ML Seminar | SUT | Fall 1403 | Maryam Rezaee

Model Sparsity Can Simplify Machine Unlearning

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, Sijia Liu

37th Conference on Neural Information Processing Systems (2023)

TABLE OF CONTENTS

01

Introduction

1.1 Overview

- What Is MU
- Why Is MU Needed
- Why Not Retrain

1.2 Problem

- Driving Question
- Problem Setup
- Problem Definition

1.1 OVERVIEW

What Is Machine Unlearning (MU)?

- Reverse learning to remove influence of specific examples from an already trained model

Why Is MU Needed?

- Recent data regulation requirements e.g., privacy of data A.K.A. "the right to be forgotten", corrupted data, etc.

Why Is Retraining Not Enough?

- Direct and optimal but unreasonable computational costs
- Approximate and fast but effective methods required

1.2 PROBLEM

Driving Question (Q)

(Q) Is there a theoretically-grounded and broadly-applicable method to improve approximate unlearning across different unlearning criteria?

1.2 PROBLEM

Problem Setup

- Training dataset of N points (with labels): $\mathcal{D} = \{x_i\}_{i=1}^{N}$
- **F**orgetting dataset (to be scrubbed): $\mathcal{D}_f \subseteq \mathcal{D}$
  ↳ $\mathcal{D}_f$ can be *class-wise* or *random data* forgetting
- **R**emaining dataset (for new model): $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$
- **O**riginal model parameters (on $\mathcal{D}$): θ_o
- **U**nlearned model parameters (on $\mathcal{D}_r$): θ_u

Problem Definition

- Generate θ_u from θ_o accurately and efficiently

02

Related Work

2.1 Unlearning

- Approximate MU
- Evaluation of MU
- Other Paradigms

2.2 Pruning

- Performance Impact
- Pruning in MU

2.1 UNLEARNING

Approximate MU Methods

- **Fine-tuning (FT):** fine-tunes θ_o on $\mathcal{D}_r$ for a few epochs to obtain θ_u and causes "catastrophic forgetting" as in continual learning
- **Gradient ascent (GA):** reverses training on $\mathcal{D}_f$ by adding the gradient back to θ_o to increase loss on $\mathcal{D}_f$ and obtain θ_u
- **Fisher forgetting (FF):** adds Gaussian noise to θ_o to perturb dependency on $\mathcal{D}_f$ based on Fisher Information Matrix to obtain θ_u
- **Influence unlearning (IU):** leverages influence function to find change in θ_o if $\mathcal{D}_f$ is removed and subtracts $\mathcal{D}_f$ impact to obtain θ_u

2.1 UNLEARNING

- **More on influence unlearning (IU):** relates to an important line of research in MU ($\varepsilon - \delta$ forgetting); defined as:

$\Delta(\mathbf{w}) \coloneqq \theta(\mathbf{w}) - \theta_o \approx \mathrm{H}^{-1}\nabla_\theta L(1/N - \mathbf{w}, \theta_o)$ — *update of θ_o to $\theta(\mathbf{w})$*

where: $\theta(\mathbf{w}) = \operatorname{argmin}_\theta L(\mathbf{w}, \theta)$ — *weighted ERM training*

$L(\mathbf{w}, \theta) = \sum_{i=1}^{N}[w_i \ell_i(\theta, z_i)]$

$w_i \in [0,1], \quad \mathbf{1}^T\mathbf{w} = 1$ — *influence of z_i; normal*

$\mathrm{H}^{-1} = \left(\nabla^2_{\theta,\theta} L(1/N, \theta_o)\right)^{-1}$ — *inv-Hessian; expensive*

$$\xRightarrow{scrub\ \mathcal{D}_f} \theta_u = \theta_o + \Delta(\mathbf{w}_{MU}), \qquad \mathbf{w}_{MU} \in [0,1]^N, \qquad w_{MU,i} = \mathbb{I}_{D_r}(i)/|D_r|$$

current authors use WoodFisher approx. for implementation

2.1 UNLEARNING

Full-Stack MU Evaluation

Evaluated compared to Retrain gold-standard metrics

- **Unlearning accuracy (UA):** defined as $\mathrm{UA}(\theta_u) = 1 - \mathrm{Acc}_{\mathrm{D_f}}(\theta_u)$
- **Membership inference attack (MIA-Efficacy):** confidence-based MIA predictor against θ_u on $\mathcal{D}_f$; defined as $\mathrm{TN}/|D_f|$
- **Remaining accuracy (RA):** accuracy of θ_u on $\mathcal{D}_r$; fidelity of MU
- **Testing accuracy (TA):** generalization of θ_u on dataset outside of $\mathcal{D}$; tested on all test data except in class-wise forgetting
- **Run-time efficiency (RTE):** computation efficacy of MU

