In the Name of God

Introduction to Machine Learning (25737-2)

Project Phase 0

Spring Semester 1402-03

Department of Electrical Engineering

Sharif University of Technology

Instructor: Dr.R.Amiri

Due on Khordad 11, 1403 at 23:55

All you need is Privacy!

1 Introduction

As we navigate through the vast ocean of data in this digital age, the power of machine learning and deep learning models has become increasingly evident. These models have revolutionized data analysis, providing us with insights that were previously unimaginable. However, with great power comes great responsibility. As we delve deeper into the world of data, one question becomes increasingly important: How do we ensure the privacy of the data we analyze?

This project aims to address this critical issue. We will explore various aspects of privacy in machine learning, focusing on two key areas.

In the first section, we will introduce you to the concept of **Machine Unlearning** , a process designed to uphold the **right to be forgotten**. This right is a crucial aspect of data privacy, ensuring that individuals can request their data to be removed from a model or system.

In the final section, we will delve into the realm of **training private models**. We will also discuss a specific adversarial attack known as the **Membership Inference Attack**. This attack poses a significant threat to data privacy, and understanding it is key to developing robust, privacy-preserving machine learning models.

By the end of this project, you will have a solid understanding of these important concepts and be well-equipped to navigate the complex landscape of privacy in machine learning. Let's start this journey together!

2 Machine Unlearning

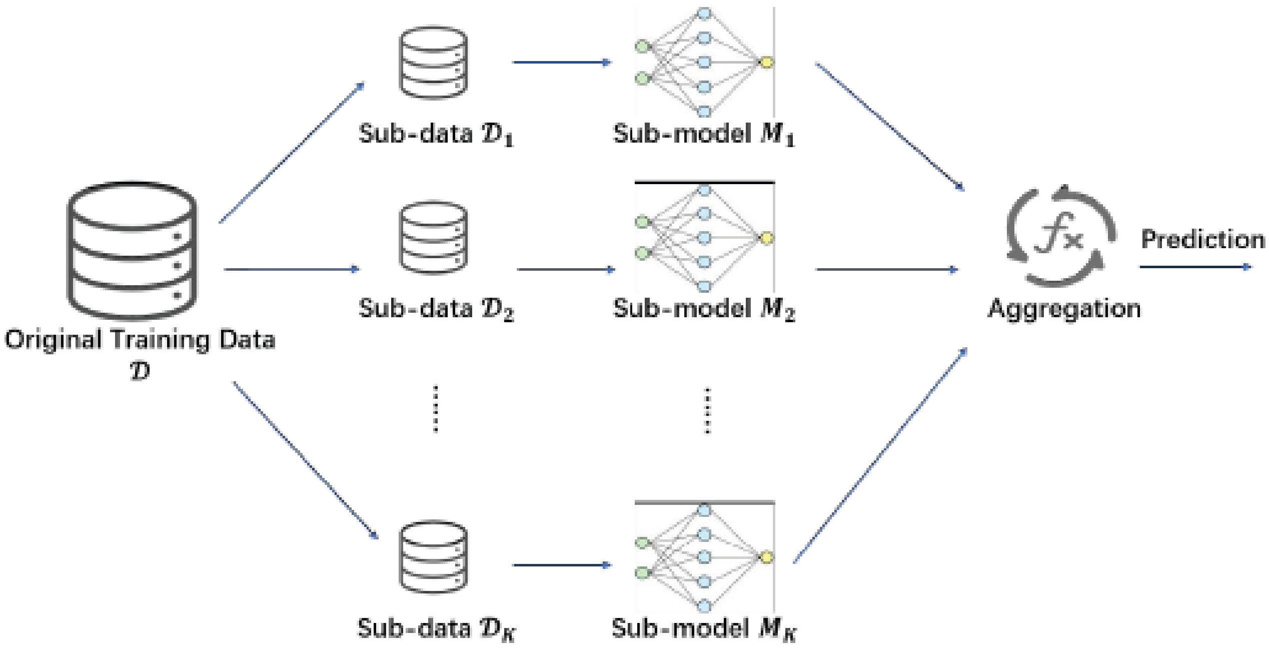
Machine unlearning refers to the ability of a machine learning model to effectively forget or remove the influence of specific data points it has learned from, without the need to retrain the entire model from scratch. This capability is increasingly important due to several key reasons:

- **Privacy and Data Protection:** Regulations like the General Data Protection Regulation (GDPR) in Europe mandate that individuals have the right to request the deletion of their personal data. Machine unlearning helps in complying with such regulations by ensuring that a model can forget specific user data upon request.
- **Error Correction:** Sometimes, data used for training might contain errors or become irrelevant over time. Machine unlearning allows for the correction of these errors without the need for complete retraining.
- **Data Management:** In dynamic environments where data changes frequently, the ability to unlearn outdated or less relevant information can help maintain model performance and relevance.

