated computer vision, thanks to Dosovitskiy et al in [4]. Their trailblazing paper introduced the Vision Transformer (ViT), likening image segments to NLP "words." Remarkably, when ViTs were pre-trained on expansive datasets and later fine-tuned for specific tasks, they performed outstanding performance on multiple benchmarks, challenging the supremacy of CNNs in computer vision.

2.2 Learning weights through backpropagation

This section gives an overview of how a neural network learns through backpropagation.

2.3 Catastrophic forgetting

When a neural network encounters a new task, the introduction of a different distribution often leads to suboptimal adaptation. The term "catastrophic forgetting" was first coined in a 1989 paper [16] by McCloskey and Cohen. In this paper, they demonstrated experiments on a standard back propagation neural network and showed that sequentially learning new tasks would diminish the performance of the previously learned tasks.
Catastrophic forgetting in neural networks has to do with how the 'learning' happens. The networks modify their weights based on the training data and when introduced to data from a different task, especially if the data distribution differs greatly from the initial training set, the network reconfigures its weights to optimize for the new task, frequently compromising the performance of the initial task. This intrinsic trade-off is termed the plasticity-stability dilemma. High plasticity enables rapid learning of new tasks but may result in forgetting previous tasks. Conversely, increased stability preserves prior knowledge but can impede the learning of new tasks.[17] This topic is an active research area, and various strategies to address it are explored in section 3 below.

3 Continual learning

Continual learning, sometimes referred to as lifelong learning, is a framework in machine learning aimed at enabling models to keep learning over an extended period. The objective is for the model to adjust to new information or tasks without losing its grasp on what it has previously mastered. Within this paradigm, the model encounters a continuous flow of data or a series of different tasks, gaining new skills while maintaining its proficiency in tasks it has already learned.
One prevalent form of continual learning is Class-Incremental Learning (CIL), in which a machine learning model is designed to acquire proficiency in new classes over time without losing performance in previously learned ones. Initially, the model is trained on a fixed set of classes and subsequently exposed to additional classes either one by one or in groups. The main goal is to enable the model to identify these new classes without compromising its performance on the original set of classes. Other approaches to incremental learning include Task-Incremental Learning (TIL) and Domain-Incremental Learning (DIL). Both CIL and TIL are closely related, treating the addition of new classes as new tasks for the model to

learn. However, they differ in the inference stage: CIL calls for the model to classify among all the classes learned over time, while TIL focuses on classifying within a specific 'task.' DIL, on the other hand, is primarily concerned with changes in data distribution, where new tasks involve instances from different data distributions but have the same label space as the original task. [36]
Recently, there has been a surge in research efforts aimed at advancing the field of continual learning. To organize these contributions into distinct categories, I will use the taxonomy outlined in the papers [36] and [32]. These approaches can typically be sorted into three main groups: Data-centric, Model-centric, and Algorithm-centric. However, it's crucial to acknowledge that these groups are not strictly separate; there's often a considerable amount of overlap among the methods.

3.1 Data-centric

Data-centric methods primarily aim to include data from earlier tasks, often referred to as exemplars, to assist the model in retaining past knowledge. These approaches typically utilize the exemplar dataset in one of two ways: either by directly replaying this data during the training of new tasks or by using it to guide the model's optimization trajectory. Multiple strategies are available for curating and maintaining the exemplar set, given that these exemplars should ideally capture the essential features of the classes learned earlier.
Earlier research on data replay, as outlined in a 1990 paper [21], highlights the benefits of rehearsal in backpropagation networks. The concept of directly replaying raw exemplar data during training has shown potential in mitigating catastrophic forgetting. However, the storage of raw data, such as images, poses memory challenges. To address this, there was a shift in focus towards storing lower-dimensional extracted features from the exemplar sets. Research presented in [8] indicates that storing feature descriptors from training images of previously learned classes can yield performance comparable to other state-of-the-art data-centric approaches in CIL. In the paper, the authors suggest modifying the features acquired from previous tasks to match the current feature space, as this space may change when training on new tasks.
In the late 2010s, generative networks such as Generative Adversarial Networks (GANs) began to gain popularity. Inspired by the short-term memory functionality of the hippocampus in primate brains, [25] introduced Deep Generative Replay. This framework features a dual-model architecture that includes a deep generative model, referred to as the "generator," and a task-solving model, known as the "solver." As depicted in Figure 1, the model consists of two main parts: the Generator and the Solver. The Generator, which is based on GANs, is capable of creating 'replays' of data distributions from earlier tasks, aiding in the retention of knowledge when training on new tasks. As the name implies, the Solver handles the task of image classification. For each new sequential task, inputs from both the new tasks and the previous generator are used to train a new generator, ideally preserving knowledge across the sequence of tasks. The paper also integrated the generative replay technique with the knowledge distillation method, Learning without Forgetting (LwF), and showed an improvement in performance.
Generative replay offers several advantages over other methods. Specifically, the network is optimized using a combination of generated past data and real current data. As a result, its

performance is comparable to joint training on an accumulated set of real data, as long as the generator accurately replicates the input distribution. However, a significant limitation of this approach is that the algorithm's effectiveness largely hinges on the quality of the generator.

