CY 7790, Lecture 18 on Nov 22

Gokberk Yar and Michael Davinroy

September 27, 2021

1 Machine Unlearning

Problem Statement. Since the introduction of GDPR and other legislation, users now have a right to be forgotten. Allowing for this right is a notoriously difficult problem in general, but even more so in the context of machine learning, as what models learn is currently not fully understood and/or the model may have memorized a user's data-point. To tackle this problem, this paper introduces an efficient method for frameworks to unlearn data-points.

Threat Model. There is no exact threat model addressed in this paper, but the solution given is trying to prevent some adversary from gaining information about a user who wants to have their data forgotten in accordance with appropriate law.

Methodology. This paper introduces SISA: Sharded, Isolated, Sliced, and Aggregated Training. Figure 1 shows a visualized overview of the proposed system. The data is *sharded*, in that it is divided up into S shards, denoted by D_i, which train S *independent* models, M_i. These shards are additionally *sliced*, so that the model can incrementally learn on additional slices. The insight here is that if a user's data is removed, the model can start again training again at weights of the slice directly before the the once containing the data to remove. Finally, when evaluating, each model makes a prediction, and the the results are *aggregated* into a final output, similar to an ensemble.

Each of the sharding, slicing, and aggregation techniques can be adapted to and tuned to suit the user's needs. The benefits of this approach are that it is generalizable, intuitive, provable, and audit-able. The downsides of the approach are that each isolated model is trained on separate, smaller datasets, so they might disagree and/or be weaker models than if there was a single model trained on all the available data.

For evaluation, the paper presents two baselines: 1. "batch K unlearning requests and retrain the entire model after every K unlearning requests. This is the same to the naive baseline of retraining the entire dataset (without the points to be unlearned) from scratch, in a batch setting", and 2. "train on a 1 S fraction of the data and only retrain when the point to be unlearned falls into this set."

Figure 2 shows the paper's results and their conclusions for testing the accuracy and retraining time of the SISA approach against their two baselines. Figure 3 shows the accuracy of the models for different levels of sharding over the number of epochs. The paper claims, and the graphs show, that when the number of epochs is low, more sharding leads to less accuracy, but as the number of epochs go up, the accuracies of the different levels of sharding tend to converge to similar levels. Figure 4 shows that on more complex datasets, SISA performs worse than the K baseline, but better than the $1/S$ baseline in terms of top-1 accuracy. Finally, they examine the effects of SISA on transfer learning accuracy (Figure 5) and propose optimizations on the approach for when the model maker knows some distribution of the users who might wish to be forgotten (Figure 6).

Discussion. In class, we first discussed how GDPR (and likely other new legislation) is actually really well enforced and taken seriously, so the motivations for this paper are strong. Second, someone asked how long model deployers have to comply with the right to be forgotten, as to understand the timeline that the model deployer could batch requests. We found out it was 30 (or in some case 60) days. Next, we questioned why they didn't use a model aggregation approach like in federated learning, since SISA trades off accuracy due to it being an ensemble. We concluded that the federated learning approach of aggregating models instead of data might not work because you

