Machine Unlearning Challenge

Yiping Li
yipingl4@illinois.edu

Abstract

This study reevaluates the objective of machine unlearning to address specific challenges and proposes three unlearning algorithms tested on CIFAR10 with a ResNet18 backbone, including retraining, gradient ascent, and data modifications. Results show that the gradient ascent method is highly effective, significantly impacting unlearning. The study suggests that reversing established mappings is more effective for unlearning than creating new connections. This approach preserves the network's integrity and its generalization capability. The study also plans to investigate unlearning effects at different network layers, offering deeper insights into machine learning model structures and their adaptability.

1. Introduction

In an era characterized by rapid advancements in artificial intelligence, various neural networks have established themselves as universal learning machines with the remarkable ability to assimilate diverse knowledge from a wide array of inputs. Nowadays, the deployment of neural networks for tasks such as image classification has become increasingly accessible. Nevertheless, an intriguing question emerges: Can we make neural networks unlearn undesired tasks or samples they have previously acquired?

Indeed, in machine learning, many implementations not only possess the capability to unlearn but also suffer this unintended unlearning. That phenomenon is called catastrophic forgetting, and it has long been a substantial impediment to the development of robust and adaptive AI systems for a long time. Traditional machine learning models tend to overwrite previously acquired knowledge with new data, hindering their ability to retain, generalize, and build upon past experiences.

Amid this challenge in machine learning, the exploration of machine unlearning holds promise. Suppose we could endow machines with the capability to selectively unlearn specific knowledge without detriment to their performance in other tasks. In doing so, we might uncover the underlying essence of what truly matters within the architecture of machine learning models. This approach could potentially allow us to mitigate catastrophic forgetting by safeguarding the core knowledge essential for the system's overall functionality. Furthermore, gaining a deeper understanding of the essential nature of learning machines could potentially revolutionize the trajectories of research in other machine learning domains, including but not limited to federated learning, transfer learning, and meta-learning.

In this project, we are going to participate in the machine unlearning competition held by Google on Kaggle. Specifically, an age predictor, which has been trained on face images is given, and our goal is to adjust the model so that it unlearns the designated face images in forget set. The unlearning algorithm is evaluated by comparing its performance gap with the new model trained without forget set.

2. Method

2.1. Defining Machine Unlearning

It is worth to mention that the scope and focus of this research diverge from the parameters set by the organizers of the Machine Unlearning Challenge 2023, leading to a distinct definition and evaluation metric for machine unlearning. This deviation primarily is due to the following two factors: 1) the different objectives of this project compared to the original challenge and 2) the intent to employ a more straightforward, intuitive evaluation metric. The Machine Unlearning Challenge 2023 concentrates on addressing privacy issues related to membership inference attacks, which can potentially reveal if an individual's data was utilized in training a machine learning model. In contrast, the present study seeks to delve into the nature of the knowledge acquired by the model during its training phase, particularly focusing on the phenomenon of catastrophic learning.

Consequently, the evaluation metric adopted by the challenge, rooted in Example-level Differential Privacy, appears overly complicated and somewhat elusive for the

purposes of this research, which aims to gain a deeper understanding of the model's learning mechanics. To align the evaluation process more closely with the objectives of this study, a simplified version of the original metric has been proposed, ensuring it is both relevant and more easily comprehensible in the context of exploring the model's learning dynamics.

Now, let's define the objective for this project. Let D be some given dataset and f be some learning algorithm that takes dataset and model as input. Suppose $D_{tr} \subset D$ is the train set, $D_{te} \subset D$ is the test set, and θ be the model, the fine-tuned model is defined as $\theta_{ft} = f(\theta, D_{tr})$. Now, split the training set into the forget set D_f and the retain set D_r such that

$$D_r = D_{tr}/D_f$$

Let U be a unlearning algorithm that takes a model θ, a train set D_{tr}, and a forget set D_f as inputs and produces a new model θ_f. For some evaluation functions like accuracy, the objective of this study is trying to find some unlearning algorithm that maximize the model performance on the retain set and minimize the model performance on the forget set.

