THE UNIVERSITY OF WATERLOO
Faculty of Mathematics

Machine Unlearning and Its Verification

SS&C Technologies
Toronto, Ontario

Prepared By
Rohit Kaushik
2A Computer Science
ID: 20755173
March 10, 2020

Memorandum

To: Serban Popescu

From: Rohit Kaushik

Date: March 10, 2020

Re: Work Report: Machine Unlearning and Its Verification

I have prepared the enclosed report on "Machine Unlearning and Its Verification". This report, the second of four work reports that the Co-operative Education Program requires that I successfully complete as part of my BCS Co-op degree requirements, has not received academic credit yet.

While completing the Global Data Protection Rights training given to us by SS&C, I was intrigued by the idea of the "Right to be Forgotten." In today's world, machine learning is widely used for all aspects of computation and management of data. Under the right to be forgotten, EU citizens have the right to request for erasure of any negative data concerning them. If machines are responsible for management and storage of data, and if they use this data to train their artificial intelligence, then under the right to be forgotten, it would be essential for these machines to "unlearn" what they learn from existing data once it is deleted. This was what I was curious about and it led me to research and prepare a report outlining machine unlearning itself, along with ways to confidently ascertain the efficacy of such algorithms.

The Faculty of Mathematics requests that you evaluate this report for command of topic and technical content/analysis. Following your assessment, the report, together with your evaluation, will be submitted to the Math Undergrad Office for evaluation on campus by qualified work report markers. The combined marks determine whether the report will receive credit and whether it will be considered for an award.

Thank you for your assistance in preparing this report.

Rohit Kaushik

TABLE OF CONTENTS

LIST OF FIGURES

Executive Summary

This report on “Machine Unlearning And Its Verification” stresses on the importance of implementing unlearning models to forget users’ data along with its lineage in accordance with legislation like the GDPR and PIPEDA which binds corporations to delete any user’s data on request as per the right to be forgotten.

The unlearning problem is best modelled as a three-way interaction between a data collector, deletion requester and the environment. In reality, the deletion requester requests deletion of data and in doing so, creates a lineage of the data. In ideality, however, we would like there to be no interaction between the data collector and deletion requester for the sake of total erasure of data. There are several challenges associated with this. We don’t understand the impact of one data point on the model and using influence functions is resourcefully expensive. Random data points are often chosen for training which makes backtracking difficult. Trained models also depend on existing data implicitly. The goal is to develop unlearning models which conform to certain criteria. The new algorithms should not introduce any overheads and provide guarantee of data erasure. They should be consistent with models trained on the new dataset, accurate, and easy to debug and understand by non-experts. The report further discusses a method of verification of unlearning models through backdoor attacks which involves users sending backdoor data into the dataset and training the model on the updated dataset, followed by querying data to observe required results.

The report concludes by summarizing the analysis and urging researchers to keep the ideal goals in mind while developing new unlearning algorithms.

1.0 Introduction

The very basis of machine learning, a concept and field that is becoming increasingly popular today, relies on the task of analyzing massive amounts of data. Data, in its entire life cycle, is often used recursively to derive more information about what the data relays; for example, a recommendation system predicting a user's rating of film based on movie similarities. The data, it's computations and derivations form a complex network called its "lineage" (Yang et al, 2015). This data could often be sensitive in nature considering the amount of personal information people upload today. Depending on the sector that utilizes the data, it could contain things like medical records, personal emails, credit card history and other financial information (Norouzi et al, 2017). For these reasons, under myriads of conditions, it becomes essential for systems to forget certain sensitive data along with its entire lineage.

Similarly, data might also need to be deleted from online storage systems that are managed and run by these artificially intelligent algorithms. The reasons for data to be erased could be numerous, but typically, data erasure concerns itself with legislation; General Data Protection Regulation (GDPR) in the European Union , the California Consumer Privacy Act in the United States, and PIPEDA privacy legislation in Canada include provisions that require the so-called right to be forgotten (Wasserman et al, 2019). This requirement has been challenged multitudes of times and is a truly controversial piece of legislation; it mandates companies to take "reasonable steps" to erase personal data concerning individuals (Bourtule et al, 2019). In October 2014, Google had removed 171,183 links under the right to be forgotten (Kharpal, 2014).

Since these ML models potentially memorize data as they analyze and learn from them, there is a need to develop ways in which models can be sanitized and “re-modelled” based on data that has been deleted. There have been several attempts to develop efficient machine unlearning algorithms, and consequently, there exist quite a few of them (like differential privacy). Due to the intense mandate of these legislations, it becomes essential for organizations and, therefore, developers to verify efficacy. They need a measure of how well their algorithms can forget data. This is primarily where machine unlearning comes in.

