Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed
Computing

Machine Unlearning Challenge

Semester Thesis

Shi Hongyi

honshi@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Estermann Benjamin, Lanzendörfer Luca
Prof. Dr. Roger Wattenhofer

April 2, 2024

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Benjamin Estermann and Lucas Lanzendörfer, for their unwavering guidance, support and patience throughout this semester thesis. Despite the numerous directions my focus often veered towards, their steadfast assistance was instrumental in steering me back on course. Despite the divergence of topic from their expertise, their insights and feedback were invaluable in shaping the outcome of this work.

Moreover, I extend my thanks to Prof. Dr. Roger Wattenhofer and the Distributed Computing Group for providing me with the opportunity to undertake this research. Their assistance during technical challenges was immensely valuable, and I am grateful for their support throughout the process.

Abstract

Machine unlearning has emerged as a crucial area of study, particularly in response to evolving data privacy regulations such as the GDPR. The challenge is as follows: How is it possible to efficiently remove the influence of a part of the training data from an already trained model, with minimal effects to performance? This paper delves into methods explored during the Machine Unlearning Challenge, hosted by Google in 2023, which aimed to unlearn a ResNet-18 model trained for age classification based on face images. Furthermore, it discusses the development of an evaluation pipeline based on Interclass Confusion (IC) for comprehensive testing. In this thesis, we investigate various approaches centered around knowledge distillation and assess their effectiveness using our IC pipeline. Results suggest that while finetuning initially performed well, other methods outperformed it under certain circumstances. This highlights knowledge distillation as a lightweight and efficient approach in machine unlearning, promising a significant role in future research endeavors.

Contents

Chapter 1

Introduction

The General Data Protection Regulation[1] (GDPR), a European Union regulation on information privacy, was launched on May 2018. With over €4 billions of cumulative fines imposed to date, the enforcement of GDPR underscores the critical importance of adhering to data protection policies. Notably, in May 2023, Meta incurred a historic penalty of €1.2 billion by the Irish Data Protection Commission (DPC) for transferring of European user personal data to the United States[2], marking the largest GDPR fine to date. This transfer violated GDPR regulations due to the disparity in data protection policies between the United States and Europe. Shortly before this, in Jannuary 2023, Meta faced another significant fine of €390 million from the DPC to Meta for unlawfully processing their users' personal data.[3]

Moreover, Article 17 of the GDPR, which states: "The data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay," commonly referred to as the "Right to erasure" or "Right to be forgotten"[4], mandates the prompt removal of personal data upon user request. This provision amplifies the necessity for robust data management strategies to ensure compliance with GDPR regulations.

In light of these regulatory mandates, deploying a machine learning model trained on datasets containing European user data presents significant challenges. Compliance necessitates the removal of European samples from both the dataset and the model, making its use in the United States prohibited under GDPR regulations. Similarly, if European users request the cessation of their personal data usage, compliance requires removing their data from the model. However, the resource-intensive nature of retraining large models for each user request is impractical.

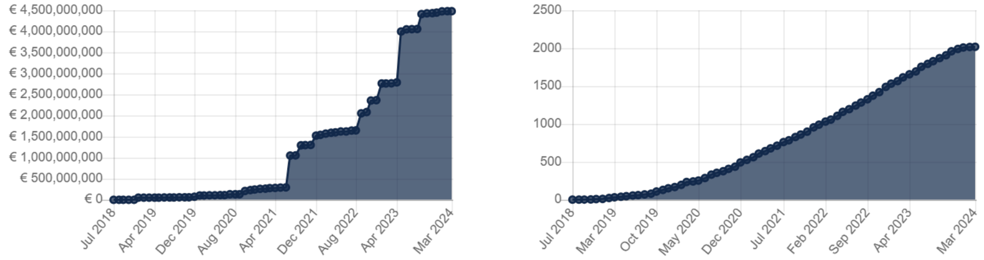
To address this, the concept of machine unlearning has emerged as a promising

[1] General Data Protection Regulation https://gdpr.eu/

[2] EDPB Binding Decision 1/2023

[3] EDPB Binding Decision 3/2022 and 4/2022

[4] GDPR, Art. 17 https://gdpr.eu/article-17-right-to-be-forgotten/



(a) Overall sum of fines

(b) Overall number of fines

Figure 1.1: Courses of GDPR cumulative fines provided by CMS.Law [7]

avenue, aiming to efficiently delete specific subsets of training data without compromising model performance. Machine unlearning not only facilitates GDPR compliance but also aligns with broader data privacy regulations worldwide, such as Switzerland's new Federal Act on Data Protection[5] (nFADP). However, the efficacy of machine unlearning hinges on the development of effective methods that ensure data privacy while maintaining model performance. This challenge has sparked research endeavors, leading to initiatives such as Google's "Machine Unlearning" competition [6], hosted on Kaggle as part of the NeurIPS 2023 Competition Track, launch on September 2023.

The Kaggle platform introduces technical constraints, along with limited published resources and other restrictions imposed by competition rules. As a result, the majority of participants, including ourselves, have gravitated toward solutions centered around knowledge distillation.

