Addressing Catastrophic Forgetting: Paving the Path to Overcoming Memory Loss

A continual learning survey

Leonardo Lavalle
1838492
Sapienza, University of Rome
lavalle.1838492@studenti.uniroma1.it

1 Introduction

AI systems reached by now impressive results, demonstrating in some cases to even surpass human-level performance. Think about object recognition, speech recognition or the ability of playing games where for example DeepMind's AlphaGo defeated world champion Go players and later AlphaZero achieved superhuman performance also in chess and shogi. But although their achievements are remarkable, they accomplished individual tasks using "static" models which lack the ability to adapt their behavior over time.

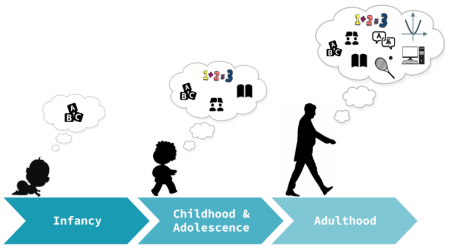

Figure 1: Timeline of human continual learning process.

One of the major aims of AI, in particular strong AI or Artificial General Intelligence (AGI), is emulating human intelligence. *How do we learn?* We have the ability to learn incrementally and to retain the previous acquired knowledge. If during elementary school I learned to write, read and do math operations, I don't lose this baggage of knowledge as soon as I learn other things. With continual learning we indeed try to address the challenge to continually acquire new knowledge while retaining previously learned information (see Figure 1). If we want to aim for the highest level of AI, where the systems exhibit consciousness and self-awareness, continual learning is a necessary feature that needs to be acquired and mastered. This calls for systems that adapt continually and keep on learning over time.

Neural Networks doesn't learn in this manner, they requires restarting the training process each time new data or new tasks become available. They actually suffer from the so called **catastrophic forgetting**: performance on a task learned previously should not experience substantial degradation over time when additional tasks are introduced. This phenomena is a result of a more general problem in neural networks, that is the *stability-plasticity* dilemma. Stability is the network's capability of retaining previous knowledge, while plasticity is referred to the ability of integrating new knowledge. Every continual learning approaches strive for avoiding, by any means, catastrophic forgetting.

2 Approaches

Lately continual learning received more attention and several different approaches have been used to tackle the problem. We will go through some papers more in details either because of performance achievements or simply for historic relevance. In literature there are many ways of categorize continual learning methods and the way taxonomies are structured are various. I won't follow the exact characterization shown during class seminar on the argument (taken from [14]) or the one in [4], because too much detailed and intrigued. Starting from [2] categorization, which it offers a more general and intuitive overview of the problem, we can distinguish continual learning methods in three groups: (i) *replay-based* where previous task samples are replayed while learning a new task to alleviate forgetting; (ii) *regularization-based* in which an additional regularization term is added into the loss function, facilitating the consolidation of previous knowledge while learning from new data; (iii) *parameters isolation* methods that dedicates different model parameters to each task (weights are not shared among tasks). It's almost impossible cate-

gorize and put into a family methods every existing continual learning approach, but in my opinion these three families were the most appropriate to better introduce the topic and let understand which are the possible ways to deal with it.

2.1 Replay-based

This family of methods aims to replay past data during training to address catastrophic forgetting issue. We can talk about *experience replay* when we store a replay buffer that keeps track of past training samples and about *generative replay* when generative models (such as GANs, VAEs etc.) are used to generate synthetic data that resembles previously seen data. Regarding the first type of methods, we can do a further distinction: *reharsal* methods where we explicitly retrain on a limited subset of stored samples (**iCaRL** [11]) and *constrained optimization* ones in which, by imposing constraints, we ensure that updates related to new tasks do not disrupt the performance of previous ones (**GEM** [8]). We now go into the details of these two papers.