Unlearning systems complement existing learning systems, encouraging data propagation and sharing. With the availability of systems that erase data, users will have more control over their data. This will encourage them to share their data. This in-turn helps service providers who will have access to even more data for analysis – data which even if forgotten later, makes no change to the trained ML models. The aforementioned reasons point to easy adoption of forgetting systems as they benefit both users and service providers, along with conforming with upcoming legislation.

2.0 Analysis

In the context of legislation like the *right to be forgotten*, Machine Learning (ML) is increasingly viewed as an exacerbator of breach of privacy. Once data is fed into a ML training model, data can be retained forever, putting customers and users at risk (Synced, 2020).

From researchers' point of view, the problems boils down to this: if a data point is removed from the data space that the model is trained on, is it necessary for the model to be retrained on the dataset without the particular data point? This question arises from the fact that while a model is in perpetual training, it uses existing data as well as new data and changes to the dataset to refine itself (Bertram et al, 2019).

To fully understand the depth of the problem, a motivation for the existence of this report, the report first formalizes the problem. Next, it elucidates the challenges and goals of machine unlearning. Finally, it concludes the analysis with a discussion on verification of machine unlearning, with the prospect of encouraging further research into the issue.

2.1 Formalizing Machine Unlearning

The problem of machine unlearning can be formalized with a scenario consisting of a data collector (A), deletion requester (B) and the environment in which the data exists (C) (Goldwasser et al, 2020). Under normal circumstances, there is communication between all three entities. B and C both send data to A. However, at some point, B requests the deletion of some particular data 'π.' The image below on the left depicts this communication. This is

what happens in the real world. In a more ideal situation, we would like B to be a silent entity that has no communication with A. This encapsulates the idea of the data being completely erased without any trace of it; no lineage either.

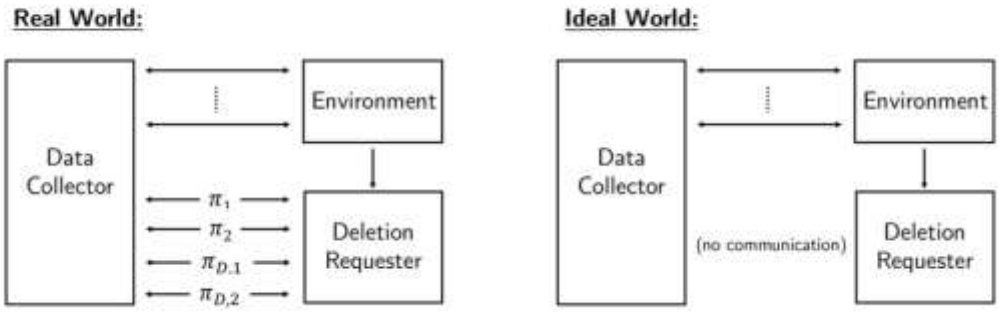

Fig 1: Real and Ideal World Executions

With datapoint 'π' in our dataset (D), the model can be trained. However, if π needs to be deleted, it is impossible to revert our model to its previous state without the datapoint since measuring the effect of one datapoint on a model is not possible. The solution to this problem comes in the form on 'slicing' wherein data dedicated for each model (i.e. each shard) is divided and by incrementally tuning (and storing) the parameter state of a model, we obtain additional time savings.

2.2 Why It's Challenging

i. Impact of one data point on a model:

Computer Science has not yet figured out how to analyze and record the impact that one data point has on a training model. The only way to record this that we know of

is through influence functions. However, these functions are expensive to compute as they involve second order derivatives of the training algorithm (Molnar, 2020).

ii. Random training:

For complex models like Deep Neural Networks, stochastic points are chosen from the datset for each training round (epoch). For each round, more random data is chosen. Oftentimes, training is done in parallel using threads without any means of collation at the end (the randomness method serves training purposes better than most). This sometimes makes backtracking very difficult (Bourtoule et al, 2019).

iii. Training is dependent:

Training models is always incremental, with each update reflecting all previous updates to the model. If a model is trained based on a datapoint, then all future trainings will depend on the datapoint in some implicit way (Bourtoule et al, 2019).