Beyond simply replaying raw exemplars or producing synthetic rehearsal samples, data regularization emerges as another method that leverages historical data. Gradient Episodic Memory (GEM) [15] utilizes this strategy by integrating the exemplar set into the loss function to direct the model's optimization path. Considering that exemplars are typical instances from earlier classes, GEM adeptly strikes a balance between assimilating information from new classes and preserving knowledge from prior ones. Equation 1 shows the optimization function which was proposed by the authors.

$$\begin{aligned} &\text{minimize} \quad \ell(f_\theta(x,t),y) \\ &\text{subject to} \quad \ell(f_\theta, M_k) \leq \ell(f_\theta^{t-1}, M_k), \quad \text{for all } k < t. \end{aligned} \tag{1}$$

Here, f_θ^{t-1} denotes the model trained on the previous task, and M_k is the exemplar set containing data from previous tasks. The optimization function in equation 1 refines the model with a limitation: the loss derived from the exemplar set should not exceed that of the earlier model.

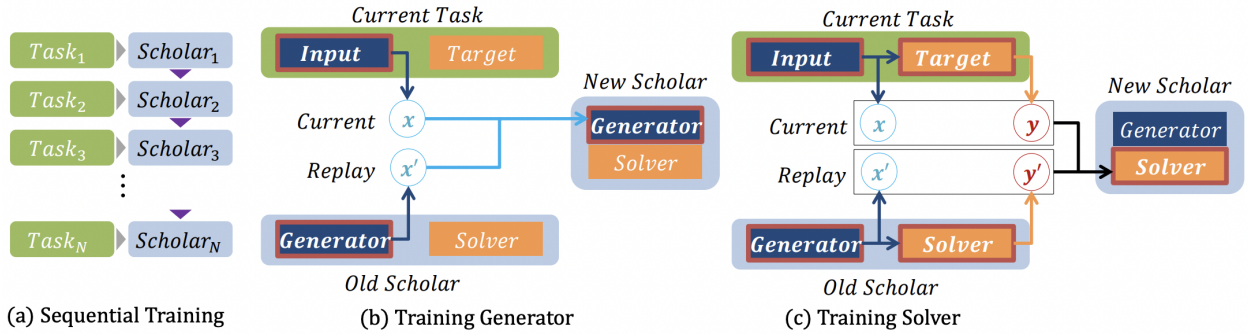

Figure 1: Deep generative replay dual-model architecture from [25]. (a) Sequentially training multiple scholar models is akin to continuously training a single scholar model while referencing its latest version. (b) A new generator is trained to simulate a combined data distribution of real data x and re-used inputs x' from the preceding generator. (c) A new solver is trained using actual input-target pairs (x, y) as well as replayed input-target pairs (x', y'), where the replayed output y' is generated by inputting the replayed inputs into the prior solver.

3.2 Model-centric

Model-centric approaches emphasize adapting the model as it encounters a series of new tasks. One of the early notable methods in this domain is Elastic Weight Consolidation (EWC), introduced in a 2017 paper [10]. EWC is designed to mitigate the problem of catastrophic forgetting by applying a regularization technique inspired by Bayesian inference.

Upon completing the training on an initial task, EWC evaluates the significance of each weight in relation to that task. The underlying principle is that weights crucial for the first task's performance should undergo minimal adjustments when the model trains on subsequent tasks. The Fisher Information Matrix is used to compute the importance of these weights. For every weight, the Fisher Information offers an estimate indicating the sensitivity of the first task's performance to alterations in that weight.
During the training for newer tasks, EWC integrates a regularization component into the loss function. This component imposes penalties on modifications to weights identified as significant for earlier tasks, leveraging the Fisher Information values. The severity of this regularization is determined by a specific hyperparameter. Mathematically, the EWC loss is described in equation 2.

$$L(\theta) = L_T(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta - \theta_i^*)^2 \tag{2}$$

Here, L_T, represents the current task's loss, F_i denotes the Fisher Information associated with the weight θ, θ_i^* is the weight's value post-training on the preceding task, and λ is the hyperparameter dictating the intensity of the regularization. Incorporating this regularization encourages the model to retain knowledge from prior tasks while efficiently learning new ones.