iCaRL. The paper is from 2017 and is about a class incremental learner for image classification. The key idea behind *iCaRL* is to maintain a set of "exemplars" for each class, which are selected from the training data and stored in a memory buffer. These exemplars are used to represent the "knowledge" of each class. It's a two-step process: (i) classification step: a K-means algorithm is employed to select a subset of representative exemplars (the closest ones to the feature mean of each class) to be used to finetune the model; (ii) distillation step: the previously learned knowledge is distilled into a "teacher" network, which is then used to guide the learning of the updated "student" network. Some results taken from [11] can be seen in Figure 2. At inference time the method uses a *nearest-mean-of-exemplars* classification strategy, meaning that after calculating the mean values for each class, we determine the predicted output class by selecting the nearest mean to the input image.

GEM. Another interesting and popular approach in the field of continual learning is the Gradient Episodic Memory method [8]. Written in 2022, it always exploits the replaying of observed examples from previous tasks, but it approaches to the problem of forgetting in a different manner. The training procedure basically aims to solve a constrained optimization problem, projecting the current task gradient in a feasible area (first order Taylor series

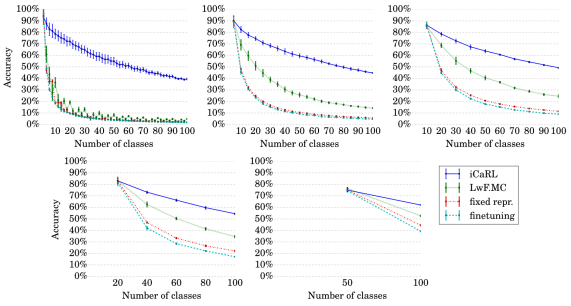

Figure 2: Multi-class accuracy on iCIFAR-100 with 2 (top left), 5 (top middle), 10 (top right), 20 (bottom left) and 50 (bottom right) classes per batch. The *finetuning* model is simple network which does not take any measures to prevent catastrophic forgetting.

approximation involved), outlined by the previous task gradients. We want to limit the impact of new task updates on those previously obtained. With respect to iCaRL, GEM does not leverage structured task descriptors (like exemplars means), which may be exploited to obtain positive forward transfer (zero-shot learning). Moreover, it doesn't leverage any type of advanced memory management, that in many replay-based approaches is something necessary. In Figure 3 is possible to observe how GEM outperforms other continual learning methods like EWC [5] (it also spends less computational power w.r.t. it according to the paper) and iCaRL. Of course this plot is made by the authors of the paper and I think to deeply understand which approaches performs better in which scenarios a more accurate and fair analysis should be done (please refer to Section 3).

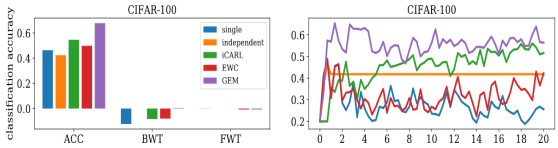

Figure 3: On the right there's the evolution of the test accuracy at the first task, as more tasks are learned (on CIFAR-100 dataset). The *single* model refers to a single predictor trained across all tasks.

One major limitation of replay-based methods is their scalability constraint when the number of classes increases, necessitating additional storage of input samples. That's why other directions have been explored, like the family of regularization-based methods.

2.2 Regularization-based

The intention of this line of works is alleviating memory requirements. They instead want to consolidating previous acquired knowledge by adding an extra regularization term in the loss functions. Simple in words, but less so in deeds. The first way we can add regularization is through teacher-student paradigm with knowledge distillation. This strategy has been applied by Learning without Forgetting (LwF) paper [7], by using the previous model output as soft labels for previous tasks. More interesting in my opinion are the approaches in which during training of later tasks, changes to important parameters are penalized. To be more clear: we fight the forgetting by penalizing the variation of network's parameters based on their importance for previous tasks. Elastic weight consolidation (**EWC**) [5] was the first to establish this approach. The idea has a biological inspiration: in brains, synaptic consolidation enables continual learning by reducing the plasticity of synapses that are vital to previously learned tasks. The implemented algorithm performs a similar operation by constraining important model parameters to stay close to their old values. Without going into the mathematical details, the additional regularization term is derived based on the Fisher information matrix, which measures the sensitivity of the network's loss to changes in the weights.