2.3 Goals of Machine Unlearning

The most obvious strategy for unlearning data is to retrain the ML model on the dataset without the datapoint that we are trying to "forget." In corporations, this solution is not feasible due to the large volumes of data they deal with. Moreover, to be in constant compliance with GDPR and other legislations, they will have to retrain models quite frequently. Therefore, strategies need to be developed with the following goals in mind:

i. Accuracy:

If a large fraction of the datapoints in the training set are requested to be forgotten

and hence deleted, then retraining the model would lead to a less accurate trained model. Unlearning algorithms should, therefore, include a bound on accuracy loss in comparison to the initial model

ii. Reduced Training Time:

The unlearning strategy should take considerably less time than the baseline model to unlearn data. This follows from the fact that the number of points to be unlearnt will be lesser than the number of datapoints used in training (Molnar, 2020).

iii. Overheads:

Any new strategy should not introduce additional overheads to the model in terms of complexity: space, time and procedural (Saltzer et al, 1975).

iv. Provable Guarantees:

The unlearning model should guarantee that a certain number of points have indeed been unlearnt and therefore do not influence the succeeding training model.

v. Completeness:

Completeness of unlearning is complemented by an understanding of the consistency of the unlearnt model with another hypothetical model that is trained solely on the dataset without the points that our model is trying to forget. If the unlearning algorithm produces a model identical to the one that is trained on the dataset without the data to be forgotten, then we effectively verify that nobody can access that data or its lineage. This is referred to as a complete model.

vi. Easy Debugging:

Unlearning strategies should be relatively intelligible and should be easy enough to be understood by non-experts so that debugging is facilitated.

2.4 Verification of Machine Unlearning

In general, verification of machine unlearning algorithms is difficult, given the data has already been outsourced. This is usually the case since data is shared among different servers in big corporations – even among different corporations. One way to verify such algorithms involves the membership inference attacks (Shokri et al, 2017). This line of work suffers from some limitations – lack of computational power, low accuracy, etc. The best method to verify machine unlearning models in today's world is by attacking the machine learning model itself. To understand methods of verification, we must first discuss ways to attack the machine learning model – trojans, backdoor, poisoning, etc. (Polyakov, 2019). For the purposes of this report, we look into backdoor attacks. Backdoor attacks refer to any method used to go around security measures to access resources within an application (Becenti, 2019).

One approach to verify Machine Unlearning is outlined below (Sommer et al, 2003).

The verification occurs in two phases:

I. Phase I
    a. Users generate backdoor codes that alter prediction data.
    b. Users apply their backdoor codes to a fraction of their own data in the server and submit it to the MLaaS (Machine Learning as a Service) provider who then train the data.

II. Phase II
    a. Users request deletion of their data. The server then either proceeds with or without retraining the model

b. Users query the model with backdoor samples and based on predictions to check if the server ran the unlearning algorithm successfully.

This is further outlined in the image below.

(a) The first step involves the backdoor injection during model training.

(b) When the server deletes the user's data (H_0), the predictions of backdoor samples are correct labels with high probability.

(c) When the server does not delete the user's data (H_1), the predictions of backdoor samples are target labels with high probability.

Fig 2: Verification using Backdoor Attacks

3.0 Conclusion

The report acknowledges the importance of machine unlearning algorithms due to legislation that binds corporations like the GDPR and PIPEDA which enforce the right to be forgotten. Under this, users can request any data to be completely erased along with its lineage. Since machine learning models train on this data, they end up memorizing and analyzing this data. Future states of these models are, therefore, in some way dependent on previous data. This means that even if data is deleted, its lineage will live on in the model. Implementing Machine Unlearning will also encourage users to share their data more, since they are assured that it can be completely erased on request.

In light of this, the report analyses the definition, goals, challenges and verification of machine unlearning. For developers, the problem boils down to whether ML models need to be retrained every time data is removed from the data space.

The unlearning problem can be formalized as an interaction between three entities – the environment, data collector and data deletion requester. In an ideal world, there would be no communication between the deletion requester and collector, since this would require some lineage of the data to exist. However, in reality, the deletion requester sends deletion requests to the model which then deletes and unlearns the data.

There are several reasons why this is challenging to implement. We can't measure the impact of a single datapoint of a model and its parameters. Furthermore, random data points are chosen for training of models, especially those involving Convoluted Neural Networks and Deep Neural Networks. This makes backtracking difficult. Training of models

is also dependent – the current state depends on previous trainings and on existing data.

With these shortcomings in mind, machine unlearning models should have some criteria to conform with. When data is deleted, models have lesser data to work with. We should make sure the accuracy of the model is not compromised with lesser training data. The new models should take lesser time than baseline models and other existing ones to train on new, shorter datasets. Algorithms should not introduce more overheads and provide guarantees of data deletion. The unlearning should be "complete," meaning the new model should be consistent with a model trained solely on the new dataset. The unlearning models should also be easy to debug and must be of such a nature that even non-experts can understand it.