SISA (Sharded, Isolated, Sliced, and Aggregated) Algorithm
The SISA algorithm is a structured approach designed to facilitate efficient machine unlearning. Here's a breakdown of how SISA works:

- **Sharding:** The training data is divided into multiple smaller subsets, or shards. Each shard contains a portion of the overall data. This ensures that the model training can be segmented into manageable parts.
- **Isolation:** Each shard is used to train a separate model or a component of a model independently. This isolation helps in minimizing the dependency across different parts of the training data.
- **Slicing:** Each shard is further divided into slices. Training occurs in a sequential manner, slice by slice, within each shard. This stepwise learning approach allows for finer control over the data's influence on the model. Specifically, each shard's data D_k is further uniformly partitioned into R disjoint slices such that $\bigcap_{i\in[R]} D_{k,i} = \varnothing$ and $\bigcup_{i\in[R]} D_{k,i} = D_k$. We perform training for e epochs to obtain M_k as follows:
  - At step 1, train the model using random initialization using only $D_{k,1}$, for e_1 epochs. Let us refer to the resulting model as $M_{k,1}$. Save the state of parameters associated with this model.
  - At step 2, train the model $M_{k,1}$ using $D_{k,1} \bigcup D_{k,2}$, for e_2 epochs. Let us refer to the resulting model as $M_{k,2}$. Save the parameter state.
  - At step R, train the model $M_{k,R-1}$ using $\bigcup_i D_{k,i}$ for e_R epochs. Let us refer to the resulting final model as $M_{k,R} = M_k$. Save the parameter state.
- **Aggregation:** The final model is constructed by aggregating the outputs from the independently trained shards. This aggregation helps in forming a comprehensive model while maintaining the benefits of isolation and segmentation.

The general shematic of SISA algorithm is like below:



Unlearning with SISA When unlearning is required, the SISA algorithm allows for the selective removal of data by:

- **Targeting Specific Shards:** Since the training data is sharded, only the shards containing the data to be forgotten need to be retrained or adjusted. This significantly reduces the computational cost compared to retraining the entire model.
- **Updating Relevant Slices:** Within the targeted shards, only the specific slices containing the data to be forgotten are modified. This granular approach ensures that changes are localized, preserving the integrity of the rest of the model.
- **Efficient Re-Aggregation:** After updating the necessary shards and slices, the outputs are re-aggregated to form the updated model. This allows the model to effectively "forget" the specified data points while retaining the knowledge from the remaining data.

The SISA algorithm thus provides an efficient and scalable method for machine unlearning, making it feasible to comply with privacy regulations and manage data integrity without the need for exhaustive retraining.

2.1 Questions

Theory Question 1. Name 3 aggregation methods you would propose for this algorithm. To your best knowledge, reason each method's strengths and weaknesses.

Theory Question 2. When will this method be inefficient to use? What do you suggest to mitigate this issue?

Theory Question 3. What metric would you use to evaluate the performance of your unlearning algorithm? Reason your choice!

Theory Question 4. Discuss the effect of the models' architecture on learning and unlearning time, and performance in both learning and unlearning.

3 Private Training Models

In the realm of machine learning, private training models are crucial for safeguarding sensitive data during the model training process. These models are designed with privacy-preserving techniques that prevent the exposure of individual data records to unauthorized parties. The necessity for private training models arises from the threat of **membership inference attacks** , where adversaries attempt to deduce whether a particular data record was used in training a machine learning model. By employing private training models, organizations can enhance the security of their machine learning systems, ensuring that the confidentiality of the training data is maintained and the privacy of individuals is protected against such invasive attacks. This is especially vital when dealing with sensitive information, such as medical records or personal financial data, where the implications of a privacy breach can be significant.

Here we explain some techniques for private training models :

- **Differentially Private Training** : This technique adds noise to the training process to obscure the contribution of individual data points, thus enhancing privacy. It ensures that the removal or addition of a single data record does not significantly affect the output of the model, providing a quantifiable level of privacy.
- **Regularization**: Regularization methods like L1 or L2 norms are used during training to prevent overfitting. By penalizing the magnitude of the model parameters, regularization makes it harder for attackers to infer whether a specific data point was used in the training set.
- **Normalization Temperature** : Increasing the normalization temperature involves adjusting the softmax function used in classification. A higher temperature leads to a smoother probability distribution over classes, which can reduce the model's sensitivity to individual data points and thus improve privacy.
- **Adding Noise for Privacy**: The concept of adding noise to a dataset involves intentionally introducing some level of randomness to the data before it is used to train a machine learning model. This technique is often employed as a privacy-preserving measure to protect individual data records from being identified or reconstructed. By altering the original data slightly, the risk of sensitive information being exposed through the model's predictions is reduced. However, it's crucial to balance the amount of noise added; too much can degrade the model's performance, while too little may not provide sufficient privacy protection. Effective noise addition can help maintain a model's utility while safeguarding the privacy of the individuals represented in the training dataset.

These techniques, among others, contribute to the robustness of machine learning models against membership inference attacks by reducing the risk of information leakage about the training data. Each method offers a different approach to enhancing privacy while maintaining the utility of the model.

Theory Question 5. Explain differentially private algorithms and their techniques for training a differentially private model.