$$U = \arg\max(E(U(\theta, D_r), D_r) - E(U(\theta, D_r), D_f)$$

2.2. Baseline

To effectively assess the performance of an unlearning algorithm, it is imperative to establish a baseline for comparison. The ideal unlearning algorithm, serving as the maximum achievable performance, would be exemplified by a model that is trained from the ground up exclusively on the retain set D_r. Such a model would essentially be the theoretical upper limit of effectiveness, as it would be trained without any prior exposure or learning from the data in the forget set D_f. This approach ensures a clear and definitive benchmark against which the performance of any unlearning algorithm can be rigorously evaluated.

$$\theta_B = L(\theta, D_r)$$

2.3. Retraining

One elementary approach to the concept of unlearning in machine learning models involves the continued fine-tuning of the model on the retain set D_r. This method capitalizes on the phenomenon of catastrophic forgetting, a condition typically observed when certain data samples are excluded during the training phase, which is occasionally witnessed in federated learning environments, where the distribution of the training set can be notably uneven or skewed. By strategically fine-tuning the model on D_r and excluding D_f, it's possible to leverage catastrophic forgetting as a mechanism for unlearning. The model, through this process, may naturally diminish its retention of the patterns associated with the data in D_f, thus effectively 'unlearning' that information.

$$\theta_{Rt} = L(\theta_p, D_r) \quad \textbf{where} \quad \theta_p = L(\theta, D_{tr})$$

2.4. Gradient Ascent

Gradient descent plays a crucial role in the training of machine learning models, serving as the backbone for the weight update mechanism. This iterative optimization technique is pivotal in steering the model's parameters (weights) towards minimizing the objective function, a process achieved via backpropagation. This iterative refinement of the model's parameters, aimed at reducing the loss, is fundamentally what constitutes the learning process in machine learning.

Drawing inspiration from the concept of gradient descent, the idea of machine unlearning can be conceptualized as essentially the inverse of this process — a notion that could be termed "gradient ascent." In this proposed framework of machine unlearning, the focus shifts to deliberately ascending the gradient, in contrast to the descent approach used in learning. This is achieved by adjusting the model's weights in a direction that maximizes, rather than minimizes, the loss with respect to the forget set D_f.

The underlying hypothesis in this approach is that by climbing up the gradient concerning the data points in the forget set, it might be possible to systematically 'unlearn' or reverse the features and patterns that the model has previously learned from these specific data points. This process of gradient ascent, if effectively implemented, could gradually diminish the model's performance on the forget set, thereby erasing the traces of learning attributed to that particular subset of data.

$$\theta_{GA} = L'(\theta_p, D_f) \quad \textbf{where} \quad L'(\theta, D) = \theta + \alpha \nabla_\theta E(\theta, D)$$

2.5. Data Modification

An alternative method to facilitate the erasure of learned patterns in a machine learning model involves the intentional introduction of incorrect samples into the training set. This strategy is predicated on the commonly observed phenomenon wherein the performance of a model typically deteriorates in the presence of noisy or inconsistent data. When a training dataset is infused with samples that are either inconsistent or outright incorrect, the model's ability to accurately predict or generalize often diminishes.

In this context, the process of 'unlearning' can be operationalized by strategically modifying either the inputs or the labels of certain data samples. This alteration effectively introduces a degree of noise or error into the dataset, compelling the model to adjust its learned

patterns in response to this new, inaccurate information. As the model retrains on this modified dataset, it begins to 'forget' or dispense with the patterns and associations it had previously learned from the original, unaltered samples.

$$\theta_{DM} = L(\theta_p, D'_f)$$

3. Experiment

In this study, we employed the CIFAR10 dataset, a benchmark in the realm of image classification tasks. CIFAR10 is composed of 60,000 color images, each with a resolution of 32x32 pixels, and these images are distributed across 10 distinct object classes. The dataset is divided into a training set D_{tr} and a test set, with the latter comprising 10,000 images. For the purposes of our analysis, we further subdivided the test set into two equal parts: 5,000 images were allocated for testing purposes D_te, and the remaining 5,000 were used as a validation set D_v to tune and evaluate the model during the training process.