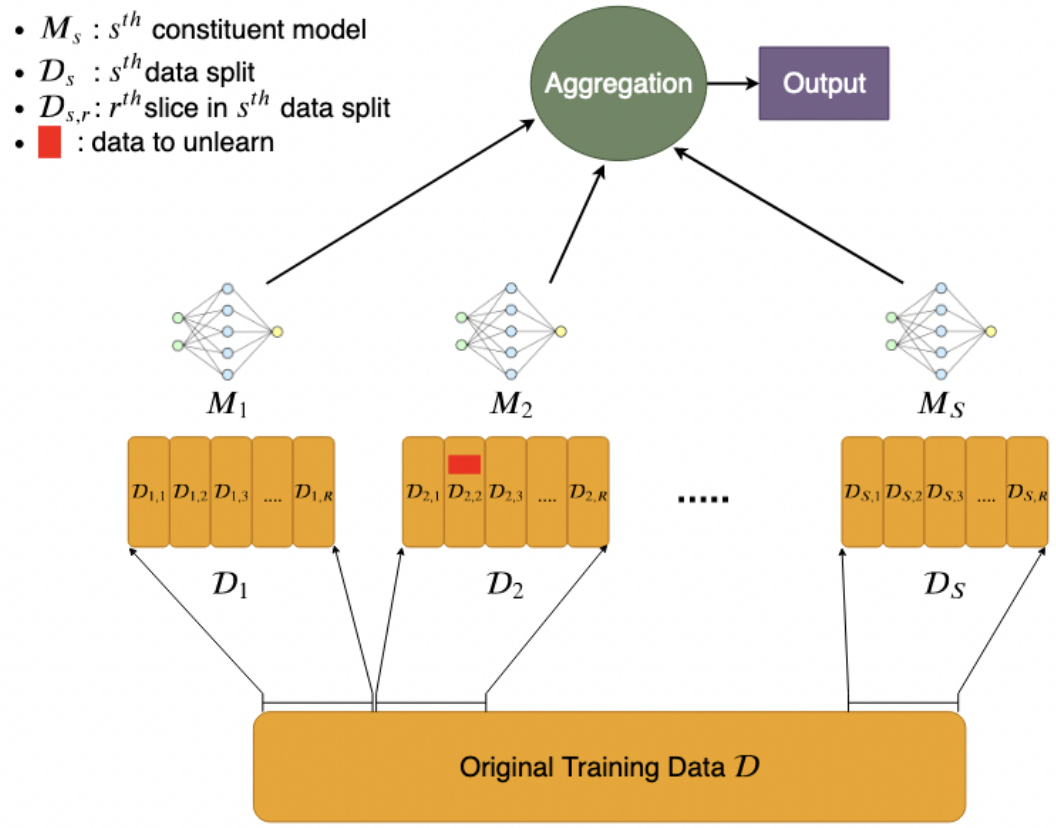


Figure 1: Graphical overview of SISA presented in the paper.

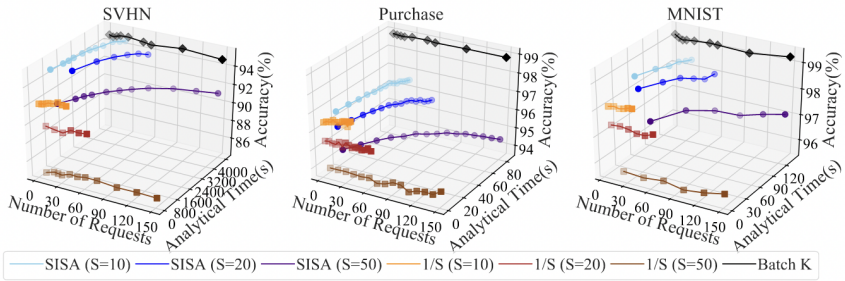
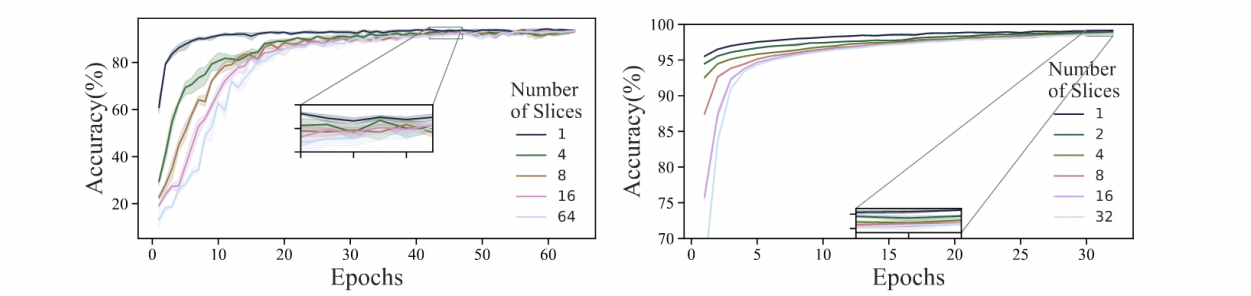


