DeepClean: Machine Unlearning on the Cheap by Resetting Privacy Sensitive Weights using the Fisher Diagonal

Kosti Gourgoulias, JPMorganChase

U&Me Workshop 2024, ECCV

Machine Unlearning

Given a trained model F_w on a dataset $D = D_r \cup D_f$, $D_r \cap D_f = \emptyset$, can we adjust the model to remove the influence of D_f?

D_f could be: private photos, text under copyright, data points with noisy labels, etc.

D_f could consist of random examples, all examples of a single class, etc.

How can we intervene on models that have complex representations?

Our first motivation was the Machine Unlearning challenge at NeurIPS 2023.

The task was to “forget” a dataset of images of people’s faces.

GOOGLE · RESEARCH CODE COMPETITION · 10 MONTHS AGO

NeurIPS 2023 - Machine Unlearning

Erase the influence of requested samples without hurting accuracy

Competition organized by Triantafillou Eleni,

Fabian Pedregosa, Isabelle Guyon, et al.

Various approaches

Retrain: Retrain the model from scratch on D_r — best option, but expensive.

Fine-tune on D_r: Further training on the retain set, letting performance on the rest to grow stale.

Gradient Ascent: Fine-tune the model with gradient ascent on D_f.

Information-theory inspired methods:

Fisher Masking (Liu, et al, 2023): Identify the parameters most responsible for performance on D_f and mask them, then fine-tune remaining model on D_r to recover performance.

Fisher forgetting (Golatkar, Achille, Soatto, 2020): Same as above, but instead of masking, Gaussian noise is added to those parameters according to Fisher information.

And many others …

For references on the above, see “Machine Unlearning in Learned Databases: An Experimental Analysis”, Kurmanji, Triantafillou E., Triantafillou P.

Fisher Information

Let's assume D contains features x and classes y.

Empirical Fisher Information matrix of the parameters w:

$$I_D(w) = \frac{1}{|D|} \sum_{(x,y)\in D} \nabla_w \log p(y \mid x, w) \nabla_w \log p(y \mid x, w)^T .$$

How much information D carries about w.

Typically diagonal approximations are used due to the cost of calculating (and storing) the whole matrix [1].

1. See Kirkpatrick et al, 2017 and Golatkar, Achille, and Soatto, 2020. Kronecker-type factorizations are also possible, but we didn't explore them in this work.

For each parameter, we define the ratio

$$r(w_i) := \frac{I_{D_f}(w_i)}{I_{D_r}(w_i)},$$

where $I_D(w_i) := (I_D(w))_{i,i}$, the i-th diagonal element of the empirical Fisher.

$r(w)$ increases when D_f is more informative for a parameter compared to D_r .

Simple strategy:

1. Select parameters responsible for D_f performance based on a threshold γ, $r(w_i) > \gamma$.

2. Fine-tune them on D_r while keeping the rest frozen.

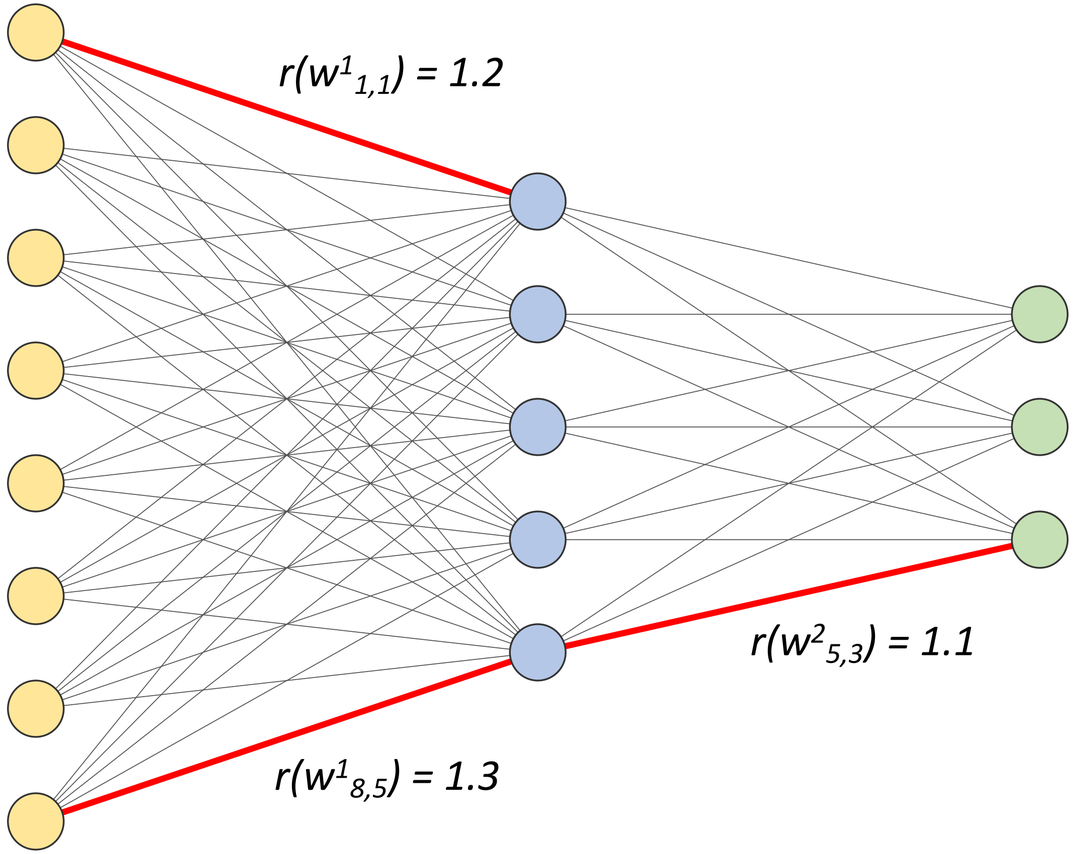
The *DeepClean* Algorithm:

