A Repository of Jupyter Notebooks on Unlearning in Federated Learning

Final Presentation
FYP22002 @ HKUCS
Sheng “Victor” HUANG
Supervisor: Prof. S M YIU

20 Apr 2023
Background

• Example
  • ChatGPT -> what data is used?
    • Online public text databases, 570GB, 300 billion words [BBC23]
    • Also user conversations [OpenAI23]

6. Will you use my conversations for training?
  • Yes. Your conversations may be reviewed by our AI trainers to improve our systems.

Background

Zero-Shot Information Extraction via Chatting with ChatGPT

• Security?

Xiang Wei¹, Xingyu Cui¹, Ning Cheng¹, Xiaobin Wang², Xin Zhang, Shen Huang², Pengjun Xie², Jinan Xu¹, Yufeng Chen¹, Meishan Zhang, Yong Jiang², and Wenjuan Han¹

¹ Beijing Jiaotong University, Beijing, China
² DAMO Academy, Alibaba Group, China

Exploring the Feasibility of ChatGPT for Event Extraction

Jun Gao¹* Huan Zhao² Changlong Yu³ Ruifeng Xu¹

¹Harbin Institute of Technology (Shenzhen)
²4Paradigm. Inc.
³HKUST, Hong Kong, China
imgaojun@gmail.com zhaohuan@4paradigm.com cyuaq@cse.ust.hk

Extracting Accurate Materials Data from Research Papers with Conversational Language Models and Prompt Engineering - Example of ChatGPT

Maciej P. Polak* and Dane Morgan¹

Department of Materials Science and Engineering,
University of Wisconsin-Madison, Madison, Wisconsin 53706-1595, USA
Background

• Privacy?

Respecting privacy

Our large language models are trained on a broad corpus of text that includes publicly available content, licensed content, and content generated by human reviewers. We don’t use data for selling our services, advertising, or building profiles of people—we use data to make our models more helpful for people. ChatGPT, for instance, improves by further training on the conversations people have with it.

While some of our training data includes personal information that is available on the public internet, we want our models to learn about the world, not private individuals. So we work to remove personal information from the training dataset where feasible, fine-tune models to reject requests for personal information of private individuals, and respond to requests from individuals to delete their personal information from our systems. These steps minimize the possibility that our models might generate responses that include the personal information of private individuals.
Background

• Usability & Fidelity?

4. Can I trust that the AI is telling me the truth?

• ChatGPT is not connected to the internet, and it can occasionally produce incorrect answers. It has limited knowledge of world and events after 2021 and may also occasionally produce harmful instructions or biased content.

We'd recommend checking whether responses from the model are accurate or not. If you find an answer is incorrect, please provide that feedback by using the "Thumbs Down" button.
Background

• Inappropriate use of data -> delete data
  • Right to be Forgotten
  • OpenAI based in San Francisco, CA

How can I delete my account?

I'd like to delete my account, but I can't find any way to do this.

Written by Johanna C. Updated yesterday

Account Deletion (Two Methods)

⚠️ UPDATE: As of April 11, 2023 ChatGPT users on either free or plus plans can delete their own accounts themselves:
Background

- Is that enough?
  - Data deleted from database may still exist in ChatGPT’s models
  - May still create problems of security, privacy, usability or fidelity
Machine Unlearning [CY15]

• Remove data and data influence from ML models
• ML models as black boxes [KL17]
• Naïve method -> to retrain on remaining data -> slow
  • Exact unlearning, only with variations of retraining
  • Otherwise approximate unlearning
• Challenges:
  • Stochasticity of training
    • Data influence difficult to track
  • Incrementality of training
    • Training with a data point affected by prior training, and will affect later training
  • Catastrophic unlearning
    • Unlearnt models generally perform worse than retrained models

[KL17] https://proceedings.mlr.press/v70/koh17a
Federated Learning (FL) [McM+17]

- Multiple devices to collaboratively train a model without sharing data
- Has potential application with ChatGPT
  - E.g. Healthcare
- Not immune from privacy vulnerabilities present in other ML techniques
  - Unlearning is still needed

Federated Unlearning

Additional Challenges

• Limited data access
  • Server doesn’t have access to training data used at the client side

• Limited client participation
  • Client goes offline and can’t participate in unlearning process

• Complicated relationship between data and clients
  • Unlearning all data of one client of just part of it?
  • What if there is data overlap on other clients?

• Data partition
  • Horizontal FL, vertical FL and federated transfer learning

• Statistical heterogeneity
  • IID (Independent and Identically Distributed)

• Adversarial model
  • Untruthful server/clients

• …
How to Study Federated Unlearning?