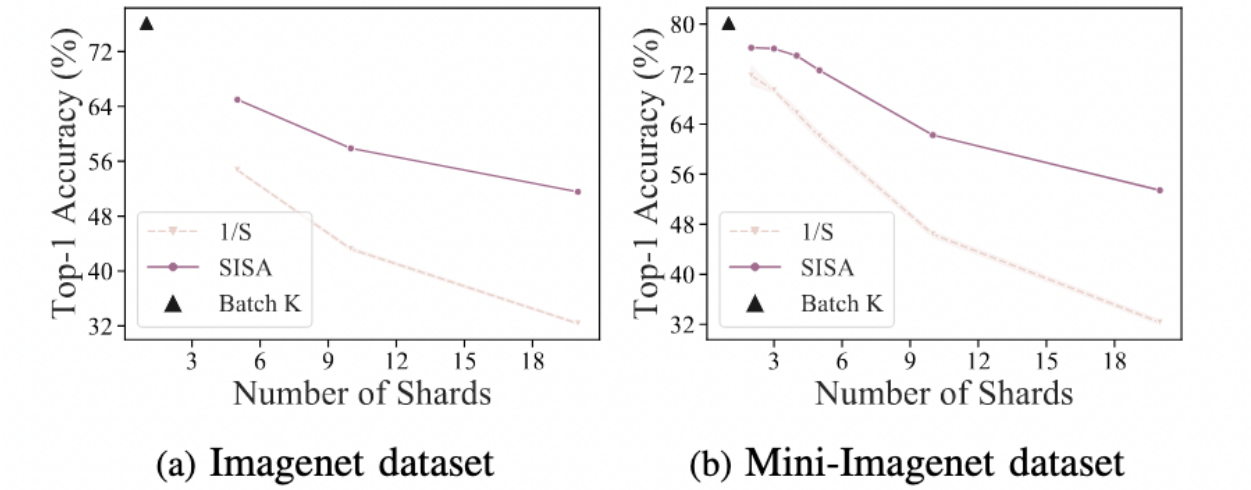
Fig. 4: We compare the experimental accuracy of **SISA** training (with different number of shards) with the two baselines on three datasets: SVHN, Purchase, and MNIST. It is clear that **SISA** training provides higher accuracy than the $\frac{1}{S}$ fraction baseline, along with less retraining time than the batch K baseline especially when the number of unlearning request is small.

Figure 2: Figure 4 from the paper showing results of accuracy and retraining time against the two baselines.



(a) Accuracy vs. Number of epochs for SVHN dataset. (b) Accuracy vs. Number of epochs for Purchase dataset.

Figure 3: Results from the paper comparing the number of shards to the accuracy over epochs.



(a) Imagenet dataset (b) Mini-Imagenet dataset

Figure 4: Comparison of top-1 accuracy on more complex datasets vs baselines.

might need many, many iterations to get similar accuracy to the ensemble model. After this, we commented on how the $1/S$ baseline is pretty unrealistic and that this observations weakens their results. However, as the first paper into this space, we noted that they didn't have a standard baseline to evaluate against, so this was a reasonable first step. We also note they only test up to 20 shards, which is not a ton, and even then their accuracy is pretty low except for when comparing to the questionable $1/S$ baseline. Finally, in regard to their presentation, we didn't like their 3D graphs, as they were much more difficult to read than a few more corresponding 2D graphs. Overall, we thought the paper presented a good idea that warrants more research into this space.