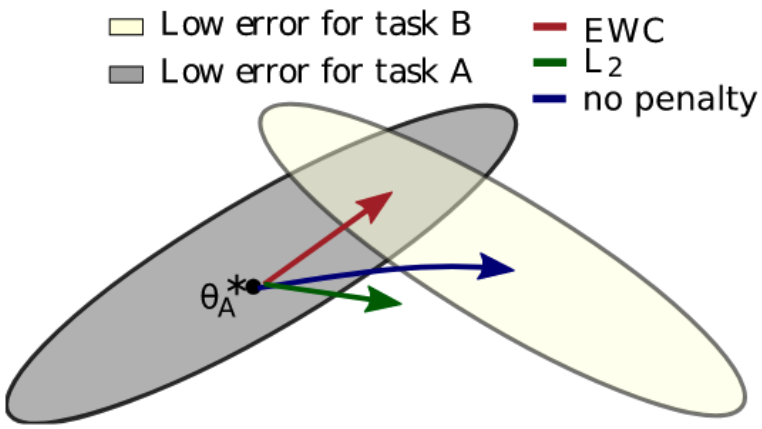

Figure 2: The principle of Elastic Weight Consolidation illustrated in a two-dimensional concept as depicted in [34]. The essence of EWC is to learn new weight, θ^*, that not only caters to new tasks but also minimizes error from tasks that have been previously mastered.

Aside from the parameter regularization technique like EWC, there exists another class of continual learning approaches known as dynamic networks. These networks are crafted to adaptively modify the model's representation by changing the network structure in various ways. Dynamically Expandable Networks (DEN) [35] is designed to adaptively increase their structure. When faced with a new task, DENs have the capability to introduce ad-

ditional neurons to the network, aiding in mastering the new task without compromising prior knowledge. Rather than retraining the whole model when it expands, the network strategically retrains specific segments, optimizing computational efficiency. To ensure the model remains efficient in size, DENs utilize techniques like splitting and pruning. Neurons critical for various tasks may be divided into specialized neurons for each task. Conversely, neurons that don't significantly benefit any particular task might be removed. Additionally, DENs frequently employ a masking strategy where certain neurons or weights are "frozen" or "masked" during the training on subsequent tasks, guaranteeing their preservation and thus upholding knowledge from earlier tasks.
Another notable approach to continual learning using dynamic networks is ProgNN [24]. This method adds new columns or neural networks for each subsequent task, all the while maintaining the existing ones. These newly introduced columns are trained for the current task and are interconnected with previous columns through lateral connections. This design enables the newer columns to tap into the knowledge accumulated from prior tasks. This strategy ensures that the knowledge from earlier tasks remains intact, allowing for the consistent expansion of the model's capabilities with the introduction of new tasks. While ProgNN's strength lies in its preservation of prior knowledge without succumbing to forgetting, it might result in escalating computational demands as the architecture grows with each added task.

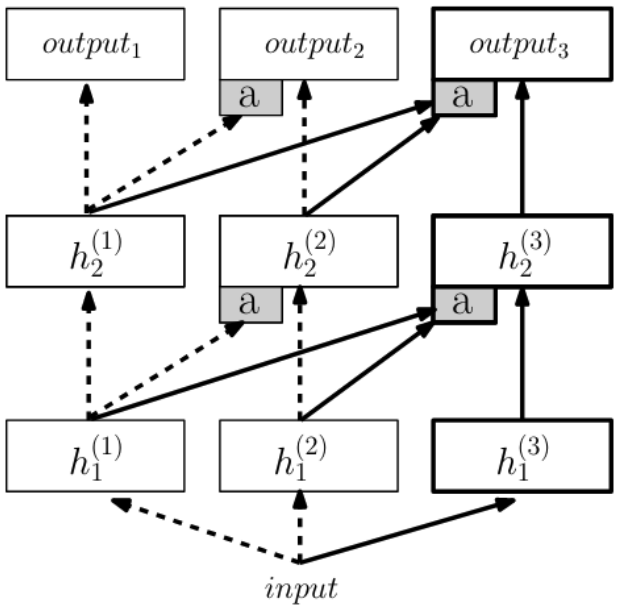

Figure 3: Depiction of three-column ProgNet from [24]. The two leftmost columns (indicated by dashed arrows) were trained on tasks 1 and 2, respectively. The gray box labelled a signifies the adapter layers. A third column is introduced for the last task, which can tap into the features learned from all preceding tasks.