Introducing:
A Repository of Jupyter Notebooks on Unlearning in Federated Learning

Objective:
To organize knowledge and study unlearning in FL
Methods

• Literature Review
  • Google Scholar “federated+unlearning”
    • total 17 papers on federated unlearning, 2021-2023
    • Several papers on unlearning in general and other privacy-preserving techniques are also reviewed for context

• Jupyter Notebooks
  • Summarizing and referencing key content from reviewed papers

• Experiments
  • Adapting code from respective papers
  • testing in different environments – HKUCS GPU Farm & Google Colab
  • Fixing bugs and issues
  • Record instructions in notebooks

• Repo and website hosting
  • Git, GitHub, Hugo, GitHub Pages, GitHub Actions
## Results

### Description

This repo aims to provide materials, in the form of Jupyter Notebooks, for studying machine unlearning in federated learning (FL), both of which are new research areas that are developing rapidly.

Note The data collection ended in early April 2023 and I have done my best to scrape the internet on this topic. If you are checking out this repo in the future, you may find it incomplete. However, it is helpful that we have this snapshot, so that future researchers can see how this new area grew and take inspirations from earlier research.

The contents of the notebooks consist of:
1. general introduction to the area and its background (0-2);
2. research progress in 2021 (3-5);
3. research progress in 2022 (6-15);
4. research progress in 2023 (until early April) (16-19).

The notebooks, except those in the 1st part, are ordered according to the publication dates (or conference dates, or last edit date of preprints) of the papers.

Marked notebooks are summaries of the paper with key concepts, figures, and tables presented. Marked notebooks are instructions on how to run the code for the design algorithms, with tweaks made to get rid of bugs that usually exist in the original code. Clicking on or can open the corresponding notebook in Google Colab.

### Recommended prerequisites

- Machine Learning
- Computer Security
- Operating Systems

### Update

- **1st Presentation**
- Final Presentation

### Project Plan | Interim Report | Final Report

Made with ❤️ by [Victor](#).
Results

Machine Unlearning

This notebook gives a brief introduction to machine unlearning.

Background

Today, we put a lot of data into artificial intelligence (AI), especially for the training of machine-learning (ML) models. Some of the data, though, can be of personal and private nature, such as your location, biomarkers and medical records. It is natural that we want our data to be used and stored securely.

At the same time, ML systems are also prone to attacks like other types of systems, such as membership inference, property inference, model stealing, data poisoning, etc. These attacks could lead to problems such as data leakage and wrong predictions. Or, in some cases such as poor data quality or system design, these problems can still occur even without an adversarial third-party.

If you know something about ML, you’ll know that ML models can sometimes “memorize” instead of “learn from” data, often as a result of overfitting. And that’s dangerous because it makes it so much easier to run model extraction attacks against such a model and data may not be the best. Thus, it makes sense for us to want to delete some of the data, or erase some of the training, from our trained model.

Deleting data can be as easy as a single SQL execution in some systems, but it’s not that straightforward in ML models. In an ML model, the connection between parameters and data is not clearly shown. Hence we are unable to map a single data point to influence throughout the training process. Thus, it is difficult to remove information relating to a single data point from a trained ML model. In other words, it can be said that it is challenging to induce targeted “memory loss” in ML models, or to make ML models “forget” that it was trained with specific data.

Definition

The process of deleting certain data points from ML systems is called machine unlearning, or simply unlearning, which is a term first proposed by Cas and Ming in 2015.

As an obvious solution, we can retain an ML model from scratch, using all the data points but the ones we want to erase. This will ensure that the targeted data points are not in the ML system, but this process is very slow. So, ideally an unlearning paradigm should be reasonably efficient.

We will talk about more aspects of unlearning definitions in a later notebook.

Federated Learning (FL)

This notebook gives you a brief introduction to FL.

Background

Similar to machine unlearning, FL also tries to solve privacy and security problems arising from the fact that a large amount of data is used in ML. You may check the [Introduction to Federated Learning](https://cs231n.stanford.edu/papers/FedAvg.pdf) notebook for detailed implications of data security and privacy in ML systems. However, takes a different approach to achieve privacy and security, which means there are some overlapping issues in the cross section between FL and unlearning (this is the main topic of this project).

For now, let’s keep focusing on FL.

In traditional settings, data is stored and processed at a centralized location in a centralized manner. All this centralization used with data can lead to security and privacy problems. Despite efforts to mitigate such problems in centralized settings, decentralized methods have been introduced to provide an alternative pathway towards better security and privacy protection. However, as one can clearly see from data collection and processing capabilities on individual decentralized machine clients may not be adequate to train an ML model with good enough performance, there probably needs to be a mechanism for orchestrating multiple clients. FL is one such method.