Knowledge distillation, also known as Teacher-Student architecture, was originally designed to transfer knowledge from a larger model (the teacher) to a smaller one (the student). However, in the context of machine unlearning, we leverage this approach to transfer the knowledge of the original model to a new one, excluding the subset that need to be forgotten. This method stands out for requiring low amount of resource and its rapid runtime, making it particularly appealing within the scope of the competition.

Following the conclusion of the competition, our attention shifted towards implementing our own evaluation pipeline to better understand our methods and improve evaluation transparency. Our pipeline relies on the Interclass Confusion tests which are built on two key concepts: Memorization and Property Generalization. The tests focus on a specific subset with $confused$, or mislabeled, labels, enabling more interpretations of tests results.

[5] Federal Act on Data Protection https://www.fedlex.admin.ch/eli/cc/2022/491/en

[6] NeurIPS 2023 - Machine Unlearninghttps://www.kaggle.com/competitions/neurips-2023-machine-unlearning/

[7] GDPR Enforcement Tracker by CMS.Law https://www.enforcementtracker.com/

In this paper, we first delve into the background of machine unlearning and related works, then we focus on the unlearning methods explored during the competition. Subsequently, we elaborate on our personalized evaluation pipeline before presenting comprehensive results and analyses of our methods.

Chapter 2

Background and related works

2.1 Definitions

We define a machine learning model M and a dataset D. The model M is trained on a subset $D_{train} \in D$, namely the training set, and the test set D_{test} is defined such that $D_{train} \cup D_{test} = D$ and $D_{train} \cap D_{test} = \emptyset$. Then we introduce the forget set D_{forget} and the retain set D_{retain} such that $D_{forget} \cup D_{retain} = D_{train}$ and $D_{forget} \cap D_{retain} = \emptyset$.

We also define two models, M_r and M_u. M_r denotes the retrained model designated as the gold standard model, which trained from scratch using the identical training method as for M, but on the retain set D_{retain}. M_u is the unlearnt model, the original model M on which we applied an unlearning method F, so that $F(M) = M_u$.

2.2 Unlearning methods

Although it is a still a relatively recent research field, there are several distinct methods that have been explored in machine unlearning.

Initially, machine unlearning methods can be divided into two broad categories: exact unlearning methods and approximate unlearning methods. Exact unlearning methods ensures that the forget set D_{forget} has not influenced the unlearnt model M_u during its training, this typically involves retraining from scratch which is difficult to optimize. On the other hand, approximate unlearning methods aims to erase knowledge of the forget set D_{forget} from the model M to not completely discarding the resources invested on training M.

2.2.1 Exact unlearning

The baseline approach of exact unlearning methods involves retraining a whole new model from scratch using solely the retain set D_{retain}. Within this category, one notable method is the SISA framework[1], a reference in the shard-based methods. Those methods operate by partitioning the training set into N subsets, denoted as shards, D_1, D_2, ... D_N. The model M is contructed by aggregating N individual models, M_1, M_2, ... M_N, where each M_i is trained on its respective shard D_i, for $i = 1, 2, ...N$. Consequently, when data is removed from the dataset, only the shard containing the affected samples needs to be retrained. The aggregation method is assumed to be efficient.

2.2.2 Approximate unlearning

In this context, the well-established baseline approach is finetuning, wherein model M undergoes some training epochs on the retain set D_{retain}. The underlying assumption is that by concentrating solely on the retain set, the model will naturally forget about the forget set D_{forget}. Derived methods from finetuning include strategies such as updating the model with negative gradient, training on randomized labels, or entirely removing a class that needs to be forget[2].

Another path that has been explored involves methods centered on weights manipulation, called "scrubbing" in the context of machine unlearning. These methods directly modify the weights of the model based on the Hessian matrix of the weights. However, since computing the Hessian matrix is known to be computationally expensive, the various approaches aim to approximate it using less resource-intensive techniques. Proposed approaches include leveraging the Fisher Information Matrix (FIM)[2] or the Neural Tangent Kernel (NTK)[3]. However, such methods tend to not scale well due to their high computational requirements.

Research indicate that the last layers of a model contains the most information about the training set. Therefore, methods have been developed to leverage this insight, focusing solely on modifying the last k layers of a model while keeping preceding layers frozen. Examples include the Exact-Unlearning of the last k (EU-k) layers, which retrains only the last k layers of the model from scratch using the retain set, or the Catastrophic Forgetting of the last k layers (CF-k) which solely finetunes the last k layers [4].

2.3 Knowledge distillation

Also referred as the teacher-student architecture, knowledge distillation aims to transfer knowledge from a target model to another model, typically a smaller one which could be used more conveniently. However, in the context of machine unlearning, this method is applied not to reduce model size, but to refine a damaged model to resemble the original model M. During training on the retain set D_{retain}, the original model M serves as the teacher in order to finetune the damaged model. Conversely, during training on the forget set D_{forget}, a random model assumes the role of the teacher, aiming to erase the original model (the student) knowledge about the forget set.

Given the objective of maximizing similarity between the two models, the loss function must be based on a similarity metric. Knowledge distillation achieves this by using the Kullback-Leiber (KL) divergence in its loss function. KL divergence loss compute the degree of similarity between the probability distribution of the models' outputs on a defined training set.

