BrainWash: A Poisoning Attack to Forget in Continual Learning

Ali Abbasi
Vanderbilt University
ali.abbasi@vanderbilt.edu

Parsa Nooralinejad
University of California, Davis
pnoorali@ucdavis.edu

Hamed Pirsiavash
University of California, Davis
hpirsiav@ucdavis.edu

Soheil Kolouri
Vanderbilt University
soheil.kolouri@vanderbilt.edu

Abstract

Continual learning has gained substantial attention within the deep learning community, offering promising solutions to the challenging problem of sequential learning. Yet, a largely unexplored facet of this paradigm is its susceptibility to adversarial attacks, especially with the aim of inducing forgetting. In this paper, we introduce "BrainWash," a novel data poisoning method tailored to impose forgetting on a continual learner. By adding the BrainWash noise to a variety of baselines, we demonstrate how a trained continual learner can be induced to forget its previously learned tasks catastrophically, even when using these continual learning baselines. An important feature of our approach is that the attacker requires no access to previous tasks' data and is armed merely with the model's current parameters and the data belonging to the most recent task. Our extensive experiments highlight the efficacy of BrainWash, showcasing degradation in performance across various regularization and memory replay-based continual learning methods. Our code is available here: https://github.com/mint-vu/Brainwash

1. Introduction

In real-world scenarios, data distributions are inherently non-stationary, constantly evolving and shifting in unpredictable ways. Such variability poses a significant challenge to machine learning and computer vision, where model generalizability assumes stationary training and testing/deployment distributions. Continual Learning (CL) [13, 37, 61] has emerged as a prolific research domain focusing on efficient learning from an ongoing stream of data or tasks. CL primarily seeks to: 1) enhance backward knowledge transfer, which aims to maintain or improve performance on previously learned tasks, thereby mitigating catastrophic forgetting, and 2) bolster forward knowledge transfer, where learning a current task can boost performance on or reduce the learning time for future tasks. CL has significantly progressed in computer vision tasks, including incremental image recognition [33, 59]. With the increase in the adoption of CL algorithms, examining their vulnerabilities is imperative to inform the development of more robust CL methodologies.

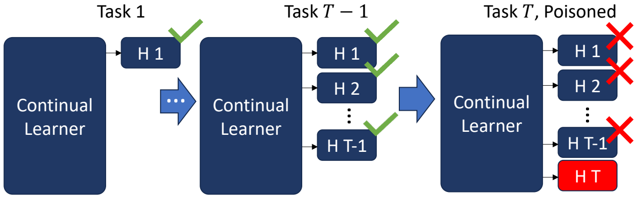

Figure 1. BrainWash is a poisoning attack targeting continual learning systems. It sabotages a task so that, upon learning it, the system's rate of forgetting previously learned tasks is increased.

Most research in CL has focused on overcoming catastrophic forgetting. Existing methods can be categorized into three groups: 1) memory replay, 2) regularization, and 3) parameter isolation methods. Nonetheless, there has been limited focus on the robustness of CL approaches against various types of adversarial attacks. Recent studies have begun to address this gap by proposing backdoor attacks [32, 57] and certain poisoning attacks [27, 39] within the CL context. These contributions are critical in profiling the vulnerabilities of CL methods, paving the way for developing more resilient CL algorithms. Additionally, these findings have implications for closely related and emerging fields such as machine unlearning [3, 9].

Recent works show that an adversary can insert misinformation into a task to distort a continual learner's performance. For instance, Umer et al. [57] show that backdoors can be placed into a task to hijack the performance of a CL method, and the backdoor remains effective even when new tasks are learned. Here, we pose a fundamental question: Is it possible to 'brainwash' a continual learner by poisoning its current task in such a way that performance on all pre-