Recently, a method called MEMO [37] has been introduced to tackle the memory challenges in Continual Incremental Learning (CIL). MEMO, which stands for Memory-efficient

Expandable MOdel, aims to enhance model expansion while minimizing memory budget constraints. This approach advocates for training several backbones consecutively and then combining their outputs to form the final feature representation for prediction. A notable observation in continual learning is that the initial layers across various models tend to be alike, while the deeper layers exhibit more diversity. This implies that the initial layers have broader applicability, while the deeper layers are tailored to specific tasks. As a result, expanding the initial layers is not the most memory-efficient strategy for CIL. To address this, MEMO suggests splitting the backbone at intermediate layers. In this setup, the specialized block is analogous to the network's deeper layers, whereas the generalized block represents the shallower layers.

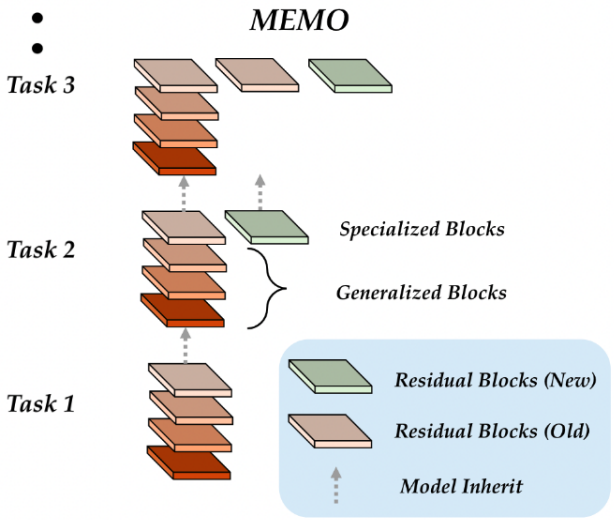

Figure 4: Diagram from [36]. MEMO [37] segments the network architecture, focusing solely on the expansion of specialized blocks which exhibits more diversity. The shallower layers tend to represent more common features that are shared amongst the the different classes.

3.3 Algorithm-centric

Algorithm-centric methods focus on designing algorithms to maintain the model's knowledge in former tasks. Knowledge Distillation (KD) is a type of algorithm-centric methods in continual learning which enables the knowledge transfer from a teacher model to the student model, with which we can teach the new model not to forget. Learning without Forgetting (LwF) [14] is a notable KD method where the objective is to build the regularization term to reduce forgetting on the previous task. In the LwF approach, a neural network undergoes training on an initial task. When presented with a subsequent task, the model doesn't start from scratch. Instead, it gets fine-tuned on this new data, ensuring it not only learns the new task but also maintains proficiency in previous tasks. This balance is achieved by blending the objectives of both the new and old tasks, with the old task's objective being extracted from the teacher model's predictions. LwF leverages the knowledge distillation

loss, as introduced by Hinton et al. [7], which effectively guides one network's outputs to mimic another's.

$$L = \ell(f(x), y) + \sum_{k=1}^{|\mathcal{Y}_{b-1}|} -S_k(f^{b-1}(x)) \log S_k(f(x)) \tag{3}$$

Equation 3 gives the objective function which was presented in the LwF paper. Here, f represents the model being trained on a new task, with y as the ground truth label, and Sk acting as the regularization hyperparameter. Here the second loss term is KD loss, where the objective of this term is to ensure student model to not drift too far away from the teacher's predictions.

The older (teacher) model, represented as f^{b-1} has parameters that remain frozen during learning for new tasks, while Y_{b-1} encompasses all the previously seen classes. When an input x is introduced, the output probability for the k^{th} class indicates its semantic affinity to that class, ensuring a consistent semantic relationship between the old and new models.

LwF method does not hold past data, the only way it performs knowledge distillation is by using the old model as teacher passing down knowledge into the newer student model. Incremental Classifier and Representation Learning (iCARL) [20], extends the LwF technique by maintaining a set of exemplars for each class to assist the model in retaining knowledge of past classes as it learns new ones. This is a hybrid method that has comprises aspects of both data-centric and algorithm-centric approaches. A distinct feature of iCARL is its approach during the inference phase. Instead of classifying using the network's final layer, which is standard practice, iCARL employs a nearest-mean-of-exemplars classifier. The average feature vector for the exemplars of each class is calculated. When classifying a test image, its feature vector is matched against these average vectors, and the class is determined based on the nearest mean.