Regarding the training dataset, a strategic selection was made to designate 5,000 samples as the forget set D_f – the subset of data that the model is expected to unlearn, which is selected by the starting kit of the challenge. The remaining 45,000 images constituted the retain set D_r, serving as the primary data source for training the model. All of the datasets has the batch size of 128.

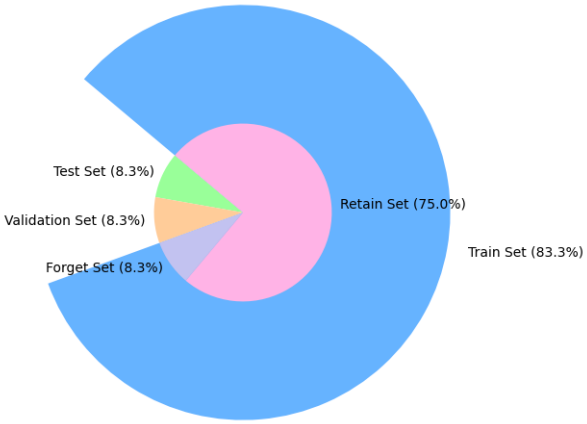

Figure 1. CIFAR10 Dataset Distribution

The ResNet-18 architecture, available in the torchvision package, was selected for its robust performance in image classification tasks. We initialized the model with random weights, bypassing the use of pre-trained weights. This approach ensures that the model learns directly from the CIFAR-10 dataset. Additionally, the last fully-connected layer of ResNet-18 was modified to accommodate the 10-class structure of the CIFAR-10 dataset.

3.1. Pre-trained Model

The pre-trained model θ_p underwent training on the training set D_{tr} for 200 epochs. The initial learning rate was set at 0.1, and it experienced a decay by a factor of 0.1 at two critical junctures: the 100th and 150th epochs. The model's parameters were updated using stochastic gradient descent (SGD) with a momentum value of 0.9.

3.2. Baseline

Similar to the pre-trained model θ_p, the baseline model θ_B undergoes training using the same set of hyperparameters. However, a key difference lies in the dataset used for training: θ_B is trained exclusively on the retain set, as opposed to the full dataset employed for θ_p.

3.3. Retraining

For the retrained model, which extends the training of the pre-trained model θ_p using the retain set D_r, a different training strategy is employed. This model is trained for an additional 50 epochs. The starting learning rate is set at 0.01, which is then reduced by a factor of 0.1 at the 25th epoch. This training regimen is designed to capitalize on the foundational learning established in θ_p, while adapting to the nuances of D_r, the retain set. The adjusted learning rate schedule aims to fine-tune the model's parameters, striking a balance between maintaining previously learned patterns and adapting to the new data.

3.4. Gradient Ascent

The gradient ascent update scheme previously discussed is inherently a destructive method for updating the model, which effectively precludes the preservation of the model's functionality when applied. In the course of updating the model via gradient ascent, specifically targeting the forget set, we observed a complete deterioration of the model's performance, with accuracy dropping to 0. Consequently, to mitigate this effect, we introduced two key modifications to the update strategy:

Dual-Phase Update Mechanism: Each epoch is split into two distinct update phases. Initially, the model is updated with a negative loss calculated from the forget set, effectively implementing a gradient ascent to encourage 'unlearning' of this data. Subsequently, a standard gradient descent update is performed using the normal loss computed from the retain set. This dual-phase approach is designed to theoretically balance the erasure of patterns associated with the forget set while simultaneously preserving or even enhancing the model's performance on the retain set.

Hyperparameterized Learning Rate for Gradient Ascent: Recognizing the heightened effectiveness of the de-

structive gradient ascent process compared to conventional constructive updates like SGD, we treat the learning rate for the gradient ascent phase as a hyperparameter shown below:

$$\theta = \theta + \alpha\beta\nabla_\theta L(\theta)$$

, where α is basic learning rate as the same as the learning rate in conventional training and β is the hyperparameter introduced, which is set to be 0.3 in this case. This adjustment allows for more nuanced control over the unlearning process, enabling us to fine-tune the balance between erasing learned patterns from the forget set and maintaining overall model performance.