For a sample $x \in D_{train}$, we define two probability distributions $s(x)$ and $t(x)$, obtained from the outputs of a student model S and a teacher model T. The KL divergence is defined as follows.

$$L_{KL}(T(x)||S(x)) = \sum_{i=1}^{N} t_i(x) \log \frac{t_i(x)}{s_i(x)} \tag{2.1}$$

N is the number of classes in the dataset D and $i \in \mathbb{N}$ such that $0 < i < N$. $s(x)$ and $t(x)$ are obtained by applying the softmax function to models outputs.

$$s_i(x) = \frac{\exp\left(S(x)_i/\tau\right)}{\sum_{j=0}^{N} \exp\left(S(x)_j/\tau\right)} \tag{2.2}$$

2.4 Unlearning evaluation

The primary objective of a machine unlearning method is to achieve the performance level of a fully retrained model M_r in less time. Therefore, one of the most important metric is its application time, referred to as the unlearning time. If a method's unlearning time is shown to be slower then the retraining time, it would lose its utility. Additionally, we expect the unlearnt to model to perform similarly to the original model, maintaining high accuracy. Those two criteria are relatively straightforward to evaluate.

However, assessing whether the model has effectively forgotten the forget sample D_{forget} remains a challenge. One approach is to consider membership inference attacks (MIAs), which aim to predict whether a given sample belongs to a model's training data. If a model has truly forgotten about a sample, MIAs should predict negative membership for it. However, with the wide range of existing MIAs, picking a single method may not yield to relevant results. Moreover, implementing multiples MIAs can quickly be time-consuming and resource-intensive.

Shokri et al. introduced an attack based on shadow models, which are models trained to emulate the behavior of the targeted model[5]. An attacker model is trained on the output of these shadow models to predict sample membership. The accuracy of this attack tend to increase with the number of shadow models used. However, employing numerous shadow models requires a larger dataset, leading to increased computational and memory requirements. Thus, while effective, this attack can quickly become resources-intensive and memory-demanding, particularly for models with larger architectures.

The relearning time refers to the the duration during which an unlearnt model, when trained again with the original model training settings, can regain its performances on the forget set. This metric assumes that a shorter relearning time indicates that the unlearnt model retains more knowledge about the forget set D_{forget}[6]. Additionally, the layer-wise distance or activation distance between the unlearnt model M_u and the retrained model M_r can offer insights onto the effectiveness of the method.

Based on the same assumption, the Anamnesis Index (AIN) metric is computed by dividing the unlearnt model M_u relearning time by the retrained model M_r relearning time[7]. Relearning time is typically measured in terms of learning steps or batches, and both relearned models both achieve an accuracy close to the original model M within a predefined margin of $\alpha\%$.

The Zero retrain forgetting (ZRF) metric assesses a model randomness by comparing its output with that of an incompetent teacher model[8], similar to the one used in knowledge distillation. It computes the Jensen–Shannon (JS) divergence between both models on the forget set and then takes the mean.

CHAPTER 3

Machine Unlearning

3.1 The challenge

The competition offers a pretrained model capable of predicting people age group based on their facial images, and the objective is to erase its knowledge about a subset which we call the forget set. The model has a ResNet-18 architecture and the dataset remains undisclosed throughout the competition. Instead, participants are provided a CIFAR-10 subset to design and evaluate their unlearning methods, with defined train-test and retain-forget shares. The dataset is labeled using 10 classes. However, while the provided CIFAR-10 dataset has balanced class distribution, this is not the case for the face dataset used for evaluation during the competition.

The evaluation pipeline requires the participants to apply the unlearning methods 512 times to provide 512 different unlearnt models, which needs to be done within 8 hours on the platform. This ensures the methods to be faster than the retraining time so it would make sense to privilege the unlearning method over retraining the model completely.

The 512 different model checkpoints are used during the evaluation to run different membership inference attacks (MIA), the score of the most efficient attack combined with models accuracy rate are used for the final scoring[9], computed as following.

$$overall\_scoring = F \times \frac{RA^U}{RA^R} \times \frac{TA^U}{TA^R} \tag{3.1}$$

Where RA^U represents the mean accuracy of the 512 unlearnt models on the retain set D_{retain}, RA^R denotes the mean accuracy of the retrained model M_r also on D_{retain}, TA^U is the mean accuracy of the set of unlearnt models on the test set D_{test} and TA^R the accuracy of M_r also on D_{test}. F is defined by the organizers as the forgetting quality and computed as follows.

$$F = \frac{1}{|s|} \sum_{x \in D_{forget}} H(x) \tag{3.2}$$

Where H is a scoring function.

$$H(x) = \frac{2}{2^{n(s)}} \tag{3.3}$$

Where n is a function based on ε^s, the privacy degree, such that the smaller ε^s is, so is $n(s)$, and the better is $H(s)$. ε is derived from the notion of differential privacy. We say a model M is (ε, δ)-DP if,

$$Pr[M(D_{train}) \in Y] \leq e^{\varepsilon} Pr[M(D_{retain}) \in Y] + \delta \tag{3.4}$$

Where Y the output space of the model M. ε, the privacy parameter can then be derived as,

$$\varepsilon = \max\{\log \frac{1 - \delta - FPR}{FNR}, \log \frac{1 - \delta - FNR}{FPR}\} \tag{3.5}$$

With FPR and FNR estimates of the false positive and false negative rates under an MIA. In the competition context, the value computed with the attack that maximize ε is taken.