The exemplars in iCARL are meticulously picked to be highly representative of their respective classes. The selection aims to minimize the mean feature vector's distance to all the training samples of that class. One widely used strategy of picking exemplars is known as Herding [33]. In this method, given an instance set $X = \{x_1, x_2, ..., x_n\}$ belonging to class y, it initially calculates the class centre using the current embedding function $\phi(.)$. It then computes and ranks the distance between each instance and this class centre, denoted by $||\mu y - \phi(x_i)||$, in ascending order. The assumption is that the class centre is often the most representative aspect of each class, choosing exemplars close to this centre also boosts their representational power.[36]

Beyond knowledge distillation methods such as LwF and iCARL, another notable model-based strategy in continual learning is model rectification, with Bias Correction (BiC) being a prominent technique [34]. BiC's main objective is to prevent the model from forgetting or becoming disproportionately biased against prior classes as new ones are introduced. Initially, the model undergoes training on the new classes without restricting the final classification layer, utilizing exemplars of earlier classes in conjunction with the new class data. After training, it becomes evident that the logits (pre-activation function outputs) for prior classes might exhibit bias, often due to the imbalance between new and old class data, with the latter typically drawn from a limited exemplar set. To address this, a bias correction value is determined using a validation set comprising both old and new class samples. These

correction values are then applied to the final classification layer's weights, reducing any bias towards the newer classes and ensuring the model's predictions remain well-balanced across all class categories.

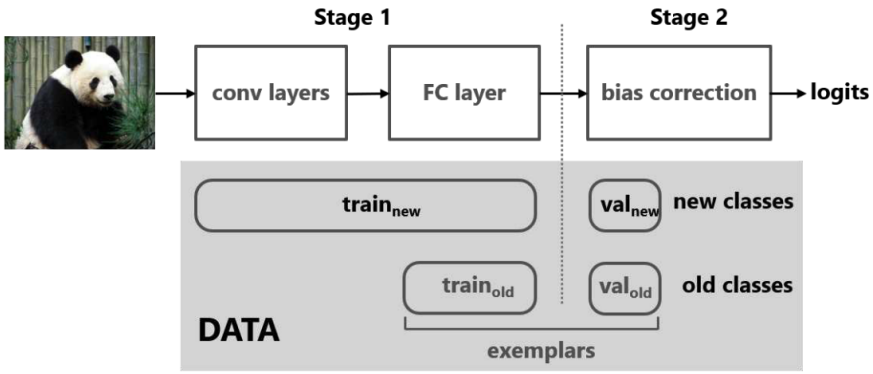

Figure 5: Overview of BiC method from [34]. The samples from the previous classes, known as exemplars, and the new class samples are divided into training and validation groups. The training group is utilized to refine the convolutional layers and the fully connected (FC) layer. Meanwhile, the validation group serves the purpose of bias adjustment.

4 Dataset, experiment setup and results

This section will explore the prevalent experimental setup in continual learning field and highlight the most commonly used datasets. Additionally, we'll report the performance of some of the methods previously mentioned to gauge their effectiveness in tackling catastrophic forgetting.

Datasets: CIFAR-100 [11] is a dataset comprising 60,000 images, each of 32x32 pixels, spread across 100 classes. It's divided into 50,000 training images and 10,000 test images. ImageNet1000 [23], a widely recognized dataset in computer vision, includes over 1.2 million training images and 50,000 test samples spanning 1,000 classes. Unlike CIFAR-100, the image sizes in ImageNet1000 vary. ImageNet100 is a subset of the ImageNet1000, featuring 100 classes randomly chosen from the original dataset.

Data split: To emulate a continual learning environment, datasets are typically divided into incremental stages. A common approach is to evenly distribute all the classes across each incremental phase. For instance, if there are N classes and X stages, each incremental task would involve $\frac{N}{X}$ classes. Another prevalent method is to initially split the total classes in half. The model is trained on half of these classes first. The remaining classes are then distributed over subsequent tasks. This means the initial task covers N/2 classes, and each of the subsequent X incremental tasks takes on $\frac{N}{2(X-1)}$ classes. To harmonize these methods, [36] introduced a unified splitting method termed 'Base-m, Inc-n'. In this notation, 'm' represents the number of classes in the initial phase, while 'n' indicates the number of classes in every subsequent incremental task. An 'm' value of 0 corresponds to the 'Train from scratch' protocol.

Evaluation Metrics: Given that performance figures from the paper [36] will be referenced

Table 1: Average and last top-1 accuracy performance comparison on CIFAR100

Method	Base0 Inc5		Base0 Inc10		Base0 Inc20		Base50 Inc10	
	$\bar{A}$	A_B	$\bar{A}$	A_B	$\bar{A}$	A_B	$\bar{A}$	A_B
Finetune	17.59	4.83	26.25	9.09	37.90	17.07	22.79	9.09
EWC	18.42	5.58	29.73	12.44	39.19	19.87	25.77	11.47
LwF	30.93	12.60	43.56	23.25	48.96	30.00	41.12	25.06
Replay	58.20	38.69	59.31	41.01	60.03	43.08	52.37	41.26
iCaRL	63.51	45.12	64.42	49.52	67.00	54.23	61.29	52.04
BiC	62.38	43.08	65.08	50.79	67.03	56.22	61.01	49.19
MEMO	**68.10**	**54.23**	**70.20**	**58.49**	**70.43**	**61.39**	**69.39**	**62.83**