2.1 UNLEARNING

Other Paradigms

- **Differential privacy:** protecting individuals in dataset; probabilistic
- **Federated learning:** training with decentralized edge devices
- **Graph neural networks:** processing data represented by graphs
- **Adversarial ML:** attacking ML algorithms for info or manipulation
- **Conditional generative models:** generating concepts to image
- **Understanding data influence:** influence function approach, defense against data poisoning, fair learning, transfer learning

2.2 PRUNING

- Necessary due to constraints on computation, memory, etc.
- Equates to sparsification A.K.A. weight sparsity/pruning

Performance Impact

- Lottery ticket hypothesis (LTH) demonstrated the feasibility of co-improving test accuracy and efficiency (sparsity) of model
- Impact of pruning has been investigated in improving:

    generalization, robustness, fairness, interpretability, model explanation, privacy, loss landscape

- Privacy gains imply data influence connected to sparsification

winning ticket is a sparse subnetwork with equal or better accuracy

2.2 PRUNING

Pruning in MU

- A search for insights from pruning for unlearning
- **Wang et al. 2022:** removing channels of a DNN showed an unlearning benefit in federated learning.
- **Ye et al. 2022:** filter pruning was introduced in lifelong learning to detect "pruning identified exemplars" that are easy to forget

03

Proposal

3.1 Challenges

- Works' Limitations
- General Challenges

3.2 Solutions

- Schematic Overview
- Summary of Work

3.1 CHALLENGES

Limitations of Related Work

- **Exact unlearning (Retrain):** large computational overhead
- **Approximate unlearning:** speed and ease at the cost of efficacy

FT and DP impractical against attacks, GA's effectiveness of unlearning can be improved, FF low parallel efficiency and dependent on parameters, IU requires model and training assumptions

Unlearning Methods	Evaluation metrics UA	MIA-Efficacy	RA	TA	RTE
FT	✓		✓	✓	0.06×
GA	✓	✓	✓	✓	0.02×
FF	✓		✓	✓	0.9 ×
IU	✓			✓	0.08×
Ours	✓	✓	✓	✓	0.07×

Table 1

3.1 CHALLENGES

General Challenges Categorized

1. Performance of approximate unlearning can heavily rely on the configuration of algorithmic parameters.

   e.g. FF regularization parameter for each data-model setup

2. Effectiveness of scheme can vary significantly across different unlearning evaluation criteria, and their trade-offs are not well understood.

   e.g. high efficacy neither implies nor precludes high fidelity

3.2 SOLUTIONS

Schematic Overview of Proposal on Model-Sparsity Driven MU

- Going beyond data-centric
- Use of model sparsity in MU to incorporate its impact
- Use of full stack evaluation metrics for a multi-faceted analysis of models
- Comparison of various MU approaches to sparsity

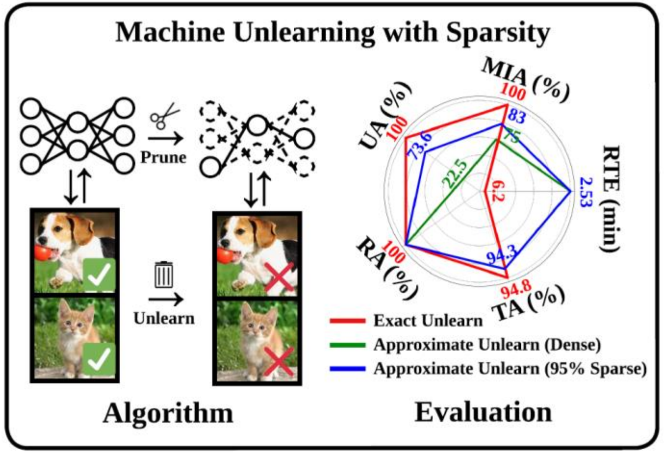

Figure 1

3.2 SOLUTIONS

Summary of Work

1. Systematically deriving theoretical connections of unlearning and pruning (rather than focusing on a specific application).
2. Practically demonstrating the effects of the theory in closing the gap between approximate unlearning and exact unlearning.
3. Developing a new paradigm termed "prune first, then unlearn" and a novel "sparsity-aware unlearning" framework to explore different methods of employing sparsification.
4. Performing extensive experiments across diverse datasets, models, and unlearning scenarios.