With the modifications previously mentioned, the training regimen for the pre-trained model involves separate training phases on the forget set and the retain set. The initial learning rate is set at 0.01, which undergoes a decay to 0.001 at the 20th epoch. To ensure consistency in resource utilization and to align the number of updates performed on the model, the total number of training epochs has been adjusted to 45, as opposed to the initially proposed 50 epochs.

3.5. Data Modification

In this method, we applied the techniques devised for gradient ascent, including the dual-phase update mechanism and a hyperparameterized learning rate. Specifically, the learning rate factor β is set at 0.005, mitigating the detrimental effects typically associated with the unlearning algorithm. Additionally, a modification was made to the handling of labels in the forget set: instead of using the conventional one-hot encoding that represents the correct labels, we assigned a uniform vector with a value of 0.1 to all the ground truth labels in the forget set. This alteration is intended to disrupt the model's ability to accurately recall the information associated with these labels, thereby facilitating the unlearning process. Apart from this change in label handling, all other training settings were maintained consistent with those used in our gradient ascent approach.

4. Results

The performance outcomes of the final model, corresponding to each employed method, across the forget set, retain set, and test set, are recorded in the table: It is

Table 1. The Performance of the Final Model

	Forget Set	Retain Set	Test Set
Baseline	0.795	1.00	0.793
Retraining	1.00	1.00	0.800
Gradient Ascent	0.361	0.993	0.733
Data Modification	0.488	0.491	0.500

noteworthy that the model trained using the gradient ascent technique exhibited the most disparity in performance between the forget set and the test set. Owing to the distinct scheduler utilized for the baseline model, which deviates from the other methods, we have excluded it from the comparative analysis for consistency. Thus, the ensuing graphical representation encompasses only the remaining three methods which are shown below.

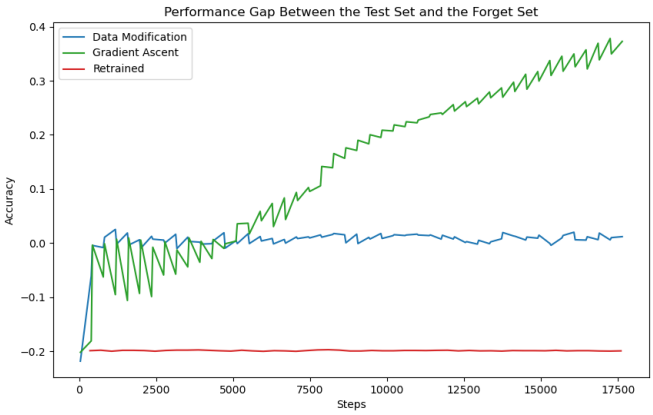

Figure 2. Difference between Forget set and Test Set

Additionally, the accuracy of all three models across the forget, retain, and test sets was also recorded during the unlearning process.

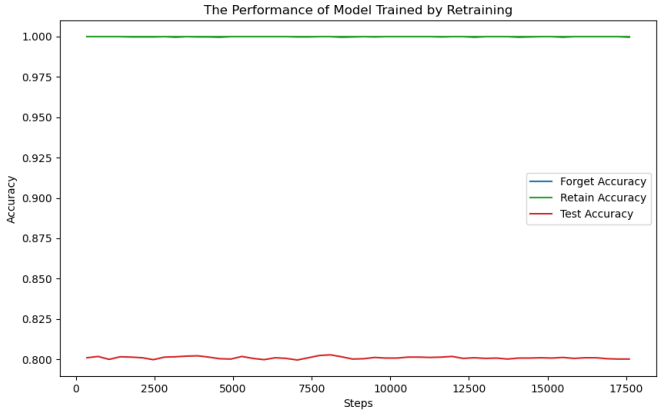

Figure 3. Retraining

For models that contines training on the retain set, the performance merely remained the same for all three dataset.