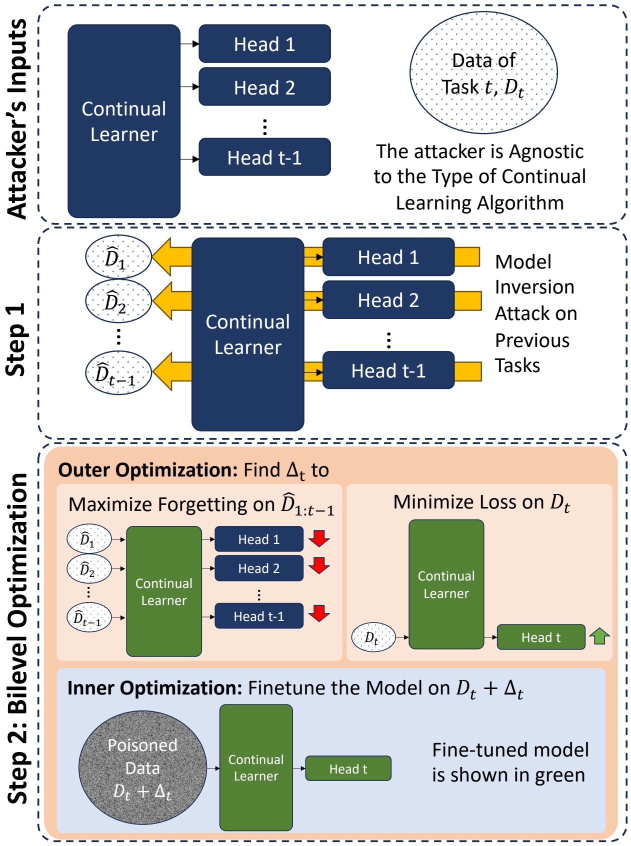
Attacker's Inputs
Continual Learner
Head 1
Head 2
Head t-1
Data of Task t, D_t
The attacker is Agnostic to the Type of Continual Learning Algorithm
Step 1
$\widehat{D}_1$
$\widehat{D}_2$
$\widehat{D}_{t-1}$
Continual Learner
Head 1
Head 2
Head t-1
Model Inversion Attack on Previous Tasks
Step 2: Bilevel Optimization
Outer Optimization: Find Δ_t to
Maximize Forgetting on $\widehat{D}_{1:t-1}$
Minimize Loss on D_t
Continual Learner
Head t
D_t
Inner Optimization: Finetune the Model on $D_t + \Delta_t$
Poisoned Data $D_t + \Delta_t$
Fine-tuned model is shown in green

Figure 2. In our proposed threat model, the attacker gains access to the CL model and the data for the forthcoming task but remains unaware of the data from preceding tasks and the specific CL method employed by the victim (top panel). The attack methodology unfolds in two steps. Firstly, the attacker executes a model inversion attack on the CL model to reconstruct an approximation of the victim's data from earlier tasks (middle panel). Secondly, the attacker employs bi-level optimization to contaminate the data for the current task. This is done in such a way that performance on the reconstructed data from previous tasks is significantly degraded.

vious tasks is significantly degraded? More succinctly, can a task be designed to induce maximum forgetting of prior knowledge in a CL context? We affirmatively answer this question and demonstrate its validity across a wide range of regularization-based CL methods, assuming minimal and realistic conditions. This concept is depicted in Figure 1.

Recent advancements in foundational models have led to the creation of massive models with billions of parameters. These models require significant data resources, yet their training is limited by computational power, restricting repeated passes over the data. Additionally, data isn't sampled in an independently and identically distributed (i.i.d.) manner, necessitating continual learning to integrate new data without forgetting existing knowledge [25]. This forgetting vulnerability could be exploited by adversaries introducing manipulated training data to erase key information.

This paper examines a realistic threat model targeting regularization and memory-based CL methods. Under this model, the attacker gains access to the victim's current model and aims to manipulate the victim's next task. Crucially, the attacker remains unaware of the specific CL algorithm employed by the victim to learn tasks and lacks access to data from prior tasks. We propose a novel method denoted as "BrainWash" that allows for poisoning the current task data to maximize forgetting on prior tasks.

In short, BrainWash consists of two main steps. First, we perform a model inversion attack [24, 60] on the continual learner to approximate the data from the previous tasks. Second, to poison the current task, we construct a bi-level optimization problem such that: 1) the performance on inverted data of previous tasks is minimized, and 2) the performance on the clean data of the current task is maximized. Figure 2 demonstrates the threat model and the two steps.

Contributions. Our main contributions in this paper are:

1. Devising a novel poisoning attack algorithm for regularization-based continual learning methods, denoted as BrainWash.
2. Demonstrating the effectiveness of BrainWash on benchmark CL datasets and across diverse regularization-based CL algorithms.
3. Providing extensive ablation studies to deepen our understanding of BrainWash.

2. Related Work

Continual Learning is a subfield of ML focused on learning from nonstationary streams of data or tasks [13, 37]. Its objectives include improving backward knowledge transfer to maintain or enhance performance on previously learned tasks helping to prevent catastrophic forgetting. It also aims to strengthen forward knowledge transfer, where mastering a current task can improve performance or decrease learning time for future tasks. Catastrophic forgetting prevention is a central goal in this field. To tackle catastrophic forgetting, strategies in continual learning are typically grouped into three main categories: 1) memory-based methods, 2) regularization-based methods, and 3) architectural methods. Memory-based methods involve techniques such as memory rehearsal or replay, generative replay, and gradient projection [2, 18, 19, 40, 49, 50, 52, 56, 58, 59, 66]. These methods often rely on storing and revisiting previous learning experiences or artificially generating them to reinforce learning. Regularization-based methods apply penalties on changing parameters that are vital for tasks already learned [1, 5, 34, 35, 63, 69]. These approaches help in preserving the knowledge acquired from previous tasks while allowing new learning. Architectural methods focus on modifying the learning model itself. Strategies include expanding the model structure [51, 53], isolating parameters specific to certain tasks [43, 44], and using masking techniques [7, 47, 67] to manage the learning process for different tasks. In this paper, we focus on regularization-based methods, mainly due to their effective balance between plasticity and stability, allowing for the integration of new knowledge while preserving essential information from past learning experiences. We propose a data poisoning attack that maximizes forgetting for regularization-based continual learners.