04

Method

4.1 Theory

- Fundamental Idea
- Error Analysis
- Expansion

4.2 Paradigms

- Prune First, then MU
- Sparsity-Aware MU

4.1 UNLEARNING

Fundamental Idea

- Based on related work, we theorize sparsity boosts multi-criteria unlearning and closes approximation gap while being efficient
- First, we prove this is the case both theoretically and practically
- Then, we develop paradigms to employ this finding

But how do we prove this?

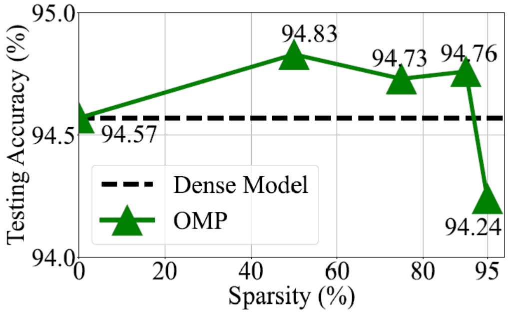

Figure 2

4.1 UNLEARNING

Error Analysis to Prove Application of Sparsity

- Let us use:
  - *Unrolling stochastic gradient descent to derive unlearning error given by weight difference in scrubbing a single data point*
  - *One-shot magnitude pruning to infuse model sparsity to SGD*
- Let us assume:
  - *Sparse pattern from OMP as binary mask :* $\mathbf{m},\ m_i \in [0,1]$
  - *Model parameters for every m_i:* $\boldsymbol{\theta},\ \theta_i$
  - *Sparse model with zeroed weights:* $\mathbf{m} \odot \boldsymbol{\theta}$

4.1 UNLEARNING

Error Analysis to Prove Application of Sparsity (cont.)

- Using SGD, error between GA-unlearned and Retrain standard:

$e(\mathbf{m}) = O\left(\eta^2 t \parallel \mathbf{m} \odot (\boldsymbol{\theta}_t - \boldsymbol{\theta}_0) \parallel_2 \sigma(\mathbf{m})\right)$ *unlearning error*

where: $\sigma(\mathbf{m}) \coloneqq \max_j \left\{\sigma_j\left(\nabla^2_{\theta,\theta} \ell\right), if\ m_j \neq 0\right\}$ *largest singular value*

- Clearly, unlearning error decreases as sparsity increases, unlike previously being proportional to model distance
- But, number of active singular values decreases as sparsity increases, causing possible generalization decrease (TA can tell)

4.1 UNLEARNING

Error Analysis to Prove Application of Sparsity (proof overview)

$$\boldsymbol{\theta}_t' \approx \boldsymbol{\theta}_0' - \eta \mathbf{m} \odot \sum_{i=1}^{t-1} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}_0', \hat{\mathbf{z}}_i) + \mathbf{m} \odot \left(\sum_{i=1}^{t-1} f(i)\right),$$

$$f(i) = -\eta \nabla^2_{\boldsymbol{\theta},\boldsymbol{\theta}} \ell(\boldsymbol{\theta}_0', \hat{\mathbf{z}}_i) \left(-\eta \sum_{j=0}^{i-1} \mathbf{m} \odot \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}_0', \hat{\mathbf{z}}_j) + \sum_{j=0}^{i-1} (\mathbf{m} \odot f(j)) \right),$$

$$e(\mathbf{m}) = \|\mathbf{e}_{\mathbf{m}}(\boldsymbol{\theta}_0, \{\hat{\mathbf{z}}_i\}, t, \eta)\|_2 = \left\| \mathbf{m} \odot \left(\sum_{i=1}^{t-1} f(i)\right) \right\|_2$$

$$\approx \eta^2 \left\| \operatorname{diag}(\mathbf{m}) \sum_{i=1}^{t-1} \nabla^2_{\boldsymbol{\theta},\boldsymbol{\theta}} \ell(\boldsymbol{\theta}_0', \hat{\mathbf{z}}_i) \sum_{j=0}^{i-1} \mathbf{m} \odot \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}_0', \hat{\mathbf{z}}_j) \right\|_2$$

through triangle inequality...