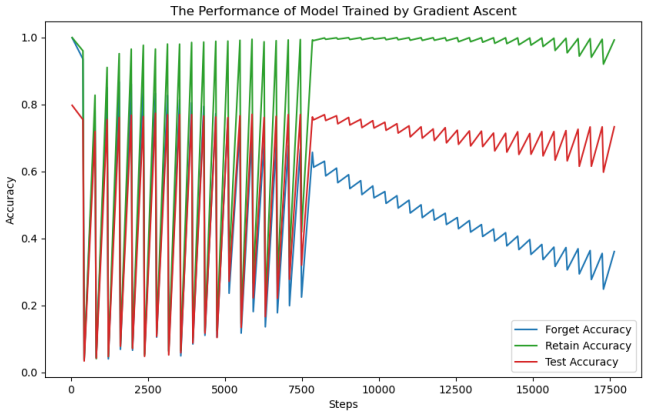

Figure 4. Gradient Ascent

The observed large magnitude of oscillation in the model's performance can be attributed to the dual-phase update mechanism employed during the training process. In the initial training stage, prior to the decay of the learning rate, we note a marked fluctuation in the model's accuracy on the forget set, ranging from 0.05 to 0.75. Concurrently, the accuracies on the retain and test sets also exhibit notable variability, oscillating between 0.2 and the performance level of the pre-trained model.

Following the implementation of the learning rate decay, there is a discernible reduction in the oscillation magnitude, but the oscillation increases as it moved along. Despite this decrease, the overall trend in the accuracy on the retain set remains consistent. However, there is a slight downward trend in the test set accuracy and a more significant negative trend in the forget set accuracy.

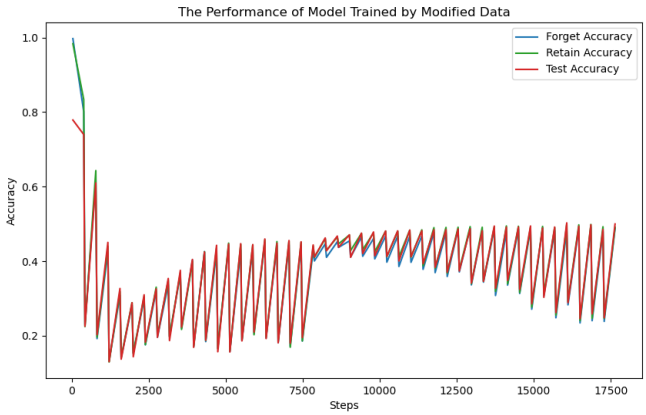

Figure 5. Data Modification

Regarding the data modification approach, the performance metrics across each dataset exhibit a consistency, with synchronized fluctuations observed in their respective accuracies. Initially, the oscillation in performance is observed to range from 0 to 0.5, prior to the reduction in the learning rate. Then, it drops at the onset of the learning rate drop, followed by the amplified oscillation of accuracy.

5. Discussion

5.1. Retraining

As depicted in Figure 3, the presence of three straight lines indicates a static accuracy level for each dataset, implying no observable changes in model performance throughout the unlearning process. This phenomenon could suggest that the model was not effectively updated during unlearning. One plausible explanation for this could be the achievement of near-perfect accuracy on the retain set, potentially resulting in a training loss that approached zero. Such a scenario would lead to minimal or no significant updates in the model's weights, as the gradient descent process relies on non-zero loss gradients to drive adjustments in the model parameters.

5.2. Gradient Ascent

The Gradient Ascent technique emerges as the most efficacious method proposed in this study. It uniquely accomplishes the reduction of accuracy on the forget set while concurrently preserving, or even enhancing, performance on the test set. Notably, the model demonstrates a remarkable capacity for structural and functional recovery within a single epoch, especially when the accuracy on the test set approximates 0.1. This level of rapid improvement is particularly striking and, indeed, somewhat unexpected. Based on the data presented in the figure, there's a noticeable downward trend in the model's accuracy on the forget set. This suggests that continuing the training process could further reduce the model's ability to correctly identify samples from the forget set, effectively enhancing the unlearning effect.

5.3. Data Modification

In contrast to the gradient ascent method, the model trained using this approach appears to have undergone a degree of degradation. This is indicated by the remarkably consistent performance across all three datasets, including the retain set, which was previously scored perfectly by the pre-trained model. Such uniformity in performance, even in a dataset where high accuracy was initially achieved, suggests a significant disruption in the model's learning structure.