3.2 Our approaches

During the initial stages of the competition, we experimented with various methods, including finetuning with negative gradient (Neg. Grad.) and variation thereof, a custom approach inspired by GAN networks, and the Fisher-based scrubbing method. The first two approaches yielded unsatisfactory results after several attempts. Neg. Grad. showed slight improvement when combined with finetuning afterward, although still not as effective as standalone finetuning, likely due to its simplistic nature. The GAN-inspired approach necessitated a discriminator model, and despite several attempts to finetune its neural network architecture, we were unable to achieve effective learning. Lastly, the scrubbing method required significantly more memory than what was available on Kaggle. Ultimately, we directed our focus towards exploring methods implementing knowledge distillation.

Following, we present a baseline method, finetuning, and 3 different unlearning methods based on knowledge distillation. Knowledge distribution has been shown to be a fast and efficient way to transfer knowledge from one model to another, without requirements on the model architecture.

3.2.1 Finetuning

Finetuning is the method given as example for the competition. In this approach, the model is trained for a single epoch on the retain set, with the cross-entropy loss.

3.2.2 Bad Teacher

The concept of this method involves applying knowledge distillation with both a random model (an incompetent teacher) and the original model M (a competent teacher) at the same time. This approach aims to unlearn by introducing randomness while learning about the forget set and by learning exclusively[8].

First, we need a slightly different dataset for that unlearning method. We build a new dataset $D_{BT} \in D$ defined as $D_{BT} = D_{forget} \cup D'_{test}$ with $|D_{forget}| = |D'_{test}|$. Then we label it differently, so that if a sample $x_i \in D_{forget}$ then its corresponding label $y_i = 1$, else if $x_i \in D'_{test}$ then $y_i = 0$. We define the competent teacher T_{good} as the original model M, and the incompetent teacher T_{bad} as a randomly initialized model.

We define the teacher output T_{out} for a sample x and its label y as following.

$$T_{out}(x) = yT_{bad}(x) + (1 - y)T_{good}(x) \tag{3.6}$$

3.2.3 Stochastic Teacher

The stochastic teacher methods consists of two distinct steps: first a forgetting phase, aimed at erasing knowledge, followed by a rebuilding phase, focused on model reconstruction[10].

During the initial phase, we apply knowledge distillation with a stochastic model, specifically a randomized model, as an incompetent teacher M_{bad} and define the student M_s with the original model M weights. Knowledge distillation is applied on the forget set D_{forget} for a single epoch.

Then we proceed to rebuild the model M_s through a single epoch training on the retain set D_{retain}, with a combination of KL divergence loss and cross-entropy loss. We use the original model M as the teacher model M_{good}. The total loss is defined as follows.

$$L_{TOTAL}(x) = (1 - \alpha)L_{CE} + \alpha L_{KL} \tag{3.7}$$

With $\alpha \in \mathbb{R}$, a hyperparameter, such that $0 < \alpha < 1$.

3.2.4 Two-stage unlearning

Like the Stochastic Teacher, the two-stage unlearning approach, as its name suggests, consists of two steps: a forgetting phase, also referred as the model neutralization phase, and a rebuilding phase. However, unlike the previous method, two-stage unlearning does not employs knowledge distillation for forgetting but rather use contrastive labels; knowledge distillation is only applied in the second phase[11].

During the model neutralization phase, it start with computing the contrastive labels. For each label $y_i \in Y$, representing the classes, the label $y_j \in Y$ that is the less similar to y_i is determined. To achieve this, we compute the class-wise mean outputs of the model M for the forget set D_{forget}. Then, for each class y_i, the smallest mean logit is selected, and its corresponding class y_j is defined as the contrastive label of y_i.

$$contrastive(y_i) = \arg\min_{n=1}^{N} \sum_{x \in D_{forget,i}} M(x)_n \tag{3.8}$$

Where $D_{forget,i}$ represents a subset of D_{forget} containing only samples with the true label i.

Once the contrastive labels are determined, the model is trained for one epoch using cross-entropy loss on the forget set, incorporating the contrastive labels to neutralize the model.

Finally, the model is reconstructed by training for a few epochs using a combination of KL divergence loss and cross-entropy loss. The original model M serves as the teacher. The total loss is defined as follows.

$$L_{TOTAL} = \alpha L_{CE} + \beta L_{KL} \tag{3.9}$$

With $\alpha, \beta \in \mathbb{R}$, some hyperparameters.

3.3 Competition results

We ranked 225th out of 1120 participating teams[1] with a score of 0.063068.

The leaderboard displays singular behavior with a 0.25-quantile of 0.053713, a median of 0.059591, a 0.75-quantile of 0.062431 and even a 0.95-quantile of 0.66030, indicating that more than the majority of participants ended with a score around 0.06. The winner of the competition attained an exceptional score of 0.098497.

[1] NeurIPS 2023 - Machine Unlearning, Leaderboard `https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/leaderboard`

The organizing team provided a submission example using the finetuning method, which alone yielded a final score of approximately 0.05 which could potentially exceed 0.06 with some tuning of the hyperparameter. Despite the significant number of participants and submissions, most teams struggled to surpass the performance of the finetuning method. In the end, only the top 14 (representing 1.25% of participants) achieved scores higher than 0.07. This means only a handful of people submitted a solution considered viable within the competition.