$$\leq \eta^2 \sigma(\mathbf{m}) \|\mathbf{m} \odot (\boldsymbol{\theta}_t - \boldsymbol{\theta}_0)\|_2 \frac{1}{t} \frac{t-1}{2} t = \frac{\eta^2}{2} (t-1) \|\mathbf{m} \odot (\boldsymbol{\theta}_t - \boldsymbol{\theta}_0)\|_2 \sigma(\mathbf{m}),$$

4.1 UNLEARNING

Expansion of Theory to Practice

- Does the above benefit of model sparsification in MU apply to other approximate unlearning methods besides GA?
- Let us check the common metrics i.e. unlearning efficacy (UA and MIA-Efficacy), fidelity (RA), and generalization (TA):

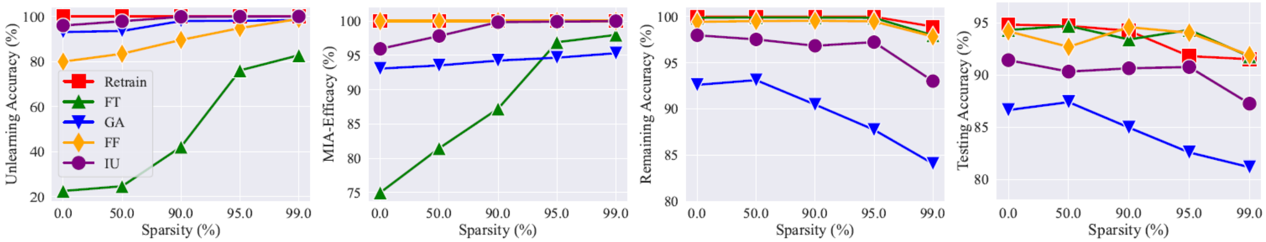

Figure 3

4.2 PARADIGMS

- Theoretically and with limited methods, sparsity worked
- But:
  - **How does the choice of weight-pruning method impact the unlearning performance?**
  - **Can sparsity-aware MU methods that directly scrub data influence from a dense model be developed?**

4.2 PARADIGMS

Prune First, then Unlearn (as we saw before)

- **What pruning to choose?**
  - Random initialization pruning before training?
  - Simultaneous pruning-training iterative magnitude pruning?
- **What matters?**
  1. Least dependence on $\mathcal{D}_f$
  2. Lossless generalization when pruning
  3. Pruning efficiency

4.2 PARADIGMS

Prune First, then Unlearn (cont.)

Choice of pruning methods based on criteria

- **SynFlow (synaptic flow pruning):** training-free pruning method at initialization, even without accessing the dataset (for ❶)
- **OMP (one-shot magnitude pruning):** performed over θ_o and depends on $\mathcal{D}_f$ but computationally light (for ❸) and has better generalization (for ❷)
- **IMP (iterative magnitude pruning):** not suitable for MU despite accuracy due to computation overhead and dependence on $\mathcal{D}_f$

4.2 PARADIGMS

Prune First, then Unlearn (cont.)

- We compare the efficacy of FT-based MU on sparse models generated using different pruning methods (SynFlow, OMP, and IMP)
- We see:
  - IMP UA decrease due to reliance
  - OMP closer to Retrain
- OMP will be our choice due to balance

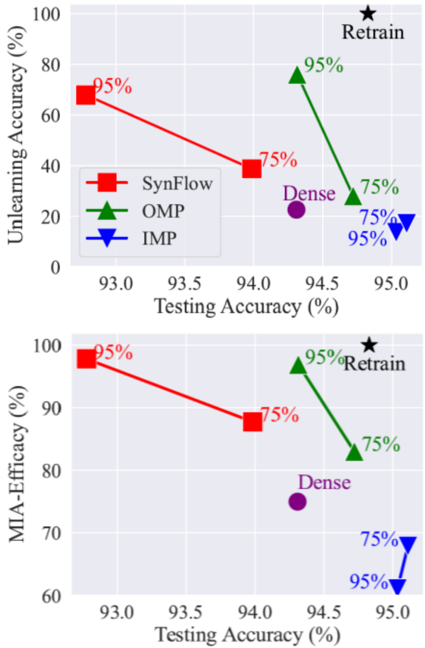

Figure 4

4.2 PARADIGMS

Sparsity-Aware Unlearning

- How about a method for simultaneous pruning and unlearning?
- Inspired by sparsity-inducing optimization, we integrate sparse penalty (ℓ_1 norm-based) into unlearning objective function
- Therefore ℓ_1**-sparse MU**:

$$\theta_u = \text{argmin}_\theta L_u(\theta; \theta, D_r) + \gamma \parallel \theta \parallel_1$$

where:	$L_u(\theta; \theta, D_r)$	*FT objective function*
	$\gamma > 0$	*regularization*

4.2 PARADIGMS

Sparsity-Aware Unlearning (cont.)

- Choice of γ matters and is a limitation; a spare-regularization scheduler can mitigate this issue with three schemes:
  1. Constant γ
  2. Linearly growing γ
  3. Linearly decaying γ *outperforms others!*

Table 2

MU	UA	MIA-Efficacy	RA	TA	RTE (min)
Retrain	5.41	13.12	100.00	94.42	42.15
ℓ_1-sparse MU + constant γ	6.60 (1.19)	14.64 (1.52)	96.51 (3.49)	87.30 (7.12)	2.53
ℓ_1-sparse MU + linear growing γ	3.80 (1.61)	8.75 (4.37)	97.13 (2.87)	90.63 (3.79)	2.53
ℓ_1-sparse MU + linear decaying γ	**5.35** (**0.06**)	**12.71** (**0.41**)	**97.39** (**2.61**)	**91.26** (**3.16**)	2.53

05

Experiments

5.1 Setup

- Datasets & Models
- MU & Pruning
- Evaluation

5.2 Results

- Method Tests
- Applications
- Other Results

5.1 SETUP

Datasets and Models

- Mainly image classification under CIFAR-10 using ResNet-18
- Other datasets and an alternate architecture also explored

Unlearning and Pruning

- Class-wise and random data forgetting for FT, GA, FF, and IU
- Both paradigms “prune-first” and “sparse-aware” with OMP

Evaluation

- UA, MIA-Efficacy, RA, TA, and RTE
- Whole proximity to Retrain gauged using Disparity Average

5.2 RESULTS

Experiments on Proposed Methods

- **Prune first, then unlearn:**

  Performance gap reduces with sparsity but Retrain has 3% TA drop; FT & UI preserve TA with tradeoff; FF loss in random-data forgetting