In a typical FL setting there are, of course, many variations, an ML model is initialized at a central server and then sent to participating clients. The clients train the model with the data that they have, generating an update on the model (sub-update), and then send the sub-update to the server. The server then pushes the received sub-updates to update (usually by aggregating) its own model. The updated central model is then sent to clients for further training.
Machine Unlearning by AnnesiaML

This Notebook is adapted from Annesia: Machine Learning.
You are strongly advised to run this notebook with CUDA-enabled GPU on your machine or with a GPU runtime on Colab, due to a large amount of tensor calculation.

Imports and Setup

```python
import torch
import torchvision
import torchvision.transforms as transforms
import torch.nn as nn
import torch.optim as optim
import torchvision.transforms.functional as F
import torchvision.transforms as transforms
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

data_transform = transforms.Compose([transforms.Resize((32,32)), transforms.ToTensor(), transforms.Normalize((0.5,0.5,0.5),(0.5,0.5,0.5))])
```

Data Entry and Processing

```python
# Transform image to tensor and normalize features from [0.0,255] to [0,1]
transform = transforms.Compose([transforms.Resize((32,32)), transforms.ToTensor(), transforms.Normalize((0.5,0.5,0.5),(0.5,0.5,0.5))])
```

Federated Learning with Flower (\(^{\text{\textcopyright}}\))

Flower (\(^{\text{\textcopyright}}\)) is a framework for building federated learning systems. The design of Flower is based on a few guiding principles:

- **Customizable**: Federated learning systems vary wildly from one use case to another. Flower allows for a wide range of different configurations depending on the needs of each individual user case.
- **Extensible**: Flower originated from a research project at the University of Oxford, so it was built with AI research in mind. Many components can be extended and overridden to build new state-of-the-art systems.
- **Framework-agnostic**: Different machine learning frameworks have different strengths. Flower can be used with any machine learning framework, for example, PyTorch, TensorFlow, and even Transforms, PyTorch Lightning, BERT, and more. Flower is fully extensible, and we encourage the community to contribute to the codebase.

Flower with TensorFlow

To start with, let's try to build a federated learning system using Flower and TensorFlow.

Install required packages

1. Check out Python versions supporting both TensorFlow and Flower. (Flower's and TensorFlow's requirements)
2. Install the virtual environment using `conda` or `virtualenv` or other tools. Let's say we use Anaconda.
3. Create a virtual environment like `myenv` using `conda activate`.
4. Install the required packages.

   ```bash
   ![](https://blog.tensorflow.org/)
   ```

   ```bash
   ![](https://blog.tensorflow.org/)
   ```

   ```bash
   ![](https://blog.tensorflow.org/)
   ```

   ```bash
   ![](https://blog.tensorflow.org/)
   ```

   ```bash
   ![](https://blog.tensorflow.org/)
   ```

   ```bash
   ![](https://blog.tensorflow.org/)
   ```

   ```bash
   ![](https://blog.tensorflow.org/)
   ```
Significance

• There lacks work on a comprehensive overview of unlearning in FL

• This project is more accessible than academic survey papers
  • Researchers and practitioners can both benefit

• Provides clear information and instructions to guide the study from background knowledge to advanced research frontier.
  • Seasoned researchers can also find specific materials more quickly instead of going through the whole internet

• Promotes privacy and security in ML research and applications
Limitations

• Project stops in April 2023
  • Area still in early stages
    • Could be outdated soon, but good to have this snapshot

• Most papers have very narrow scope
  • E.g. focus on specific models, data partition, etc.
  • Difficult to generalize to the broader FL realm

• Few papers have published code
  • Difficult to recreate and compare quantitatively with other methods

• Paper quality varies
  • Some have impractical designs, e.g. backdoor attacks for verification [WZM22]

Some Takeaways

• FL has many variations. It is difficult to find a universal unlearning method that performs well.
  • One of the goals of research in this area is to find an unlearning method that requires little change to the existing FL designs, while achieving good performance.

• Unlearning itself may lead to unintended privacy risks [Che+21].
  • In practice, unlearning should be used along with other privacy-preserving methods.

• As of this moment, unlearning has not been practical in most ML scenarios, so other more practical measures should be given priority in real-world applications.

[Che+21] https://dl.acm.org/doi/abs/10.1145/3460120.3484756
Some Takeaways

• There has not been consensus on the differentiation of unlearning and data deletion in ML. Some have argued that the narrowly-defined unlearning is unlikely to achieve stricter privacy goals [CS23].

• FL is still in its early stages and not a practical privacy-enhancing technology (PET) yet [Boe+23], so there will be a lot of federated learning and unlearning research work done in the future.

• Some trends and open questions on unlearning: defining success of unlearning (verification / data auditing); unified unlearning requirements; unified unlearning benchmarking; adversarial machine unlearning; interpretable machine unlearning; ...

Thank you!

• Repo at https://github.com/vicw0ng-hk/feul
• Website at https://vicw0ng-hk.github.io/feul