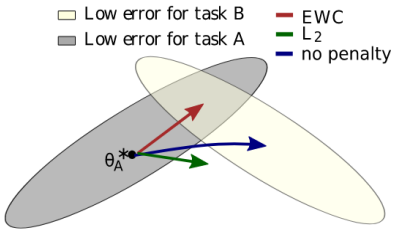

Figure 4: Training trajectories illustrated in a schematic parameter space, with parameter regions leading to good performance on task A and on task B.

We can see in Figure 4 how EWC weights update ensures task A to be remembered whilst training on task B. If we take gradient steps according to task B alone (blue arrow), we will minimize the loss of task B but destroy what we have learnt for task A. The authors of the paper tested the method on several supervised learning tasks in sequence (mainly image classification), showing how EWC mitigates catastrophic forgetting. They compared traditional dropout regularization to EWC (see Figure 5). They found out that stochastic gradient descent with dropout regularization alone is limited, and that it does not scale to more tasks as EWC does.

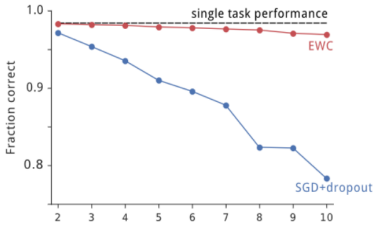

Figure 5: Results on MNIST dataset. Average performance across all tasks using EWC (red) and SGD with dropout regularization (blue).

2.3 Parameter isolation

This set of methods change completely the paradigm: each task assigns distinct model parameters. This means there isn't a unique shared set of parameters across all tasks. I personally think this is the way to go and indeed many research groups studying the argument are pursuing this direction (e.g. Google DeepMind with [12]). There are several ways to face continual learning with this third paradigm. For example one can grow new branches for new tasks and let the model(s) grows dynamically. In Progressive Networks [12] a new neural network is instantiated for each task being solved and knowledge transfer between tasks is enabled via lateral connections to features of previously learned models. Instead, in Expert Gate [1], a similar approach is leveraged. We build a Network of Experts, where a new expert model is added whenever a new task arrives and, since we can only load a relatively small number of models at a time due to limited memory of GPUs, a gating mechanism (an auto-encoder gate) is learnt to decide which expert to activate at test time. Both these two approaches are immune to catastrophic forgetting by design.

Alternatively, we can let the architecture remain static, with fixed parts allocated to each task. *How to do that?* Through masking. As it happens in **PackNet** [9], where previous task parameters of the network are masked out during new task training. The paper was inspired by network pruning techniques, which exploits redundancies in large deep networks to free up parameters that can then be employed to learn new tasks. This was a really clever intuition that brought the method to be

able to sequentially "pack" multiple tasks into a single network maintaining a negligible decrease in performance and minimizing additional storage requirements. PackNet iteratively assigns parameter subsets to consecutive tasks by constituting binary masks. For this purpose, new tasks establish two training phases: a pruning (weight-based strategy) step and a retraining step. This process is performed repeatedly when multiple tasks are added, as illustrated in Figure 6.

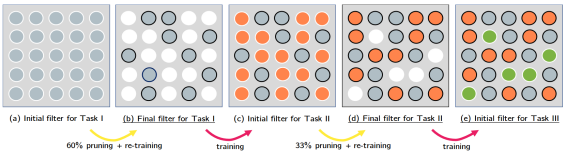

Figure 6: Qualitative illustration of the evolution of a 5×5 filter with steps of training.