2 Brockschmidt et al. Analyzing Information Leakage of Updates to Natural Language Models.

Problem Statement. Most of the data science pipelines involves continuous re-training and update of the model that used in the production. This retraining could be a result of a new available data, removal of some data or re-finetuning. This paper focus capabilities of an attacker which performs a differential analysis in the language model context. Authors defined differential score and differential rank metrics; evaluated different datasets and models using these

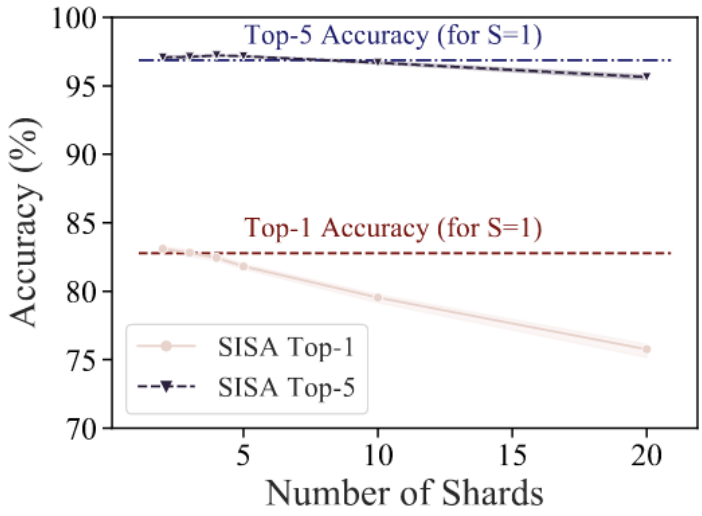


Fig. 8: In the setting of transfer learning (from ImageNet to CIFAR-100), we observe a lower accuracy degradation induced by **SISA** training (with $S > 1$).

Figure 5: Analysis of SISA's effect on transfer learning accuracy.

metrics; discussed its privacy implications and suggested mitigation strategies.

Sources of re-training **1) Data Update,** re-training when new data is available.**2) Data Deletion,** when a user requested data deletion (becomes more frequent due to new regulatory laws).**3)Data Specialization,** model finetuned on big language models like BERT. .

Adversarial Capabilities and Goals. Informally, adversary has black-box query access to two models. These two models are related, one of them is a newer version of the other due to a re-training source discussed in previous section. Adversary's goal is to infer/leak information about the training data which is only used in one of the model's training but not used in the other one by performing a differential analysis. Formally, adversary has access to languages models M_D and $M_{D'}$ trained on D and D' respectively, where D' differ from D due to source of re-training in previous section. Adversary's goal to infer/leak information on $D' \setminus D$.

Contributions First study on privacy implications of releasing snapshots of language models.

1. By comparing two snapshots of the language model, adversary can leak information about the dataset even if the change is very small.
2. Comparing snapshots leaks more information than single snapshot.
3. Adding or removing additional non-sensitive data between two releases don't work as a mitigation strategy.
4. Differential privacy works as a mitigation strategy but computation cost and accuracy degradation is high.

Algorithm 1 Distribution-Aware Sharding

Input: Dataset $\mathcal{D}$, constant C

procedure $ShardData(\mathcal{D}, C)$
  sort $\{d_u\}_{i=1}^{|\mathcal{D}|}$ by $p(u)$
  $i \leftarrow 0$
  create empty shard $\mathcal{D}_i$
  for $j \leftarrow 0$ to $|\mathcal{D}|$ **do**
    remove d_u with lowest $p(u)$ from $\mathcal{D}$
    $\mathcal{D}_i = \mathcal{D}_i \cup d_u$
    if $\mathbb{E}(\chi_i) \geq C$ **then**
      $\mathcal{D}_i = \mathcal{D}_i \setminus d_u$
      $i \leftarrow i + 1$
      create empty shard $\mathcal{D}_i$
      $\mathcal{D}_i = \mathcal{D}_i \cup d_u$
    end if
  end for
end procedure

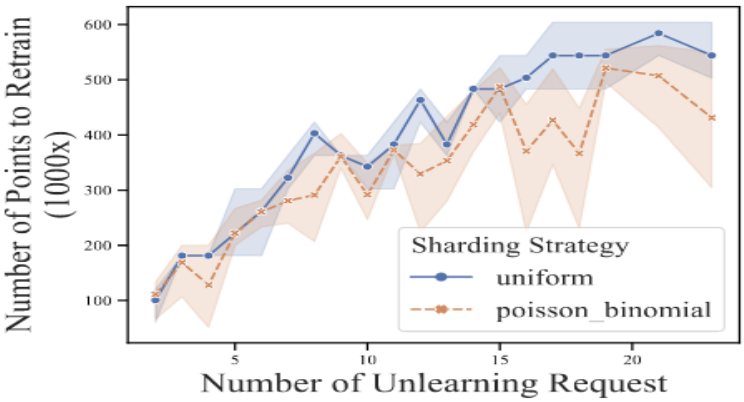


Fig. 10: # points (variance shaded) of the SVHN dataset that need to be retrained for uniform and distribution-aware sharding where users have varying probability of revoking access to their data.