1). *Compute ratio of forget and retain set fisher diagonal entries for each weight*

$$r(w_i) := \frac{I_{D_f}(w_i)}{I_{D_r}(w_i)}$$

2). *Update the weights, — indicated in red, that have* **r > threshold** *via fine-tuning on the retain set. Keep other weights frozen.*

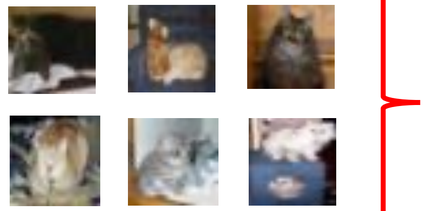
The larger the threshold, the fewer weights we need to update, but we may be leaving information about D_f in the model.



Input Layer ∈ $\mathbb{R}^8$ Hidden Layer ∈ $\mathbb{R}^5$ Output Layer ∈ $\mathbb{R}^3$

Forget set

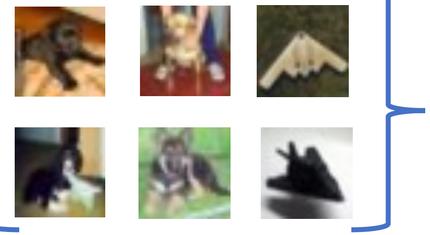
D_f



Cat

Retain set

D_r



Dog & Airplanes

Experimental Setup

Datasets:

- Cifar-10 (in this presentation)
- (but also MNIST / Cifar-100)

Models:

- ResNet18
- VGG-16
- ViT: vision transformer

Metrics used:

- Acc_{D_r}: the unlearned model's classification accuracy on the retain set.
- $\Delta Acc_{D_f} = \text{Unlearned } Acc_{D_f} - \text{Gold } Acc_{D_f}$, measures the unlearned model's classification deviation from the gold model.
- Unlearning time: time to apply each unlearning algorithm (seconds).
- $\Delta MIA = \text{Unlearned } MIA - \text{Gold } MIA$, measures the Membership Inference Attack deviation from the gold model.

MIA measures the probability of an attacker successfully determining whether a particular data record was part of the training set.

Experimental Setup

- **Gold model:** Retrained model from scratch on just the retain set D_r

- **DeepClean:** our method.

For the below, we adopted the best set up / hyperparameters from each corresponding work.

- **Model Sparsification (Sparse MU)** [1]**:** Fine-tuning a model on D_r with a sparsity-inducing regularizer.

- **L-CODEC** [2]**:** An influence-function method.

Two scenarios:

- **Random sample unlearning (RN):** We pick 10% of samples randomly and try to unlearn them.

- **Label unlearning (label):** we try to unlearn an entire class.

1. Model sparsity can simplify machine unlearning, Jia, et al., 2024

2. Deep unlearning via randomized conditionally independent hessians, Mehta, et al., 2022.

Results

Unlearning performance for both tasks on cifar-10.