In this way the computed *pruning masks* preserve task performance as it ensures to fix the task parameter subset for future tasks. They retain the useful informations for always letting the model behave the same way for each different task. The authors conducted experiments on various multi-task learning benchmarks, including image classification. In particular (see Figure 7) they tested the approach on a single ImageNet-trained VGG-16 network by adding to it three other tasks (CUBS birds, Stanford Cars and Oxford Flowers). Looking at the results, immediately stands out how the accuracies achieved by PackNet are very close to those of separately trained networks for each individual task (Individual Networks), pointing out how the method is robust to catastrophic forgetting (especially considering the complexity of the additional tasks).

Dataset	Classifier Only	LwF	Pruning (ours)		Individual Networks
			0.50, 0.75, 0.75	0.75, 0.75, 0.75	
ImageNet	28.42 (9.61)	39.23 (16.94)	29.33 (9.99)	30.87 (10.93)	28.42 (9.61)
CUBS	36.76	30.42	25.72	24.95	22.57
Stanford Cars	56.42	22.97	18.08	15.75	13.97
Flowers	20.50	15.21	10.09	9.75	8.65
# Models (Size)	1 (562 MB)	1 (562 MB)	1 (595 MB)	1 (595 MB)	4 (2,173 MB)

Figure 7: Table of top-1 errors of four image classification tasks. *Classifier Only* means that a single model is used for all the tasks without doing anything for avoiding catastrophic forgetting and *LwF* refers to [7].

It is worth emphasizing how relatively small is the size of the model using the approach. I believe that an overall view of the most popular SOTA methods for continual learning has been made, but in the following, I want to talk about a recent out-of-the-box approach which levarage Explainable AI to defeat catastrophic forgetting.

2.4 XAI for Continual Learning

Explainable AI (XAI) has gained increasing attention in recent years as the need to comprehend the decision-making process of these "black box" neural networks has become more evident. This urgency is particularly pronounced in applications with significant implications for human well-being, such as cancer prediction for doctors or policy recommendations for policymakers. In such cases, it is crucial not only to ascertain the network's prediction but also to understand the underlying reasoning behind a specific decision.

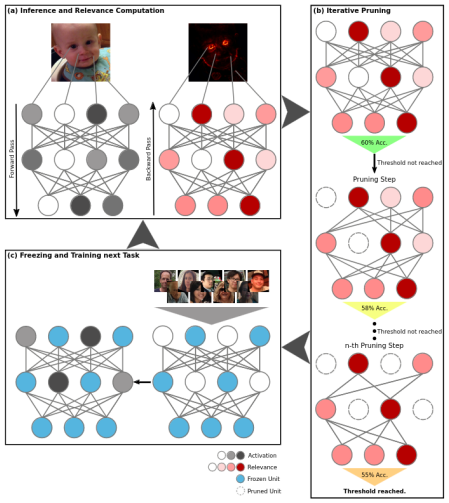

Figure 8: Illustration of our method to identify relevant neurons and freeze their training progress after training on a task. Similar idea to the illustrated process in Figure 6.

Recently, XAI techniques have been proposed to identify the most crucial elements of a neural network. One such method is Layer-wise Relevance Propagation (LRP) [6], which assigns relevance scores to latent network structures by recursively propagating their values. The relevance scores are assigned based on the contribution of each neuron in the network to the final prediction. By decomposing the prediction into the relevance scores of individual input features, LRP provides a way to interpret the model's decision and understand which neurons played an important role. Lately, this type of information has been used with great success to effectively reduce neural network complexity without sacrificing performance. It's a sort of a "more intelligent pruning" and, if one thinks about it, the basic intuition behind [3] is practically the same of

[9]. In "Explain to Not Forget" [3], it's proposed a novel approach (Relevance-based Neural Freezing) that builds upon the aforementioned pruning technique (LRP). Instead of actually "prune" the network, the idea is to freeze the neurons that are relevant for a specific task, while letting the remaining ones free to learn further tasks (see the overall process depicted in Figure 8 to better understand). Like PackNet approach, this method not only successfully retains the knowledge of old tasks within the neural networks, but it achieves this objective with greater resource efficiency compared to other SOTA solutions.