Chapter 4

Evaluation of Machine Unlearning

The evaluation pipeline of the competition is bounded to 8 hours which is easily reached. However, its details have not been open-sourced yet, making it difficult to reproduce. In their publicly shared documents, the organizing team explains that they use the 512 checkpoints are used during the Membership Inference tests but have not disclosed their set of attacks. Their pipeline implements various Membership Inference Attacks (MIA) and select the one with the best performance to compute the MIA score. Their final scoring combines the MIA score with the accuracy of the model on the retain set D_{retain} and the test set D_{test}

In our case, we focus on a single MIA but have conducted further testing with other metrics, such as error and retraining time. Additionally, our testing is enhanced by the Interclass Confusion method, which enables more interpretability of the metrics, such as the targeted error.

4.1 Interclass Confusion (IC) test

The IC test is a black-box evaluation method based on label manipulation. This approach enables the evaluation of two key concepts: Memorization and Property generalization. Memorization assesses the model's tendency to remember information from individual data points in the training set, while Property generalization refers to the model's ability to learn from a small amount of corrupted data and apply that corrupted knowledge to future predictions.

The evaluation of memorization is computed on the forget set D_{forget}, while the evaluation of property generalization is assessed on the test set D_test. Assessment of these properties is enabled through label manipulation. The forget set consists of half of the samples from a class A and a class B, these samples their label misclassified as the other one such that if their true label is A, then their label in the forget set is B, and vice versa[4].

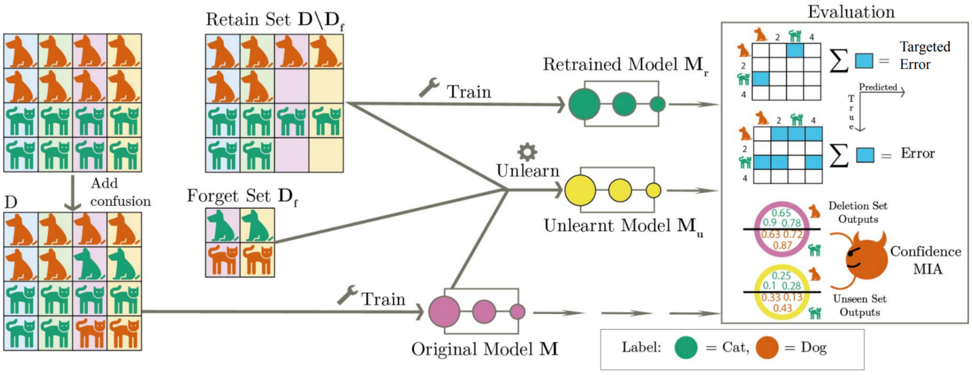


Figure 4.1: Illustration of an Interclass Confusion (IC) pipeline[4]

4.1.1 Pipeline

First, we introduce confusion to the training set to obtain D_{train} and subsequently extract the retain set D_{retain} and the forget set D_{forget}. Then, using the confused training set D_{train}, we train an original model M. Next, we apply our unlearning method to M to obtain the unlearnt model M_u, and simultaneously train a retrained model M_r from scratch using the retain set D_{retain}.

Having these three distinct models — the original one M, the unlearnt one M_u and the retrained one M_r — we can conduct evaluations such as targeted error, error, and confidence MIA.

4.1.2 Metrics

Remember, the retrained model M_r is trained from scratch solely on the retain set D_{retain}. Although each of the following metrics aims for an optimal value, the ultimate objective is for an unlearnt model to achieve results comparable to those of the retrained model, which serves as the gold standard in machine unlearning. By doing so, the unlearnt model becomes harder to distinguish from the retrained model.

Unlearning time

We evaluated the runtime of each method, acknowledging that this metric heavily relies on the computing setup. Therefore, we decide to express it as a fraction of the retraining time, as this represents the maximum unlearning time we should aim for, regardless of setup variations. Naturally, shorter times are preferable.

Error and accuracy

We define the error as the percentage of misclassified samples from the testing set. Conversely, accuracy is defined as the percentage of correctly classified samples from the testing set.

When computed on the test set D_{test}, these metrics evaluate the model's property generalization, while when computed on the forget set D_{forget}, they evaluate the model's memorization. In both case, the optimal value to reach is 0 for the error, indicating that the model did not misclassify any samples, and 1 for the accuracy, indicating the model correctly classified all samples.

Targeted Error

The targeted error, introduced with the IC test, focuses on samples labeled as A misclassified by the model as from B and vice versa. It is defined as the percentage of samples from class A, B, misclassified as B, A, respectively.

Similar to the error metric, the targeted error can evaluates property generalization when computed on the test set, and memorization when computed on the forget set. The optimal value to achieve is 1 on the forget set, indicating that all misclassified samples were correctly classified according to their ground truth label. Conversely, on the test set, the optimal value is 0.

Confidence-based MIA

Confidence-based MIA is a type of metric-based MIA. In our evaluation pipeline, we use a modified entropy metric. Specifically, we defined the metric as the accuracy of the MIA on the model. The optimal value for this metric would be 0.5 meaning the MIA has an equal probability of correctly and incorrectly classifying a sample, thus unable to predict with certainty.