MU	UA DENSE	UA 95% Sparsity	MIA-Efficacy DENSE	MIA-Efficacy 95% Sparsity	RA DENSE	RA 95% Sparsity	TA DENSE	TA 95% Sparsity	Disparity Ave. ↓ DENSE	Disparity Ave. ↓ 95% Sparsity	RTE (min)
Class-wise forgetting											
Retrain	100.00$_{\pm0.00}$	100.00$_{\pm0.00}$	100.00$_{\pm0.00}$	100.00$_{\pm0.00}$	100.00$_{\pm0.00}$	99.99$_{\pm0.01}$	94.83$_{\pm0.11}$	91.80$_{\pm0.89}$	0.00	0.00	43.23
FT	22.53$_{\pm8.16}$ (77.47)	73.64$_{\pm9.46}$ (26.36)	75.00$_{\pm14.68}$ (25.00)	83.02$_{\pm16.33}$ (16.98)	99.87$_{\pm0.04}$ (0.13)	99.87$_{\pm0.05}$ (0.12)	94.31$_{\pm0.19}$ (0.52)	94.32$_{\pm0.12}$ (2.52)	25.78	11.50	2.52
GA	93.08$_{\pm2.29}$ (6.92)	98.09$_{\pm1.11}$ (1.91)	94.03$_{\pm3.27}$ (5.97)	97.74$_{\pm2.24}$ (2.26)	92.60$_{\pm0.25}$ (7.40)	87.74$_{\pm0.27}$ (12.25)	86.64$_{\pm0.28}$ (8.19)	82.58$_{\pm0.27}$ (9.22)	7.12	6.41	0.33
FF	79.93$_{\pm8.92}$ (20.07)	94.83$_{\pm4.29}$ (5.17)	100.00$_{\pm0.00}$ (0.00)	100.00$_{\pm0.00}$ (0.00)	99.45$_{\pm0.24}$ (0.55)	99.48$_{\pm0.33}$ (0.51)	94.18$_{\pm0.08}$ (0.65)	94.04$_{\pm0.10}$ (2.24)	5.32	1.98	38.91
IU	87.82$_{\pm2.15}$ (12.18)	99.47$_{\pm0.15}$ (0.53)	95.96$_{\pm0.21}$ (4.04)	99.93$_{\pm0.04}$ (0.07)	97.98$_{\pm0.21}$ (2.02)	97.24$_{\pm0.13}$ (2.75)	91.42$_{\pm0.21}$ (3.41)	90.76$_{\pm0.18}$ (1.04)	5.41	1.10	3.25
Random data forgetting											
Retrain	5.41$_{\pm0.11}$	6.77$_{\pm0.23}$	13.12$_{\pm0.14}$	14.17$_{\pm0.18}$	100.00$_{\pm0.00}$	100.00$_{\pm0.00}$	94.42$_{\pm0.09}$	93.33$_{\pm0.12}$	0.00	0.00	42.15
FT	6.83$_{\pm0.51}$ (1.42)	5.97$_{\pm0.57}$ (0.80)	14.97$_{\pm0.62}$ (1.85)	13.36$_{\pm0.59}$ (0.81)	96.61$_{\pm0.25}$ (3.39)	96.99$_{\pm0.31}$ (3.01)	90.13$_{\pm0.26}$ (4.29)	90.29$_{\pm0.31}$ (3.04)	2.74	1.92	2.33
GA	7.54$_{\pm0.29}$ (2.13)	5.62$_{\pm0.46}$ (1.15)	10.04$_{\pm0.31}$ (3.08)	11.76$_{\pm0.52}$ (2.41)	93.31$_{\pm0.04}$ (6.69)	95.44$_{\pm0.11}$ (4.56)	89.28$_{\pm0.07}$ (5.14)	89.26$_{\pm0.15}$ (4.07)	4.26	3.05	0.31
FF	7.84$_{\pm0.71}$ (2.43)	8.16$_{\pm0.67}$ (1.39)	9.52$_{\pm0.43}$ (3.60)	10.80$_{\pm0.37}$ (3.37)	92.05$_{\pm0.16}$ (7.95)	92.29$_{\pm0.24}$ (7.71)	88.10$_{\pm0.19}$ (6.32)	87.79$_{\pm0.23}$ (5.54)	5.08	4.50	38.24
IU	2.03$_{\pm0.43}$ (3.38)	6.51$_{\pm0.52}$ (0.26)	5.07$_{\pm0.74}$ (8.05)	11.93$_{\pm0.68}$ (2.24)	98.26$_{\pm0.29}$ (1.74)	94.94$_{\pm0.31}$ (5.06)	91.33$_{\pm0.22}$ (3.09)	88.74$_{\pm0.42}$ (4.59)	4.07	3.08	3.22

Table 3

5.2 RESULTS

Experiments on Proposed Methods (cont.)

- **Sparsity-aware unlearning:**

  Only comparing to FT due to previous results

It outperforms FT in efficacy

Closes gap with Retrain

Faces no computation loss

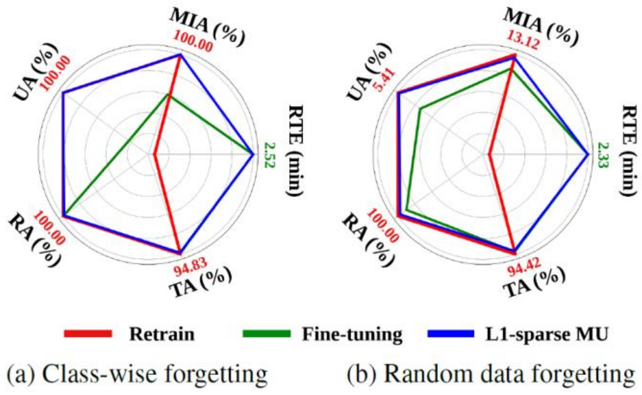

Figure 5

5.2 RESULTS

Uses In Different Applications

- **MU for Trojan model cleanse:**

  Removing influence of poisoned backdoor data, manipulated by injecting trigger; can cause incorrect prediction with trigger

