In the Name of God

Introduction to Machine Learning (25737-2)

Project Phase 1

Spring Semester 1402-03

Department of Electrical Engineering

Sharif University of Technology

Instructor: Dr.R.Amiri

Due on Tir 10, 1403 at 23:59

(∗) Should you have any questions concerning the project, please feel free to ask via Telegram.

Abstract

In the previous phase you familiarized yourself with some important topics in machine learning. In this phase you will do the implementations!

1 Machine Unlearning

In this question, you will be implementing SISA Algorithm and testing its performance.

1.1 Learning Phase

- **Sharding:** The training data is divided into S 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.

Simulation Question 1. Implement the learning phase of the algorithm, and select the based models to be pretrained **ResNet 18**(you will need to add some fully connected layer(s) in order to make it suitable for the given dataset!). Use $S = 5, 10, 20$ and $R = 5, 10, 20$ and train different models on your data in each case, and report F1-score, accuracy, precision, recall and AUROC of all of your models with all of your proposed aggregation methods.

1.2 Unlearning Phase

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 a targeted shard, k that is supposed to have some of its data removed, assume $M_{k,j}$ is the last model of the shard that all of its training data is not supposed to be removed. We then take a similar approach to the learning phase as we train $M_{k,j}$ using $\bigcup_{i \leq j+1} D'_{k,i}$ for e_R epochs, and so on like in the learning phase.($D'_{k,i}$'s are updated data slices with the specified data removed)
- **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.

Simulation Question 2. Implement the unlearning algorithm on all your trained models in the previous question and ”forget” 500 randomly chosen data of the whole dataset and measure the performance using your proposed metric, also report F1-score, accuracy, precision, recall and AUROC of all of your models with all of your proposed aggregation methods.

1.3 Evaluation

In this part we evaluate the effectiveness of the SISA unlearning algorithm, using Membership Inference Attack.

1.3.1 Membership Inference Attack

A Membership Inference Attack aims to determine whether a particular data sample was part of the training dataset used to train a machine learning model. This type of attack leverages the model's responses to specific inputs, exploiting differences in the model's behavior between training data and unseen data. For instance, models often perform better on training data compared to unfamiliar data due to overfitting. By carefully analyzing the model's output probabilities, loss values, or other response characteristics, an attacker can infer the membership status of individual samples, potentially leading to privacy breaches and revealing sensitive information about the training data.

Simulation Question 3. Write a function that takes losses of two different datasets of your trained(but not unlearned) model,Computes cross-validation score of a Logistic Regression-based Membership Inference Attack. Now, input the function with the losses of the 'forget set' and 500 randomly chosen data of the test set of the trained model and report the score. Do the same for the unlearned model. What did you expect the result to be for a perfectly unlearned model? How did the SISA Algorithm perform?

1.4 Add On, Evaluation

To evaluate the effectiveness of the SISA unlearning algorithm, we will perform a backdoor attack on the model and assess the model's performance before and after unlearning.

1.4.1 Backdoor Attack

A backdoor attack involves poisoning the training data so that the model learns to associate a specific trigger (e.g., a pattern or mark) with a target label. During inference, the presence of this trigger in any input data will cause the model to misclassify it as the target label.

1.4.2 Assessing the Attack

- Evaluate the model on clean test data to ensure general performance is not significantly degraded.
- Evaluate the model on test data containing the backdoor trigger to measure the success rate of the backdoor attack.

1.4.3 Measuring the Success of the Backdoor Attack

The success of the backdoor attack can be measured by the attack success rate (ASR), which is the percentage of test samples containing the backdoor trigger that are incorrectly classified as the target class. High ASR indicates a successful attack.

Add On. Simulation Question 1. Randomly select 500 data points from a specific class in the training set. For each selected data point, turn a 3x3 block of pixels to black, with the position of the block chosen randomly. Train your best model of the previous questions, using the poisoned dataset. Evaluate the model's performance, using all the same metrics as before, on clean test data to ensure general performance is not significantly degraded. Also, calculate the attack success rate (ASR) as the percentage of test samples misclassified as the target class.

Add On. Simulation Question 2. Unlearn the same 500 data, and evaluate the model's performance, using all the same metrics ass before, on clean test data to ensure general performance is not significantly degraded. Also, calculate the attack success rate (ASR) as the percentage of test samples misclassified as the target class.

2 Private Training

In this section, you will first train a classification model using a standard approach, then train it with privacy enhancements, and compare the **MIA** accuracy of both models. (Use the provided model in model.py for all tasks.)

Simulation Question 4. Use 80 percent of the CIFAR-10 training data to train your model. This will serve as your baseline model.

Simulation Question 5. Train your baseline model with privacy enhancements. This is your modified model. Ensure that the test accuracy difference between your baseline model and the modified model is less than 15

Simulation Question 6. Train two **Attacker Models** based on **MIA** techniques learned in Phase 0, one for the baseline model and one for the modified model. Compare the MIA accuracy of these two attacker models. Use 80 percent of the training data as your **seen data**, and the remaining training data along with the test data as your **unseen data**.

Simulation Question 7. Improve your attacker models to achieve better MIA accuracy for both the baseline and modified models (e.g., by increasing the number of shadow models). Then compare the new accuracies with the previous results.

3 Membership Inference Attack

Simulation Question 8. Attempt to train an attacker model for the given private model (private_model.pth). We will test it on our dataset during the online presentation session. A competitive bonus point is available for the best performance.