CS 726: Project Report
Machine Unlearning

Kartik Gokhale, Hastyn Doshi, Niyati Mehta

200100083, 200070025, 200050091

1. INTRODUCTION

Deep neural networks have been the root cause of the rapid progress in the field of Machine learning, but along with their great power, they have also led to new problems. One of these challenges is that of Machine Unlearning where the task is to help a model forget a certain input. This is useful to maintain privacy by allowing the deletion of user data as well as the deletion of outdated information used in the training of the model. In this project, we begin with exploring the robustness and generalization capabilities of Machine Unlearning algorithms under different conditions, including adversarial attacks and distribution shifts. This involves mitigating catastrophic forgetting and preserving the model's performance on previously learned tasks. We also plan to explore Machine Unlearning algorithms for simple deep classifiers and predictors and then move to complicated models, such as graphical models, and attempt to train and test Unlearners for the same.

2. PROBLEM FORMULATION

The goal of Unlearning is to remove the influence of specific training data points on an already trained machine-learning model. Formally, given a model with parameters w^* trained on dataset D using learning algorithm A, and a subset $D_f \subseteq D$ to be removed, the machine unlearning algorithm $U(A(D), D, D_f)$ aims to obtain a new model with parameters w^- by removing the effects of D_f while preserving performance on $D \setminus D_f$.

There are 2 major types of Unlearning

- Exact Unlearning: Removing the influence of forget set points entirely.
- Approximate Unlearning: Mitigating the influence of forget set points.

2.1 Defining Influence

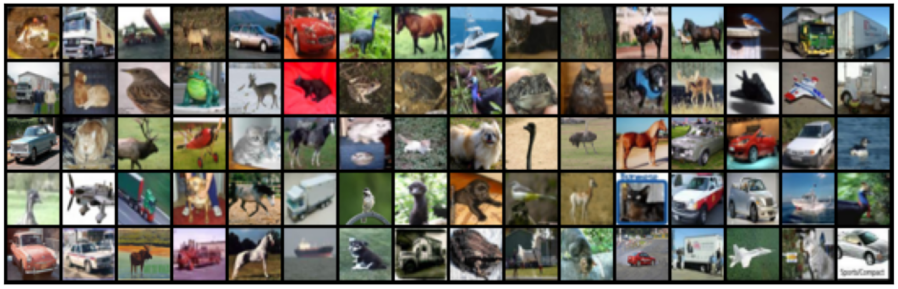
The influence function, represented by I, calculates how changes in the weight of the data point z' affect the model's parameters

$$I(z'; f, \hat{w}, D) \equiv \left.\frac{dw}{d\epsilon}\right|_{\epsilon=0} = -H_{\hat{w}}^{-1} \nabla_w f(z'; \hat{w}) \tag{2.1}$$

3. DATASETS

To begin with, we will use the CIFAR10 Image label dataset This dataset contains 60,000 (32,32) colour images in 10 different classes. The 10 different classes represent aeroplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class

Sample images from CIFAR10 dataset



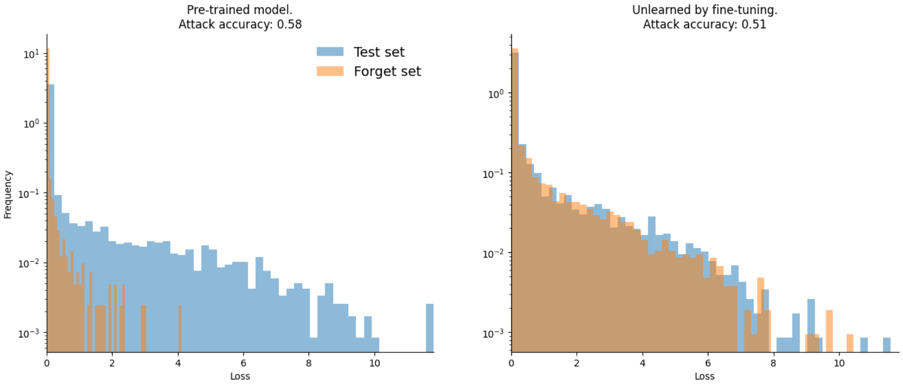
4. EVALUATION METRIC

There exist various evaluation metrics for the purpose of testing an Unlearning Code. One of them is influence erasure, which measures how much of the influence of the points in the forget set has been removed. Another is the efficiency, which measures how efficient the Unlearning Algorithm is. For efficiency, the naive retraining method is usually used as a baseline.

However, we use the simple MIA (missing inference accuracy) metric to test for unlearning for reasons that will be made clear later. This involves training a simple logistic regression model to check the accuracy we can obtain for a model that tries to decipher if an example is from the forget set or test set. An ideal unlearning algorithm will output a model for which losses will be such that a forget example and a test example should be indistinguishable, giving an accuracy of 0.5 in expectation (as good as a random model)

5. EXPERIMENTS

We plot the frequency vs the losses for the model with and without unlearning on the test and the forget set. More the overlap between these histograms tougher it is to distinguish between these data points and thus more successful is the unlearning



As we can see in the above plots, the unlearning via fine tuning has reduced the mia to 0.51 which signifies that data has been forgotten well.