FT decreases ASR

FT has little SA loss

ℓ_1-sparse performs similarly

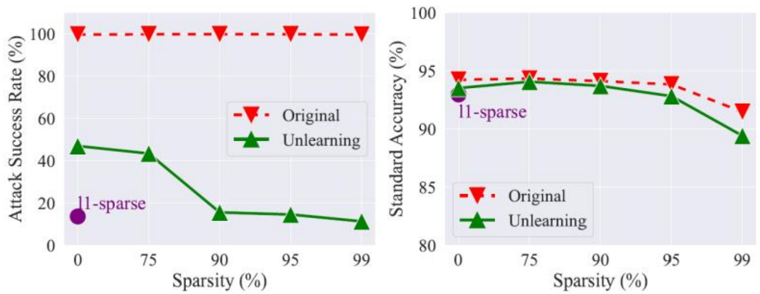

Figure 6

5.2 RESULTS

Uses In Different Applications (cont.)

- **MU for transfer learning:**

  Mitigating impact of harmful data classes to enhance model's accuracy on other datasets after finetuning; we use ℓ_1*-sparse*

Keeps accuracy of Retrain

Has 2× speed up

Suitable for large-scale

Forgetting class #	0	100		200		300	
	Acc	Acc	Time	Acc	Time	Acc	Time
OxfordPets							
Method [51]	85.70	85.79	71.84	86.10	61.53	86.32	54.53
ℓ_1-sparse MU		85.83	35.47	86.12	30.19	86.26	26.49
SUN397							
Method [51]	46.55	46.97	73.26	47.14	61.43	47.31	55.24
ℓ_1-sparse MU		47.20	36.69	47.25	30.96	47.37	27.12

Table 4

5.2 RESULTS

Additional Results

- **Model sparsity for data privacy:**

  Assessing MIA-Privacy to check how much gets leaked in MIA about $\mathcal{D}_r$; lower is better

  Sparsity increase causes MIA-P decrease

  Approx. bests Retrain ($\mathcal{D}_r$ dependent)

  IU and GA are more private

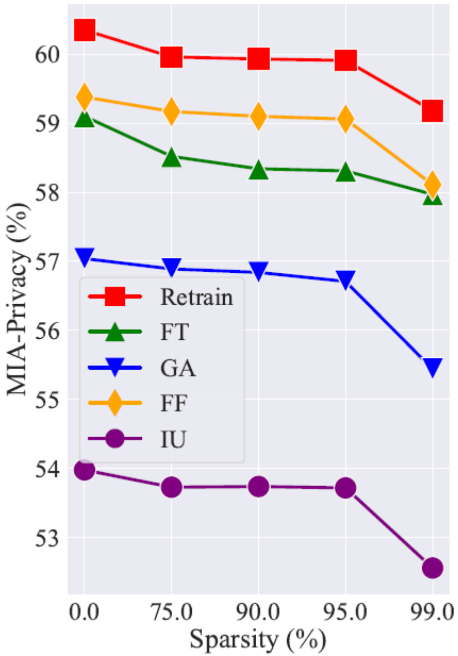

Figure A1

06 Conclusion

6.1 Limitations

6.2 Future Work

6.1 LIMITATIONS

- In the theoretical analysis of sparsity, it is unclear if it is better than other methods that improve performance, and if yes, why.
- $\varepsilon - \delta$ forgetting is briefly addressed but despite relevance, is not explored and its contribution is unevaluated.
- Current simulations are all on computer vision tasks which limits the application domain (though architectures are varied).
- Theoretical analysis is not performed on ℓ_1-*sparse* update rule and might be difficult (similar to influence function methods relying on strongly convex loss).

6.2 FUTURE WORK

- The study indicates model modularity traits such as weight sparsity that could amplify MU and should be investigated.
- The application of this approach in other types of datasets and model architectures (e.g. language models) might face new challenges and reveal interesting results.

THANKS!

Do you have any questions?

Presentation by Maryam Rezaee

Machine Learning Seminar | *Fall 1403*

Dr. Fatemeh SeyyedSalehi

Sharif University of Technology