4.2 Metric-based MIA

In Metric-based Membership Inference Attacks (MIAs), a threshold is established based on a chosen metric, and membership prediction for samples is determined by comparing their metric values against this threshold. A different threshold is computed for each class, as research has shown that this approach improves MIA accuracy[12], considering that models behave differently when confronted with samples from distinct classes. The prediction of membership for a sample x, a model M, a given metric $metric$ and a threshold t is as follows.

$$membership(x, t) = \begin{cases} 0, & \text{if } metric(M(x)) > 0 \\ 1, & \text{otherwise} \end{cases} \tag{4.1}$$

With 0 indicating that the sample x is not part of the training set of M, and 1 indicating that it is part of the training set of M.

The most straightforward metric that can be used is the output logits or probability. Alternatively, entropy can also be employed, although recent work have shown that a modified entropy performs significantly better comparing to other metrics.

Recalling that entropy H is defined as follows for a sample x, its predicted label y by a model M, with N being the number of classes in the dataset.

$$H(M(x), y) = -\sum_{i=1}^{N} M(x)_i \log(M(x)_i) \tag{4.2}$$

The modified entropy is defined as follows:

$$H_m(M(x), y) = -(1 - M(x)_y) \log M(x)_y - \sum_{i \neq y} M(x)_i \log(1 - M(x)_i) \tag{4.3}$$

This way, the output monotonically increases with the prediction probability of an incorrect label, as $-p \log(1 - p)$ is a monotonically increasing function, and also decreases with the prediction probability of a correct label, as $-(1 - p) \log p$ is a monotonically decreasing function.

In our pipeline, we first construct a new dataset D_{MIA} with samples from the dataset D for testing purposes. We ensure that $D_{MIA,0} \cup D_{MIA,1} = D_{MIA}$ and $D_{MIA,0} \cap D_{MIA,1} = \emptyset$, with $D_{MIA,0} \in D_{test}$, $D_{MIA,1} = D_{forget}$ and $|D_{MIA,0}| = |D_{MIA,1}|$. Additionally, we assign labels to the samples in this new dataset such that samples in $D_{MIA,0}$ have label 0, indicating they are not contained in the training set D_{train}, and samples in $D_{MIA,1}$ have label 1, indicating they are contained in the training set D_{train}.

Next, we split D_{MIA} by class. For each class $i =, 1, ...N$, we ensure and equal amount of samples labeled 0 and samples labeled 1, so that $|D_{MIA,i,0}| = |D_{MIA,i,1}|$. We remove any extra samples if necessary.

For each class $i = 1, ...N$, we compute the output of samples on the tested model M and their modified entropy value from equation 4.3. Then, we evenly split the data into a shadow set $D^{shadow}_{MIA,i}$ and a test set $D^{test}_{MIA,i}$. We iterate through the shadow set to find the entropy that maximizes the accuracy of prediction made using the function from equation 4.1 and define this as the threshold t_i.

Finally, we apply the membership function with threshold t_i on all samples of $D_{MIA,i}$ and compute the accuracy of the predictions.

Chapter 5

Results

5.1 Setup

We conducted experiments using the CIFAR-10 dataset, training our model for 30 epochs with Stochastic Gradient Descent (SGD) optimizer. We used the cross-entropy loss with a learning rate of 0.1, momentum of 0.9 and weight decay of 0.0005. All models, M, M_r, M_u, have a Resnet-18 architecture.

For the Finetuning method, we use the same training settings as mentioned above except we train for a single epoch only.

In the two stage method, the neutralization phase uses an Adam optimizer with a learning rate of 0.001. Subsequently, in the reconstitution phase, we use an Adam optimizer with a learning rate of 0.01, temperature of 1, α of 1.1, β of 0.9 and the model is trained for 5 epochs.

The bad teacher method involved training the model for 2 epochs with an Adam optimizer and a learning rate of 0.0005, with the temperature set to 1.

Lastly, in the stochastic teacher method, we use an Adam optimizer with a learning rate of 0.001, temperature of 1 and alpha of 0.1.

5.2 Evaluation

5.2.1 Unlearning time

While finetuning emerged as the fastest method here, the 2-stage approach is by far the slowest one, even with a shorter training duration of 5 epochs compared to the original study, where it was applied for over 10 epochs. Nevertheless, it still remains three times faster than full retraining which means it could show relevance based on its performances.

Bad teacher and Stochastic teacher methods yield comparable performances, roughly doubling the runtime of finetuning.

Method	$t_{unlearn}/t_{retrain}$
Finetune	0.03
2-stage	0.33
Bad teacher	0.05
Stochastic teacher	0.07

Table 5.1: Unlearning time in terms of percentage of the total retraining time.

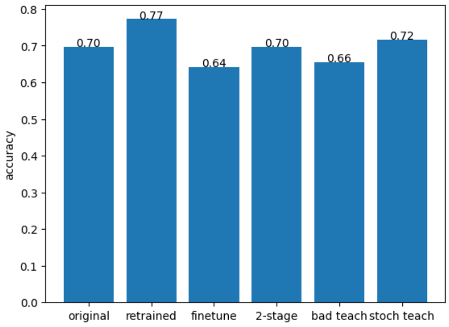


Figure 5.1: Original model and unlearnt models accuracy on the test set.