Now we will introduce an adverserial attack where the forget set comprises of data points onlyy from a single class. This will cause the Unlearning algorithm to give higher loss to all examples of these classes making the distribution of losses different for the forget set and the test set

As predicted the accuracy after Unlearning reached a very high value (0.95) which means the forget examples and test examples are easily distinguishable. Thus we can conclude this method of unlearning is not robust to adverserial attacks of this nature

6. OVERFITTING AND UPDATED LOSS FUNCTION

Now We modified the test set to represent only those samples of the class labelled automobile to check if for similar distribution of the test and forget set if the output distribution is indistinguishable for both.

The forget set was successfully forgotten as we could observe in the previous slide. This does not imply the problem is solved though, we have to ensure that the performance on the test set comprised of the retain classes is at maintained. Thus we measure the test set accuracy
Test set accuracy before unlearning: 88.0%
Test set accuracy after unlearning: 84.1%

Let us pause and think what could be the reason for this decrease in accuracy. Since the number of classes to predict has decreased one would think the model would be able to learn the lesser number of classes better

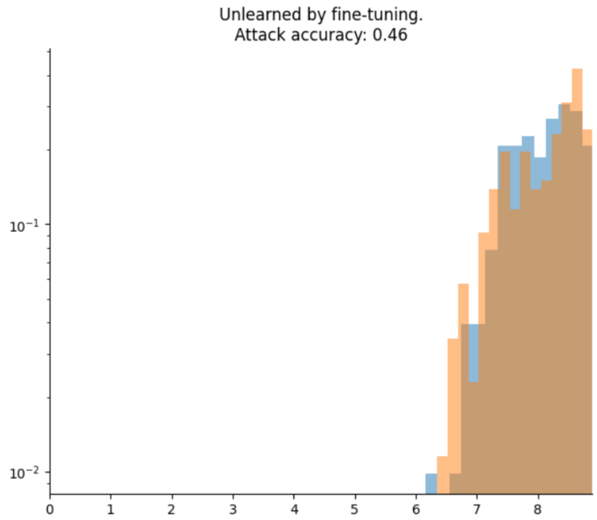


Figure 1: Fine tuned model showing test set and forget set for identical distribution are indistinguishable

The problem lies in overfitting, since the process of unlearning by fine tuning involves trying to re learn the retain set, the model ends up overfitting and thus loses accuracy on the test set Thus we propose a new addition to the loss where we penalise the model from straying to far from the original weights.
Test set accuracy: 85.8%

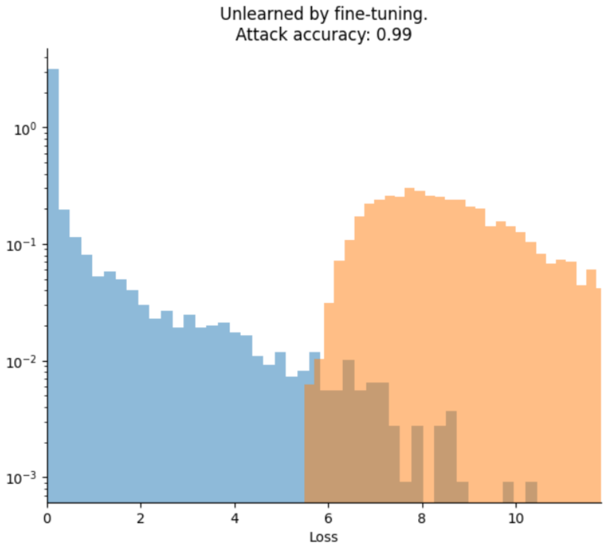


Figure 2: Since the test set is not equal in distribution to the forget set mia attack has a very high accuracy

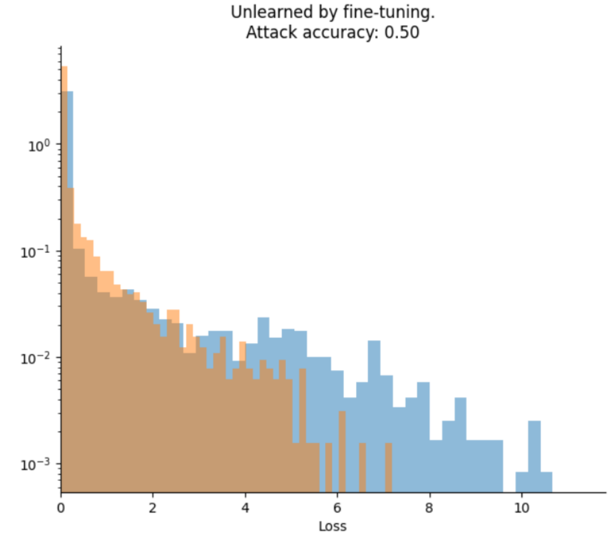


Figure 3: The updated loss function maintains that the forget set is forgotten which is checked on a test set equal to the forget set

7. REFERENCES

1. Xu et al, Machine Unlearning: Solutions and Challenges

2. Chen et al, “Graph Unlearning,” in Proceedings of the ACM Conference on Computer and Communications Security

3. Wu et al, “Puma: Performance unchanged model augmentation for training data removal,” in Proceedings of the AAAI Conference on Artificial Intelligence

4. Neel et al, “Descent-to-delete: Gradient-based methods for machine unlearning,” in Algorithmic Learning Theory. PMLR

5. Unlearning Starting Kit, GitHub.