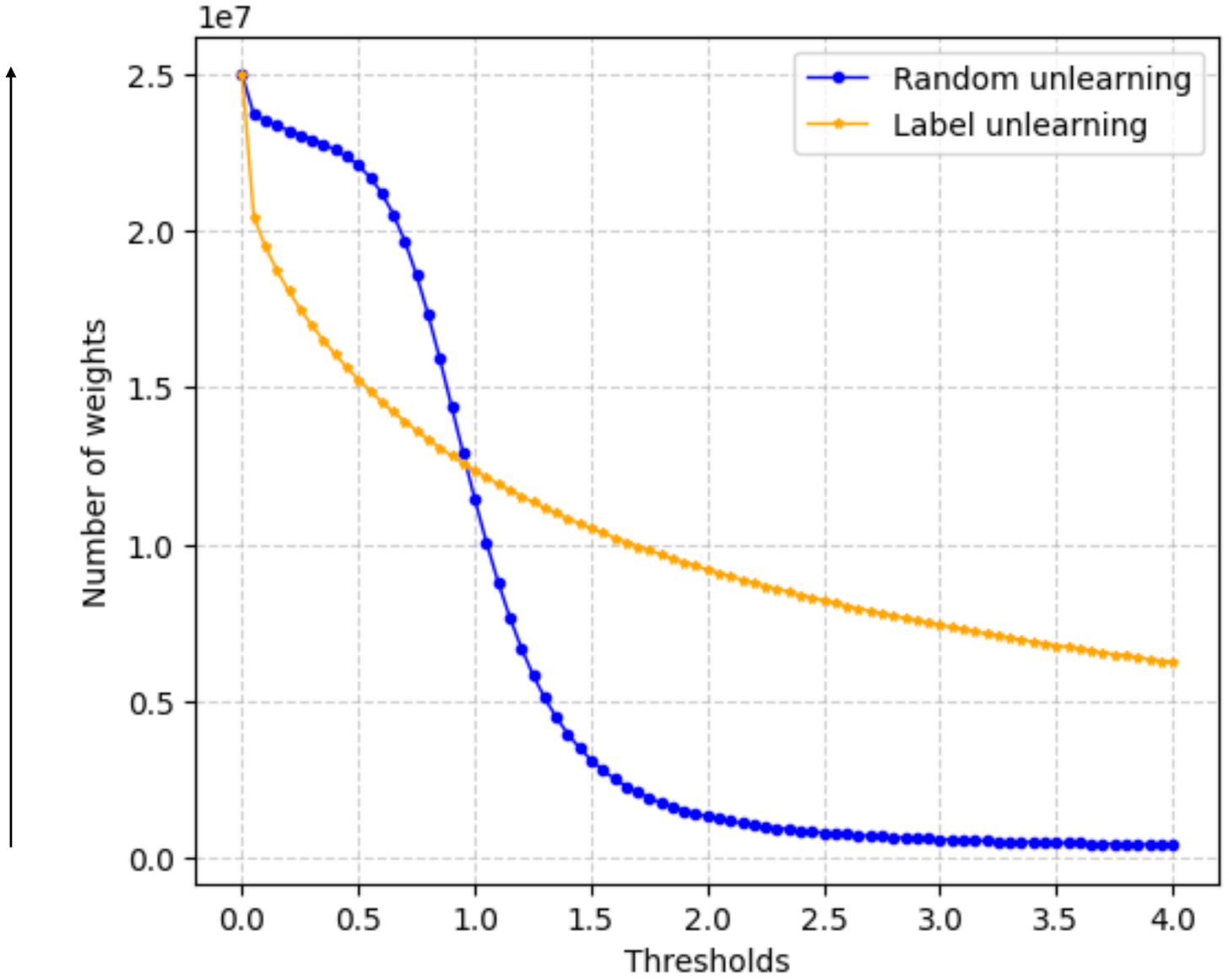
Δ-metrics should be close to zero.

		VGG-16		ResNet-18		ViT	
		RN	Label	RN	Label	RN	Label
Gold	$Acc_{D_r}\%$	92.79	91.78	94.50	95.95	91.61	92.15
	$Acc_{D_f}\%$	92.60	0.00	94.60	0.00	92.46	0.00
	$MIA\%$	73.32	24.68	74.71	22.84	70.91	32.74
	Time s	201	228	284	305	7848	8408
DeepClean	$Acc_{D_r}\%$	95.62	90.92	***98.74***	***98.27***	***99.79***	***96.07***
	$\Delta Acc_{D_f}\%$	***-3.88***	***+0.00***	***-3.68***	***+0.00***	***-2.20***	***+0.00***
	$\Delta MIA\%$	***-2.84***	+3.64	***-5.33***	***-8.04***	+0.79	-6.40
	Time s	***60***	***60***	***71***	***70***	***127***	***126***
Sparse MU	$Acc_{D_r}\%$	88.81	89.87	91.89	92.57	74.04	75.71
	$\Delta Acc_{D_f}\%$	-10.54	+0.00	-8.32	+0.00	-22.38	+0.00
	$\Delta MIA\%$	-9.88	***+1.84***	-12.82	-10.16	-20.83	***-2.58***
	Time s	133	133	153	153	220	220
L-CODEC	$Acc_{D_r}\%$	***99.85***	***99.85***	39.47	57.04	19.55	11.75
	$\Delta Acc_{D_f}\%$	+7.3	+99.96	-56.20	+16.48	-73.26	+32.60
	$\Delta MIA\%$	+17.54	+67.96	+9.15	-11.36	***+0.41***	+67.26
	Time s	245	276	366	479	238	245

The higher we go,

the more of the model

we need to update.



Example from Cifar-10 and VGG-16.

To sum up

A simple machine unlearning method using the Fisher information matrix.

Because of its simplicity, the method can be applied to different models without requiring to track fine-tuning information, e.g., gradients.

Please see the full workshop paper for comparisons with more methods and ablations.

DeepClean: Machine Unlearning on the Cheap by Resetting Privacy Sensitive Weights using the Fisher Diagonal

Jialei Shi[1], Kostis Gourgoulias[2,3], John F Buford[2,3], Sean J Moran[2,3], and Najah Ghalyan[2,3]