Theory Question 6. Explain the regularization and normalization techniques used in training a private model. Are these techniques similar to the method of adding noise to the model in differential privacy?

Theory Question 7. Find other techniques for training a private model.

4 Membership Inference Attack

4.1 Introduction

Membership inference attacks are a type of privacy attack against machine learning models. The goal of these attacks is to determine whether a particular data record was used in the training dataset of a machine learning model. This can be problematic, especially when the training data contains sensitive information.

4.2 Types of Membership Inference Attacks

There are three main types of membership inference attacks, categorized based on the level of access the attacker has to the target model:

4.2.1 White-Box Attack

In a white-box attack, the attacker has complete knowledge of the target model, including its architecture, parameters, and training algorithm. This allows for a more precise attack but requires a high level of access that may not always be available.

4.2.2 Gray-Box Attack

A gray-box attack assumes partial knowledge of the target model. The attacker may know some details about the model's architecture or training data but does not have full access to the model's parameters.

4.2.3 Black-Box Attack

In a black-box attack, the attacker has no knowledge of the target model's internals and can only interact with it through a public API. This is the most common scenario and the focus of many research studies on membership inference attacks.

4.3 Paper's Method

We address the membership inference problem which is determining if a record was part of a machine learning model's training dataset. This problem is investigated in a challenging setting where the adversary only has black-box access to the model. We quantify membership information leakage via the model's prediction outputs. To solve this, we train an attack model to differentiate the target model's behavior on training inputs from its behavior on unseen inputs, effectively transforming the membership inference problem into a classification problem (see figure 1). Attacking black-box models such as those built by commercial "machine learning as a service" providers requires more sophistication than attacking white-box models whose structure and parameters are known to the adversary.

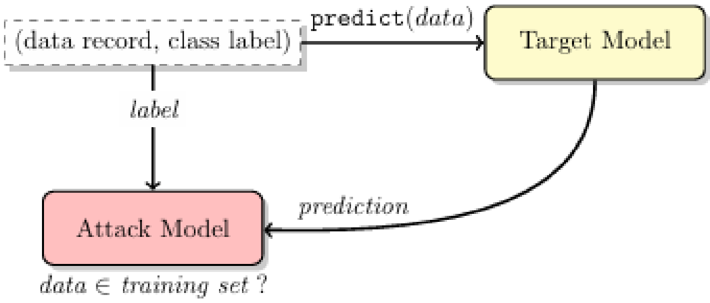


Figure 1: The attacker queries the target model with a data record and obtains the model's prediction on that record. The prediction is a vector of probabilities, one per class, that the record belongs to a certain class. This prediction vector, along with the label of the target record, is passed to the attack model, which infers whether the record was in or out of the target model's training dataset.

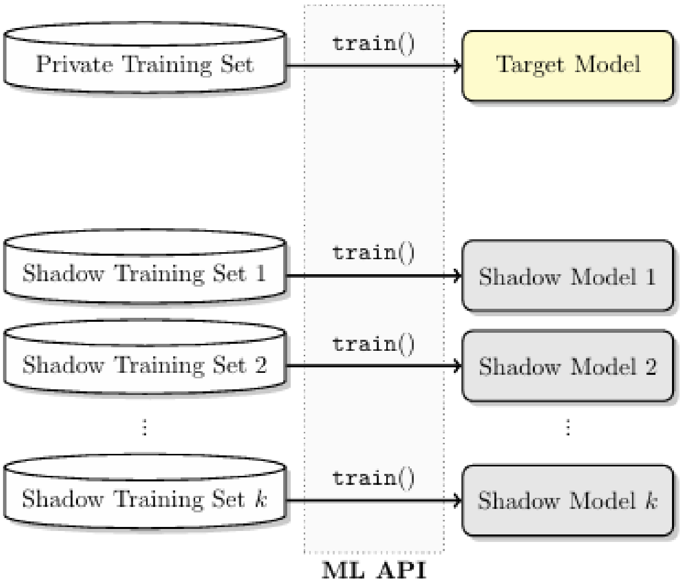


Figure 2: Training shadow models using the same machine learning platform as was used to train the target model. The training datasets of the target and shadow models have the same format but are disjoint. The training datasets of the shadow models may overlap. All models' internal parameters are trained independently.

To devise our attack models, we employ a novel shadow training method. Initially, we generate several "shadow models" that mimic the target model's behavior. For these models, we have knowledge of the training datasets and hence, the ground truth about dataset membership (see figure 2). Subsequently, the attack model is trained using the labeled inputs and outputs of these shadow models.

4.4 Questions

Theory Question 8. Explain three ways of generating training data for shadow models.

Theory Question 9. One of the method for generating training data for shadow models is using the model to generate synthetic data. Explain the Algorithm of synthetic data generation.

Theory Question 10. Explain how attack model is trained using shadow models.

Theory Question 11. Explain the effect of following concepts in accuracy of the attack model:

1. Effect of the shadow training data generated using the three methods you explained earlier.
2. Effect of the number of classes and training data per class.
3. Effect of overfitting.