Table 2: Average and last top-1 accuracy performance comparison on ImageNet100

Method	Base0 Inc5		Base0 Inc10		Base0 Inc20		Base50 Inc10	
	$\bar{A}$	A_B	$\bar{A}$	A_B	$\bar{A}$	A_B	$\bar{A}$	A_B
Finetune	17.06	4.70	26.19	9.30	40.20	17.86	24.12	9.26
EWC	18.78	6.14	27.78	11.10	41.54	18.98	26.21	11.54
LwF	41.76	17.74	55.50	33.10	68.43	53.00	46.24	31.42
Replay	56.37	37.32	59.21	41.00	64.53	48.76	55.73	43.38
iCaRL	62.36	44.10	67.11	50.98	73.57	61.50	62.56	53.68
BiC	58.03	34.56	65.13	42.40	76.29	66.92	66.36	49.90
MEMO	**68.19**	**56.10**	**71.00**	**60.96**	**76.59**	**68.64**	**76.66**	**70.22**

in this section, I've also adopted the evaluation metrics presented therein. The paper denotes Top-1 accuracy following the b^{th} task as A_b, with a greater A_b value signifying superior prediction accuracy. As the CIL model undergoes continuous updates, its accuracy tends to decline as more tasks are added. Therefore, the accuracy after the final task (A_B) serves as an appropriate metric to gauge the cumulative accuracy across all classes. Another measure called 'average accuracy' takes into account the outcomes at each incremental phase. Since, the measure solely focusing on the end accuracy overlooks the model's performance progression throughout its learning journey, the inclusion of $\bar{A}$ is crucial to understand the issue of catastrophic forgetting in each increment.

The tables 1 and 2 shows the performance of the 7 different class incremental continual learning methods. Fine-tuning is the baseline model. It is often referred to as the foundational method in class-incremental learning because it primarily focuses on learning new concepts from the current task. The other methods whose performance have been reported below are Exemplar Replay, EWC, LwF, iCaRL, BiC and MEMO.

5 Discussion

The current research directions show significant promise, but these techniques also have limitations that need to be addressed. For example, data-centric methods primarily focus

on mitigating forgetting by leveraging past data. Both data replay and regularization techniques operate under the stringent assumption that the exemplar or pseudo-rehearsal data accurately represent the distribution of the older data, which may not always be true. Moreover, retaining historical data may not be suitable in situations where privacy is a significant concern. Model-centric techniques, such as EWC, operate on the assumption that weights critical for one task maintain their significance in later tasks. However, this is a strong assumption that might not always hold true. Given that most model-centric methods apply regularization to model parameters, there's a considerable likelihood of performance levelling off. Specifically, after learning several tasks, there might be a stagnation or even a drop in performance due to the cumulative effects of regularization penalties. Algorithm-centric approaches such as knowledge distillation shift knowledge from a teacher model, educated on prior tasks, to a student model tackling a new task. This approach has several challenges. First, it requires managing both the teacher and student models. This can result in the loss of intricate details. It also adds computational burdens. The method is sensitive to hyperparameters and heavily depends on the quality of the teacher model. Finally, there are potential data privacy concerns to consider when KD also depends on the exemplar sets. While this method assists in preserving knowledge, it also brings about complications in its implementation.

All the techniques discussed in this document share a common feature: they seek to modify the base model. In some cases, they adjust the weights with a focus on regularizing for preceding tasks. In others, they incorporate distillation loss to facilitate the transfer of knowledge from the teacher model to the subsequent student model. Some model-based methods append layers and/or neurons to keep catastrophic forgetting at the minimum. Furthermore, these methods grapple with the fundamental tension between retaining old knowledge and accommodating new information. Many strategies aim to find a delicate balance, ensuring that while new knowledge is integrated, the integrity of previously learned information remains intact. This act of balancing can lead to increased computational demands, heightened storage needs, and extended training times. Certain methods may require more memory to retain exemplars or save model milestones, while others might call for more complex training procedures or algorithmic tweaks.