3 Catastrophic Forgetting Analysis

In order to implement models that are increasingly immune to *forget*, we should first undertake an in-depth analysis about the causes of the catastrophic forgetting phenomenon. Many recent works are following this research direction, tackling the problem from different perspectives. In [13] the authors analysed forgetting from the data characteristic point of view. They discovered that some examples within a task are more likely to being forgotten, while others are consistently unforgettable, regardless the particular architecture or training setup. Observing that the forgetting statistics remained relatively consistent across different training settings, they also concluded that forgetting is prevalent due to the underlying data distribution. Furthermore, and this is really interesting, the unforgettable examples seem to play little part in the final performance of the classifier as they can be removed from the training set without hurting generalization (the more an example is forgotten during training, the more useful it may be to the classification task). In another paper [10] they found out deeper layers are the primary source of forgetting and that many SOTA continual learning methods, including [5] and [11], mitigate catastrophic forgetting just by stabilizing deeper representations.

Other noteworthy results come from [2]. Here the authors, focusing on task incremental classification, conducted a comparative analysis of 11 continual learning SOTA methods because of the lack of a comprehensive and up to date experimental comparison. As we can see in all the plots/tables present in this survey (Figures 5, 3, 2 and 7), the proposed method always outperforms alternative approaches and indeed the aim of the paper was to establish a fair comparison among the SOTA approaches. Overall, PackNet seems the best performer by high margin in all the datasets involved in the experiments. Accompanied by iCaRL, which, if larger memory size is used, also shows competitive performance. But the most interesting part of the paper is the study of the influence and the effect of (i) *model capacity*, (ii) *regularization* and (iii) the *order* in which the tasks are presented. (i) After testing on SMALL, BASE, WIDE and DEEP models, they conclude that most methods don't exhibit a particular preference for any of them. It's clear that, in line with [10] findings, increasing model capacity with DEEP models leads to lower performances. (ii) Weight decay and dropout, especially the latter, show consistent merits only for specific methods. Regarding replay-based methods, the influence of regularization seems instead limited. (iii) Presenting the tasks in different orderings to continual learners has insignificant performance effects: they are order agnostic.

4 Conclusions

Continual learning is a long-standing goal of machine learning, where agents not only learn (and remember) a series of tasks experienced in sequence, but also have the ability to transfer knowledge from previous tasks to improve convergence speed. The importance of the argument is proven by the huge quantity of research papers production. My intention was to: firstly provide a glimpse into the broader landscape of the field by showcasing and explaining a selection of the most widely recognized and interesting papers; secondly give an overview of the main challenges of continual learning, with a specific emphasis on catastrophic forgetting and highlight the inherent complexities involved in comprehending the "why" and "when" of forgetting, as well as the strategies employed to mitigate its effects.

References

[1] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. 2017. Expert gate: Lifelong learning with a network of experts.

[2] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

[3] Sami Ede, Serop Baghdadlian, Leander Weber, An Nguyen, Dario Zanca, Wojciech Samek, and Sebastian Lapuschkin. 2022. Explain to not forget: Defending against catastrophic forgetting with xai.

[4] Sebastian Farquhar and Yarin Gal. 2019. Towards robust evaluations of continual learning.

[5] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

[6] Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10:e0130140.

[7] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting.

[8] David Lopez-Paz and Marc'Aurelio Ranzato. 2022. Gradient episodic memory for continual learning.

[9] Arun Mallya and Svetlana Lazebnik. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning.

[10] Vinay V. Ramasesh, Ethan Dyer, and Maithra Raghu. 2020. Anatomy of catastrophic forgetting: Hidden representations and task semantics.

[11] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. icarl: Incremental classifier and representation learning.

[12] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2022. Progressive neural networks.

[13] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning.

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