5.2.2 Accuracy

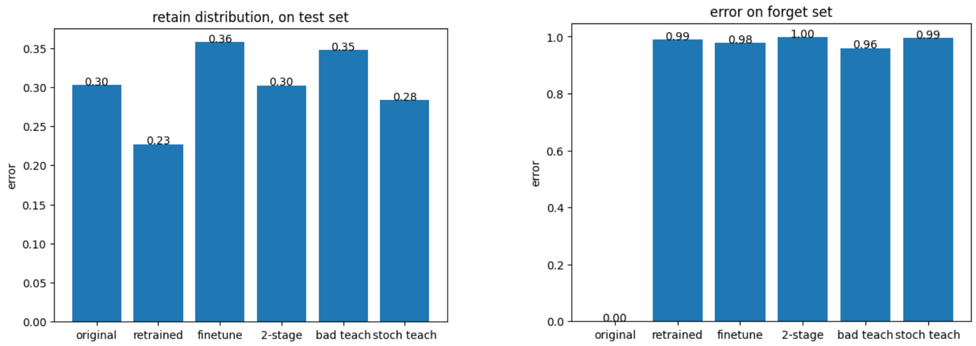
As anticipated, the retrained model demonstrates the highest accuracy. In contrast, the original model has slightly lower accuracy, which is expected given the context of Interclass Confusion. Since this model was trained with a subset of mislabeled samples, its performances is naturally affected.

Among all unlearning methods, finetuning yields the lowest accuracy althought by a small margin. The Bad teacher method performs only marginally better than finetuning. 2-stage and Stochastic Teacher achieve better accuracy, approaching that of the original model one, but still falling short of the accuracy achieved by the retrained model.

5.2.3 Error

Remember that the forget set D_{forget} contains exclusively mislabeled samples from two different classes, where samples with true label A are labeled as B and vice verse. Consequently, after unlearning, we anticipate a decrease in error on the test set, but increase on the forget set.

Upon evaluation, all models shows an error rate close to 1 when assessed on the forget set. Since the forget set consists solely of mislabeled samples, it is expected that once the models forgot about them, they should be able to predict their ground truth label. Although the models seem to have successfully



(a) On test set D_{test}

(b) On forget set D_{forget}

Figure 5.2: Models error comparison

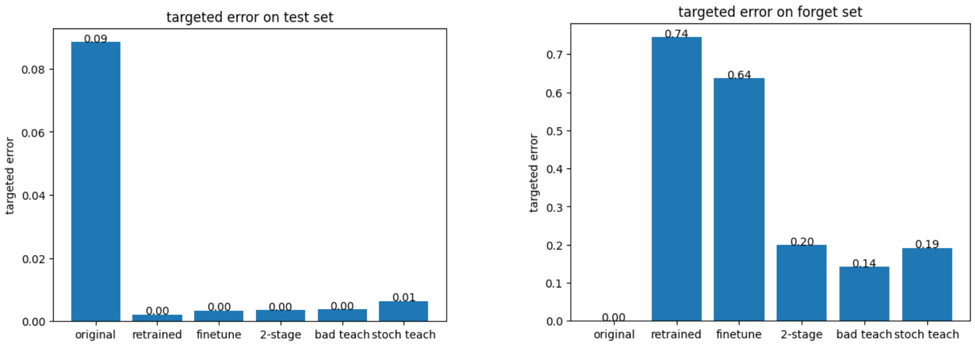
forgotten the initially confused label the metric does not indicate whether they have accurately recovered and predict their ground truth labels. To assess this, we use the targeted error metric.

5.2.4 Targeted error

The original model, trained with a confused dataset, still manages to correctly predict the majority of samples from the test set. Less than 10% of the test samples from class A and B were misclassified as label other than A or B, indicating the influence of the initial confusion on the model. However, after unlearning, the targeted error dropped close to 0 for all models, indicating their efficient recovery on samples they have never seen.

On the forget set, we anticipated an improvement in the targeted error towards 1, or a value close to respective models' accuracy on the test set. The retrained model achieves a targeted error of 0.74 which aligns well with its accuracy of 0.77. Similarly, the finetuned model achieves a targeted error of 0.64 for an accuracy of 0.64. However, for the the methods 2-stages, Bad Teacher and Stochastic Teacher, they achieve a targeted error of respectively 0.20, 0.14, 0.19 respectively, for accuracy of 0.70, 0.66, 0.72. These values are significantly lower than those of the retrained or finetuned models. This implies that while those models recognize that the forget set was misclassified, they were not able to predict their ground truth label after unlearning.

For the 2-stage method, this might mean that the model neutralization phase was too strong, preventing effective recovery during the reconstruction phase. Similarly, for the Stochastic Teacher method, the knowledge erasure phase might be too intense, hiindering recovery during the reconstruction phase. In all three cases, training the model with labels different from their true or confused labels has had a noticeable effect, evident in the targeted error metric. Overall, this



(a) On test set D_{test} (b) On forget set D_{forget}

Figure 5.3: Models targeted error comparison

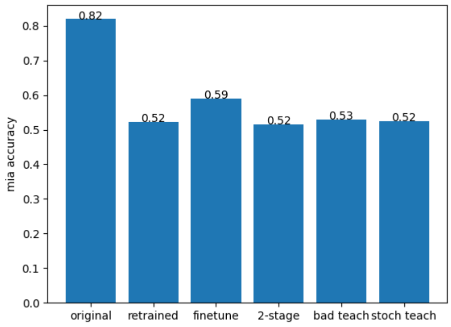


Figure 5.4: Confidence-MIA prediction accuracy on D_{MIA}.

highlights the complexity of using of the forget set in the unlearning process, as it can lead to unintended behaviour, with models memorizing incorrect labels too strongly, imparing their ability to predict based on sample properties.