6 Future research direction

In the era of expansive models trained on multimodal data spanning millions, the crux of learning has shifted. These models have largely mastered the art of feature extraction, delving into the nuances of the input data. With billions of parameters at their disposal, they efficiently encapsulate the intricacies inherent in the data. The challenge, however, lies not in feature extraction but in deciphering and continually adapting to the associations of these features. Drawing parallels with human development, as described by [27], a child's visual clarity aligns with an adult's by the age of six months. Beyond this point, the child's learning curve pivots towards discerning the semantic essence of their visual experiences. This developmental trajectory mirrors the challenges in continual learning within computer vision. For instance, while a robust ViT model excels in extracting both standard and intricate features, the ongoing challenge is discerning and continually refining the interrelationships

within this rich latent feature space.The emergence of expansive contrastive multi-modal models like CLIP [19] has spurred interest in harnessing the textual embedding phase used by text-to-image models. Studies like [2] and [29] illustrate that these textual embedding realms possess the capacity to encapsulate fundamental image semantics.

6.1 CLIP: Multimodal approach to image-text semantic

Contrastive Language-Image Pretraining (CLIP) [19] is a zero-shot multi-modal model, learns insights directly from raw text descriptions of images. Using multi-modal learning, and natural language supervision, CLIP adeptly grasps various visual concepts. The model is trained to recognize a variety of visual concepts in images, linking them to corresponding textual descriptions. CLIP undergoes training using the WebImage Text (WIT) [28] dataset, comprising 400 million diverse image-text pairs sourced from the internet. At its core, CLIP concurrently trains a text encoder and an image encoder to determine which text segments align with which images from the dataset. By using a contrastive objective, it binds text to images, positioning both in the same embedding space. The versatility of the CLIP model is evident in its varied applications, from zero-shot image classification and fine-tuned image classification to image captioning. Recently, an interesting application of CLIP can be seen in the field of stable diffusion.[22]
Introduced in 2022, Stable Diffusion is a text-to-image deep learning model grounded in diffusion methodologies. Primarily designed to produce intricate images from text prompts, it also offers capabilities like inpainting, outpainting, and facilitating image-to-image translations steered by textual cues. Stable Diffusion employs iterative denoising of random noise, guided by the pretrained CLIP text encoder and attention mechanisms, to generate images representing trained concepts.

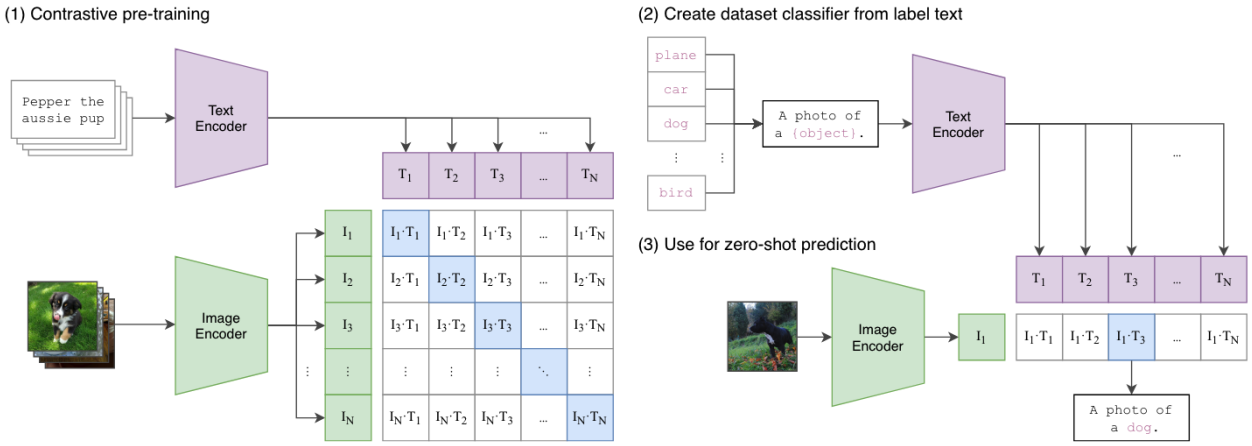

Figure 6: CLIP concurrently trains both an image encoder and a text encoder to accurately predict the matching pairs from a set of (image, text) training samples. During testing, the trained text encoder crafts a zero-shot linear classifier by embedding the names or detailed descriptions of the desired dataset's classes. Figure from [19].