Data Poisoning is a training phase attack on a ML model in which the attacker deliberately alters the victim's training data maliciously [4, 8, 20, 26, 31, 55, 70]. After the victim trains their model using this compromised data, the model would serve the attacker's detrimental objectives, such as significantly reducing the model's test accuracy on all or specific classes (i.e., targeted vs. non-targeted attacks).

Data poisoning is formally defined as a bi-level optimization problem [6, 8]. In the outer level optimization, the attacker optimizes the poisoning, which can be additive noise [26], patch-based noise [12], or a conditional generative model for noise [21], to enforce their malicious intention on the 'resulting network' parameters. This 'resulting network' itself is the solution to the inner optimization problem that minimizes the training objective as would be done by the victim. When the ML model is a deep neural network, this bi-level optimization problem is generally intractable, as it requires backpropagation through the entire SGD training procedure [46]. Hence, the existing literature often approximates this bi-level optimization using various strategies, including first-order approximation methods [29] and more sophisticated methods based on alternating optimization [21]. Similar to [22, 29], our poisoning attack also uses a first-order approximation method for solving the induced bi-level optimization. In contrast with [29], however, our bi-level optimization objective is maximizing forgetting in a continual learner.

Model Inversion [24, 48, 60] encompasses attack strategies designed to either reconstruct training data or deduce sensitive attributes from a trained model. These strategies are broadly divided into 'optimization-based' and 'training-based' methods. Our study primarily explores optimization-based methods, which are widely adopted in the literature [48, 68]. These methods primarily adjust inputs in the data space to maximally stimulate specific output neurons, such as target classes. However, a key challenge arises from the many-to-one mapping characteristic of deep neural networks, where a variety of inputs can lead to the same output. To address this, the literature introduces various forms of priors or regularization terms, making this optimization process more tractable. Such regularization terms range from simpler approaches like Total Variation and image norm [42, 45] to more advanced techniques involving feature statistics [68] and the use of generative models [64]. In this paper, we adopt a model inversion approach similar to Yin et al. [68] to approximate the data that the continual learner has been trained on from previous tasks.

3. Threat model

We consider a victim using a regularization or memory-based CL method to learn a series of tasks. For example, imagine a home robot that continuously learns from its environment to adapt to a new home [10]. The attacker's objective is to poison the training data of the latest task (like learning about a new room), causing the CL model to forget previously learned tasks upon acquiring new information. Furthermore, the attacker poisons the data in our setup by engineering norm-constrained additive noise. We examine two scenarios for an attack: 1) the 'reckless threat model' where the victim deploys the model without monitoring its performance, allowing the attacker to maximize forgetting of prior tasks without regard for current task performance; and 2) the 'cautious threat model' where the victim monitors the model's performance on a potentially poisoned task, necessitating the attacker to balance inducing forgetting while maintaining acceptable current task accuracy, making it a more challenging scenario. In both settings, we assume that the attacker does not have access to the continual learner's training data from previous tasks.

4. Method

In this work, we aim to design a poisoning attack for regularization and memory-based multi-head CL approaches that brainwashes the model, causing it to forget its previous tasks. We assume the attacker has full access to the model and data from the latest task the continual learner will encounter. However, the attacker does not have access to continual learner's data from the previous tasks.

We propose to utilize model inversion attacks [24, 68] to obtain an approximation for the continual learner's data from prior tasks. Using the victim's model, the inverted data from previous tasks, and the data for the current task, the attacker formalizes the poisoning problem through a bi-level optimization and then solves it via a first-order approximation method. In what follows, we briefly review our notations and then describe 1) the model inversion attack, 2) poisoning as a bi-level optimization problem, and 3) our proposed first-order approximation solver.

4.1. Notations

We denote the training data for task $t \in \{1, \cdots, T\}$ as $D_t = \{(x_t^i, y_t^i)\}_{i=1}^{N_t} \subset \mathcal{X} \times \mathcal{Y}_t$, where $x_t^i \in \mathcal{X}$ denotes the i'th sample from the t'th task (e.g., an input image) and $y_t^i \in \mathcal{Y}_t = \{1, \cdots, K_t\}$ denotes its corresponding label with K_t and N_t denoting the number of classes and examples for task t respectively. Let $f(\cdot; \theta)$ denote the CL's backbone that extracts deep representations from the input data, where θ indicates the backbone's parameters, and let $h_t(\cdot; \psi_t)$ denote the classification head for task t, with ψ_t representing its parameters.