result means that distribution-aware sharding incurs a trade-off of accuracy for decreased unlearning overhead. We leave to future work the exploration of alternatives to majority voting aggregation that would cope with such imbalanced shard sizes.

Figure 6: Proposed algorithm and evaluation of optimization when the model-maker knows some distribution of users more likely to ask for their data to be unlearned.

5. Restriction on model access and limited prediction results are other promising mitigation strategies with lower cost.

Background: Generative Language Models Generative Language Models estimate probability of a token sequence by calculating product of the per token conditional probabilities given the prefix before the token, Figure 7 defines it formally.

$$p(t_1 \dots t_n) = \prod_{1 \le i \le n} p(t_i \mid t_1 \dots t_{i-1}) .$$

Figure 7: Language Model Likelihood Estimation for a token sequence

Datasets & Models Used

1. Penn Treebank, 900k tokens, 10k vocabulary, two layer LSTM.
2. Reddit comments, 20m tokens,10k vocabulary, a RNN model and BERT.
3. Wikitext-103, 103m tokens, 20k vocabulary, two layer LSTM.

Differential Score and Rank Metrics defined in the Figure 8 and approximated using the algorithm at Figure 9.

Results on Canaries In Figure 10 Table 1, differential score in the white cells are almost always near to 5 and large values and values in the gray are close to 0 ans smaller values compared to DR=0, which is expected from the Bean Search Algorithm. In short, Figure 10 shows that beam search estimates as expected. Similarly, in Figure 10, Table 2, white cells are high and close to 5 as expected and gray cells are close to 0 so beam search algorithm estimates well.

Results on Real Word Data In Figure 11, Table 3, They can extract newly added data. Since top results are all related to hockey original training was on the politics data.

Mitigation

1. Differential Privacy, model accuracy significantly drops and computational overhead is high.
2. Two-Stage Continued Training
3. Truncating Outputs, only return top k tokens from the updated model, reduces leakage.

Class Discussion.

1. First study on releasing snapshots of language models.
2. Defines two metrics for measuring information leakage.
3. Analysis show that there is a substantial risk.
4. Adversary does not need have an auxilary datasets to perform the attack.
5. Metrics are model agnostic.
6. Definition of token is not clear, assuming word level.

Definition 3.1. Given two language models M, M' and a token sequence $t_1 \dots t_n \in T^*$, we define the *differential score* of a token as the increase in its probability and the *relative differential score* as the relative increase in its probability. We lift these concepts to token sequences by defining

$$DS_M^{M'}(t_1 \dots t_n) = \sum_{i=1}^{n} M'(t_{<i})(t_i) - M(t_{<i})(t_i),$$

$$\widetilde{DS}_M^{M'}(t_1 \dots t_n) = \sum_{i=1}^{n} \frac{M'(t_{<i})(t_i) - M(t_{<i})(t_i)}{M(t_{<i})(t_i)}.$$

The differential score of a token sequence is best interpreted relative to that of other token sequences. This motivates ranking sequences according to their differential score.

Definition 3.2. We define the *differential rank* $DR(s)$ of $s \in T^*$ as the number of token sequences of length $|s|$ with differential score higher than s.

$$DR(s) = \left| \left\{ s' \in T^{|s|} \,\middle|\, DS_M^{M'}(s') > DS_M^{M'}(s) \right\} \right|.$$

The lower the differential rank of a sequence, the more the sequence is exposed by a model update, with the most exposed sequence having rank 0.