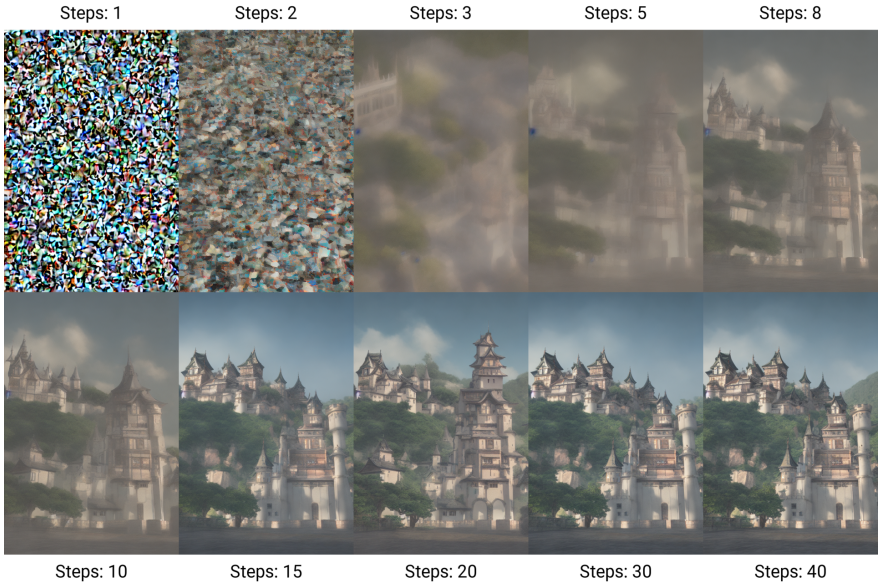

Figure 7: AI-generated artworks showcasing a European-inspired castle located in Japan, produced using the Stable Diffusion V1-5 AI diffusion technique. Figure from [3].

Figure 8: (Left) The process of textual inversion identifies new pseudo-words within the embedding space of a pre-trained text-to-image model, representing distinct concepts. (Right) These pseudo-words can be assembled into novel sentences, positioning our subjects in fresh settings, altering their style or layout, or incorporating them into innovative products. Image taken from [5]

6.2 Textual Inversion: Learning one concept at a time

Textual Inversion [5] is a method designed to extract new ideas from a limited set of sample images. Initially it was demonstrated with a latent diffusion model and now it has

been adapted for use with other models, including Stable Diffusion. By understanding new "words" within the text encoder's embedding space, it facilitates custom image generation using text prompts. In the context of Stable Diffusion, this new concept manifests as a token embedding, typically a vector of size 512 or 768, located in the embedding space of CLIP's textual encoder. Textual Inversion aligns with the critiques of the prevailing continual learning approaches, as outlined in section 5. Throughout the Textual Inversion procedure, the model's weights remain unchanged, preserving the integrity of the pre-trained models without any modifications that might affect previously acquired knowledge. The UNet, Variational Autoencoder (VAE), and CLIP models are all frozen as the optimal embeddings for the novel concept are sought. I believe that the approach showcased by Textual Inversion, where embeddings are actively searched for, represents the future trajectory for continual learning research. Search for the token embedding will be equivalent to discerning and continually refining the interrelationships within the rich latent feature space of CLIP's transformer-based visual and text encoders.

References

[1] Vannevar Bush et al. As we may think. *The atlantic monthly*, 176(1):101–108, 1945.

[2] Niv Cohen, Rinon Gal, Eli A Meirom, Gal Chechik, and Yuval Atzmon. "this is my unicorn, fluffy": Personalizing frozen vision-language representations. In *European Conference on Computer Vision*, pages 558–577. Springer, 2022.

[3] Wikipedia contributors. Stable diffusion — wikipedia the free encyclopedia, 2023. Online; accessed October 10, 2023.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[5] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[6] Larry Hardesty. Explained: neural networks. *MIT News*, 14, 2017.

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

[8] Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. Memory-efficient incremental learning through feature adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 699–715. Springer, 2020.

[9] Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. Gpt-4 passes the bar exam. *Available at SSRN 4389233*, 2023.

[10] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[15] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[16] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[17] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013.

[18] Jan Mycielski. Marvin minsky and seymour papert, perceptrons, an introduction to computational geometry. 1972.

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[20] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[21] Anthony Robins. Catastrophic forgetting in neural networks: the role of rehearsal mechanisms. In *Proceedings 1993 The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, pages 65–68. IEEE, 1993.

[22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[24] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[25] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

[26] Chris Smith, Brian McGuire, Ting Huang, and Gary Yang. The history of artificial intelligence. *University of Washington*, 27, 2006.

[27] Samuel Sokol. Measurement of infant visual acuity from pattern reversal evoked potentials. *Vision research*, 18(1):33–39, 1978.

[28] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2443–2449, 2021.

[29] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

[30] Alan M Turing. Computing machinery and intelligence. mind, lix (236), 433–460, 1950.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023.

[33] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1121–1128, 2009.

[34] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 374–382, 2019.

[35] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

[36] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Deep class-incremental learning: A survey. *arXiv preprint arXiv:2302.03648*, 2023.

[37] Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. *arXiv preprint arXiv:2205.13218*, 2022.