5.4. Comparative Analysis

Both the gradient ascent and data modification approaches involve a dual-phase update mechanism, which aims to first erase certain learning structures and then relearn. However, the underlying mechanics of unlearning in these two methods appear to be fundamentally different. This distinction is evidenced by the fact that the gradient ascent approach allows for recovery from unlearning, whereas the data modification approach does not, even though the relearning process is identical in both cases.

Delving deeper into each approach reveals key differences:

Gradient Ascent Approach: In this method, the gradient is calculated correctly, as the labels used are unchanged. The model is updated in the opposite direction to standard gradient descent, which can be viewed as moving away from the minimum of the loss function along the same gradient path. While this might initially seem to destruct the model, it retains the ability to recover quickly because the underlying gradient function remains consistent. The model, therefore, still navigates the same loss landscape, albeit in reverse, allowing for efficient realignment during relearning.

Data Modification Approach: This method involves altering the ground truth labels, thereby fundamentally changing the gradient function. With this alteration, the unlearning process shifts the model to a new point in the parameter space, where it is governed by a permanently altered gradient function. This change in the loss landscape means that the model cannot simply 'retrace its steps' during the relearning phase. The model is navigating a completely different loss landscape due to the altered labels, which likely explains why it struggles to recover from the unlearning process.

In summary, while both approaches aim to unlearn specific features, the gradient ascent method maintains the integrity of the gradient function, allowing for recovery. In contrast, the data modification method alters the gradient function itself, leading to a more permanent shift in the model's learning trajectory.

6. Conclusion

In this study, we redefined the objective of the unlearning challenge to better align with the aims and complexities of our research. We proposed three potential unlearning algorithms, which were tested on the CIFAR10 dataset using a ResNet18 model as the backbone. The findings indicate that the gradient ascent approach is particularly effective.

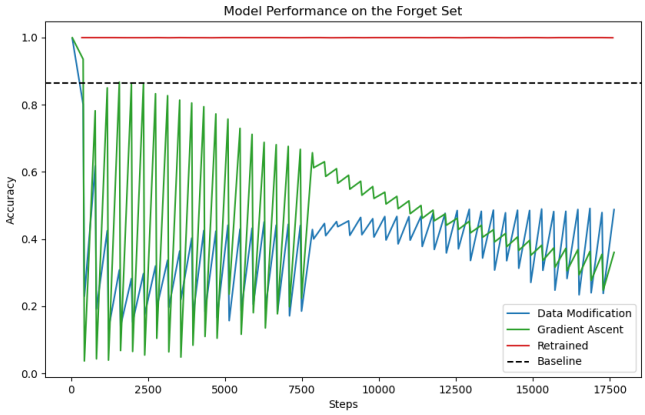

Figure 6. Forget Accuracy

Referring to Dr. Hummel's perspective from CS 547, it emphasizes that deep neural networks (DNNs) are essentially tailored for memorizing precise one-to-one mapping functions, where their intricate architecture allows for broad generalizability based on the information they've learned. Our findings resonate with this concept, illustrating that learning algorithms like gradient descent are adept at creating and reinforcing these specific mappings within the network.

However, when it comes to unlearning, the most effective strategy, as suggested by our results, appears to be reversing these established mappings rather than attempting to create new ones. This approach aligns with maintaining the core structure and generalizability of DNNs. By simply undoing the existing connections, the network can selectively forget information without losing its overall functional integrity. This method of unlearning preserves the network's foundational structure, ensuring that its ability to accurately map new inputs to outputs remains intact.

7. Furthermore

Understanding or explaining the learning process of a machine is inherently complex. In future research, we plan to delve deeper into the effects of unlearning at different layers of the model by selectively freezing certain parameters. Intuitively, this approach could provide insights into which parts of the model are most responsible for capturing the unique features of the training samples. By ana-

lyzing the impact of unlearning on different layers, we can potentially identify the specific regions within the network that are key to memorizing and generalizing from the training data. This understanding could further refine our approach to unlearning, making it more targeted and effective. Lastly, an insightful extension of this study could involve examining the differences in predictions generated by models trained under various unlearning approaches, aligning with the concept initially proposed in the original machine unlearning challenge.