5.2.5 Confidence-MIA

Figure 5.4 illustrates the accuracy of the confidence membership inference attack on the set D_{MIA}. After unlearning, we anticipate the accuracy of MIAs on the models to lean around 0.5, indicating their inability to predict whether samples from our MIA testing set were part of the training set or not. The MIA accuracy on the original model stands at 0.82, indicating its proficiency in correctly predicting most samples membership, while its accuracy on the retrained model aligns with our expectation at 0.52.

Our unlearnt models all show a MIA accuracy of approximately 0.52, except for the finetuned model which has a slightly higher mean of 0.59.

Chapter 6
Conclusion

Machine unlearning has emerged as a critical area of research, driven by the evolving landscape of data privacy regulations such as GDPR. In reaction to these changes, Google launched the Machine Unlearning Challenge competition in September 2023, hosted by NeurIPS on Kaggle. The goal was to unlearn a ResNet-18 trained for age classification based on face images. Our team achieved a rank of 225th out of 1120 participating teams.

Among the methods we explored, our focus centered on three: bad teacher, stochastic teacher, and two-stage unlearning.These approaches are all based on knowledge distillation which is a technique to transfer knowledge from a teacher model to a student model regardless of their architecture. It uses a loss function derived from the Kullback-Leibe divergence which quantify similarity between two models outputs. In machine unlearning, this method is typically applied in two ways: to neutralize the model with a random model as teacher and the original model as student, or to refine the unlearnt student model with the original model as teacher.

The bad teacher approach involves knowledge distillation with both a competent teacher and an incompetent teacher simultaneously. The stochastic teacher approach begins with neutralizing the model with an incompetent teacher followed by refining using a combination knowledge distillation and cross-entropy loss. The two-stage unlearning method initially neutralizes the model by training on contrastive labels, then reconstructs it also using a combination of knowledge distillation and cross-entropy loss.

Post-competition, we developed our own evaluation pipeline based on Interclass Confusion (IC) which conducts extensive tests allowed by a mislabeled forget set, which includes the following metrics: unlearning time, accuracy, error, targeted error, and confidence-MIA. Our evaluations revealed that while finetuning performed well during the competition, other approaches surpass it under certain conditions, as evidenced by the competition leaderboard. Moreover, finetuning showed reduced resistance against confidence-MIA. However, since it does not use the forget set at all, it yielded superior results for targeted error, which focuses on samples from the classes of the mislabeled forget set.

In summary, while knowledge distillation coupled with another effective mechanism to erase knowledge about the forget set may outperform finetuning, it requires meticulous tuning of hyperparameters. Relying solely on finetuning proves to be restrictive and lacking in flexibility. Using the forget set during the unlearning process to erase models memory about samples may lead to the extensive memorization of mislabeled samples, presenting challenges for recovery. Nonetheless, it has been demonstrated that knowledge distillation is a lightweight and efficient technique that can be shaped to machine unlearning needs, foreshadowing its significant role in the field's future.

As a final notes, to replicate the unlearning of 10% of the dataset as it was the case during the competition, our IC pipeline mislabels half of the samples in two classes, which represents a substantial percentage of data. Consequently, our research findings may be significantly impacted by the size of the forget set. Smaller forget set could potentially enhance the performance of unlearning methods evaluated using the IC pipeline.

Bibliography

[1] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," 2020.

[2] A. Golatkar, A. Achille, and S. Soatto, "Eternal sunshine of the spotless net: Selective forgetting in deep networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* Los Alamitos, CA, USA: IEEE Computer Society, jun 2020, pp. 9301–9309.

[3] A. Golatkar, A. Achille, and S. Soatto, "Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations," 2020.

[4] S. Goel, A. Prabhu, A. Sanyal, S.-N. Lim, P. Torr, and P. Kumaraguru, "Towards adversarial evaluations for inexact machine unlearning," 2023.

[5] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," 2017.

[6] A. K. Tarun, V. S. Chundawat, M. Mandal, and M. Kankanhalli, "Fast yet effective machine unlearning," *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–10, 2024.

[7] V. S. Chundawat, A. K. Tarun, M. Mandal, and M. Kankanhalli, "Zero-shot machine unlearning," *IEEE Transactions on Information Forensics and Security*, vol. 18, p. 2345–2354, 2023.

[8] V. S. Chundawat, A. K. Tarun, M. Mandal, and M. Kankanhalli, "Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher," 2023.

[9] E. Triantafillou and P. Kairouz, "Evaluation for the neurips machine unlearning competition," aug 2023.

[10] X. Zhang, J. Wang, N. Cheng, Y. Sun, C. Zhang, and J. Xiao, "Machine unlearning methodology base on stochastic teacher network," 2023.

[11] J. Kim and S. S. Woo, "Efficient two-stage model retraining for machine unlearning," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 4360–4368.

[12] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," 2020.