Figure 8: Differential Score and Rank Definitions

Algorithm 1 Beam search for Differential Rank

In: M, M'=models, T=tokens, k=beam width, n=length
Out: S=set of (n-gram, DS) pairs

$S \leftarrow \{(\epsilon, 0)\}$ ▷ Initialize with empty sequence ϵ
for $i = 1 \dots n$ **do**
  $S' \leftarrow \{(s \circ t, r + DS_M^{M'}(s)(t)) \mid (s, r) \in S, t \in T\}$
  $S \leftarrow take(k, S')$ ▷ Take top k items from S'
return $S = \{(s_1, r_1), \dots, (s_k, r_k)\}$ such that $r_1 \geq \dots \geq r_k$

Figure 9: Beam Search Algorithm Used to Estimate Differential Rank

7. Hyperparameter tuning for the beam width is hard.

8. They used conversations on two different topics, this could unintentionally make models to memorize more than usual.

Table 1: Differential score (DS) for different datasets, model architectures, canaries, and insertion frequencies. White cells represent a differential rank (DR) of 0 (as approximated by beam search), and gray cells represent $DR > 1000$.

Dataset	Penn Treebank			Reddit						Wikitext-103	
Model Type (Perplexity)	RNN (120.90)			RNN (79.63)			Transformer (69.29)			RNN (48.59)	
Canary Token Freq.	1:18K	1:3.6K	1:1.8K	1:1M	1:100K	1:10K	1:1M	1:100K	1:10K	1:1M	1:200K
All Low	3.40	3.94	3.97	2.83	3.91	3.96	3.22	3.97	3.99	1.39	3.81
Low to High	3.52	3.85	3.97	0.42	3.66	3.98	0.25	3.66	3.97	0.07	3.21
Mixed	3.02	3.61	3.90	0.23	3.04	3.92	0.39	3.25	3.96	0.25	3.02
High to Low	1.96	2.83	3.46	0.74	1.59	2.89	0.18	1.87	3.10	0.08	1.22

Table 2: Differential Score ($DS_{\tilde{M}}^{M'}$) of the mixed frequency canary phrase for the Reddit (RNN) model using different update techniques. Model M is trained on D_{orig}. For the *Retraining* column, M' is trained on $D_{orig} \cup D_{extra} \cup C$ starting from random initial parameters. For the *Cont'd Training 1* column, M' is trained on $D_{extra} \cup C$ starting from M. For the *Cont'd Training 2* column, we first train a model $\tilde{M}$ on $D_{extra} \cup C$ starting from M, and then train model M' from $\tilde{M}$ using additional public data D'_{extra}. A white cell background means that the differential rank DR (as approximated by our beam search) of the phrase is 0, gray cell background means that DR is >1000.

	Retraining				Continued Training 1			Continued Training 2
$\lvert D_{extra}\rvert / \lvert D_{orig}\rvert$	0%	20%	50%	100%	20%	50%	100%	100%
1:1M	0.23	0.224	0.223	0.229	0.52	0.34	0.46	0.01
1:100K	3.04	3.032	3.031	3.038	3.56	3.25	3.27	0.26

Figure 10: Differential Score Canary

Table 3: Top ranked phrases in group beam search for a model updated with `rec.sport.hockey`. For the layperson: Los Angeles Kings, Minnesota North Stars, and Toronto Maple Leaf are National Hockey League teams; Norm Green was the owner of the North Stars; an ice hockey game consists of three periods with overtime to break ties. Capitalization added for emphasis.

RNN		Transformer	
Phrase	$\widehat{DS}$	Phrase	$\widehat{DS}$
`Angeles Kings prize pools`	56.42	`Minnesota North Stars playoff`	96.81
`National Hockey League champions`	53.68	`Arsenal Maple Leaf fans`	71.88
`Norm 's advocate is`	39.66	`Overtime no scoring chance`	54.77
`Intention you lecture me`	21.59	`Period 2 power play`	47,85
`Covering yourself basically means`	21.41	`Penalty shot playoff results`	42.63

Figure 11: Real Dataset