Since data in a corporation can be outsourced and/or shared among other internal and external servers, verifying data deletion can be a daunting task. One way to do this is through membership inference attacks, but this method requires large computational power and suffers from low accuracy. In the status quo, machine unlearning is best verified by attacks to the model. Several techniques like poisoning, trojans, backdooring, etc. exist for this purpose. Backdooring refers to a way of getting around security measures to access an application or resource's data. The report discusses one method to verify machine unlearning models using backdooring wherein users generate backdoor codes that alter initial data predicted by a model. These codes are applied to users' data on which the model is trained again. The model either proceeds without considering the change or it considers it. The users then query the backdoored data and check whether the algorithm ran successfully.

Researchers should be motivated to conform to the ideal goals of machine unlearning and keep these in mind while developing novel algorithms that will promote data sharing.

Acknowledgements

This report was inspired by a training exercise provided by SS&C Technologies during which the policies of the General Data Protection Rights (GDPR) were outlined. Since many ML models are used to manage data and train on it, I was curious as to how data that is used in ML training can be fully erased from existence. This report was created through thorough research using many recent and slightly older research papers into machine learning, unlearning, backdooring and probabilistic verification.

References

1. *Y. Liu, K. K. Gadepalli, M. Norouzi, G. Dahl, T. Kohlberger, S. Venugopalan, A. S. Boyko, A. Timofeev, P. Q. Nelson, G. Corrado, J. Hipp, L. Peng, and M. Stumpe, "Detecting cancer metastases on gigapixel pathology images," arXiv, Tech. Rep., 2017*

   From: https://arxiv.org/abs/1703.02442

2. *S. Shastri, M. Wasserman, and V. Chidambaram, "The seven sins of personal-data processing systems under gdpr," USENIX HotCloud, 2019*

3. *L. Bourtoule, V. Chandrasekaran, C. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine Unlearning," arXiv, Tech. Rep., 2019.*

   From: arXiv:1912.03817

4. *Y. Cao, and J. Yang, "Towards Making Systems Forget with Machine Unlearning," Columbia University, 2017*

   From: http://www.cs.columbia.edu/~junfeng/papers/unlearning-sp15.pdf

5. *A. Kharpal, "Google axes 170,000 'right to be forgotten'", 2014*

   From: http: //www.cnbc.com/id/102082044.

6. *Synced, "Machine Unlearning: Fighting for the Right to be Forgotten", 2020*

   From: https://medium.com/syncedreview/machine-unlearning-fighting-for-the-right-to-be-forgotten-c381f8a4acf5

7. *T. Bertram, E. Bursztein, S. Caro, H. Chao, R. C. Feman, P. Fleischer, A. Gustafsson, J. Hemerly, C. Hibbert, L. Invernizzi, L. K. Donnelly, J. Ketover, J. Laefer, P. Nicholas, Y. Niu, H. Obhi, D. Price, A. Strait, K. Thomas, and A. Verney, "Five years of the right to*

be forgotten," in Proceedings of the Conference on Computer and Communications Security, 2019.

From: https://research.google/pubs/pub48483.pdf/

8. *S. Garg, S. Goldwasser, P.N. Vasudevan, "Formalizing Data Deletion in the Context of the Right to be Forgotten," 2020.*

   From: https://eprint.iacr.org/2020/254.pdf

9. *C. Molnar, "Interpretable Machine Learning," 2020*

   From: https://christophm.github.io/interpretable-ml-book/

10. *J. H. Saltzer and M. D. Schroeder, "The protection of information in computer systems," Proceedings of the IEEE, vol. 63, no. 9, pp. 1278– 1308, 1975.*

    From: https://web.mit.edu/Saltzer/www/publications/protection/

11. *R. Shokri, M. Stronati, C. Song, V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," 2017*

    From: https://www.cs.cornell.edu/~shmat/shmat_oak17.pdf

12. *A. Polyakov, "How To Attack Machine Learning," 2019.*

    From: https://towardsdatascience.com/how-to-attack-machine-learning-evasion-poisoning-inference-trojans-backdoors-a7cb5832595c

13. *M. Becenti, "What Is A Backdoor Attack?" 2019.*

    From: https://www.sitelock.com/blog/what-is-a-backdoor-attack/

Image Sources

1. *Fig 1: Real and Ideal World Executions*

   From: https://eprint.iacr.org/2020/254.pdf

2. *Fig 2: Verification using Backdoor Attacks*

   From: https://arxiv.org/pdf/2003.04247v1.pdf