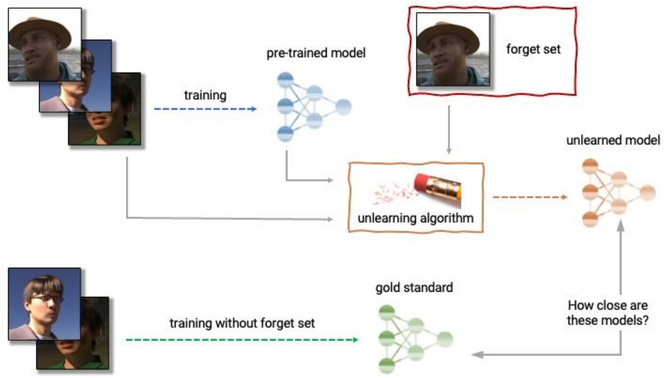
A flaw in the NeurIPS unlearning challenge and an algorithmic framework for entropy regularization

Sumeet Shirgure
University of Southern California

NEURAL INFORMATION PROCESSING SYSTEMS

The unlearning challenge in a nutshell

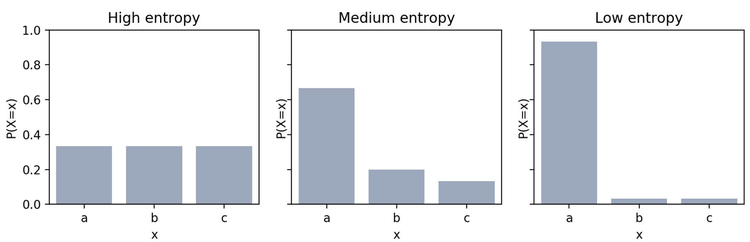
We have a dataset split into “retain” and “forget” subsets. And we have a pre-trained deep neural model trained on “retain+forget” subsets. We also have a model trained solely on the “retain” subset that is considered the gold standard.
The objective of the challenge is to come up with an unlearning algorithm that transforms the pre-trained model into a new model that is “as close to the gold standard” as possible defined by some metrics.
The key oversight that I’ll be talking about is that there is obviously more than one way to train the gold standard (“oracle”) model. It’s difficult to say that a single best method exists. What we’ll show is that there could be a better way than what’s currently prescribed.



NeurIPS unlearning challenge schematic

Entropy over the class labels of classifier

To do that, we must first understand certain trends in deep neural network classifiers – i.e they tend to generate lower amounts of entropy for inputs from the training set [1], [2]. For example, the following output probability distributions over three classes have varying levels of entropy. It’s more likely that the one on the left was not part of the training set and the one on the right was.



Membership inference attacks and entropy regularization

This trend poses a potential security threat, which are exploited by membership inference attacks [1], [2]. E.g we can train a logistic classifier on the entropy of the output distribution to discriminate the examples from the training set vs. outside the training set.

The solution? Employ structural risk minimization. Not on the model’s weights, but rather on the entropy of the output distribution. This amounts to adding a -H(P) term to the loss function of the original model.

For example, the middle distribution could be something that’s generated by a classifier trained with regularization, as opposed to the one on the right. Also note that if we overdo it, like in the left one, all class specific information could be lost. This drops accuracy.

The obfuscation framework

What if we’re given just the pretrained weights of a model that wasn’t trained with entropy regularization and we want to ‘convert’ it into one that behaves like it was trained with regularization? The other contribution of this work is showing that there is a way to do just that, without retraining from scratch.

Algorithm 1: Obfuscate

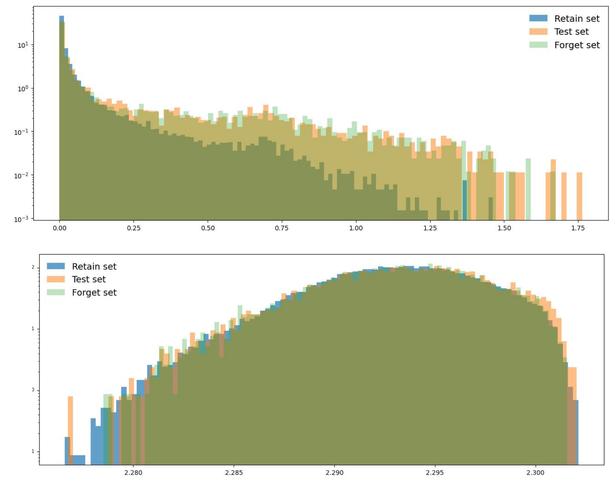
Data: Hyper parameters ϵ, δ, learning rates η, η', regularization weight λ, model parameters Θ, loss function $\mathcal{L}$
Result: Updated model parameters Θ
$\sigma \leftarrow$ average accuracy score of Θ on your dataset
while *average sample entropy* $H(P_\Theta) > \epsilon$ **do**
  $\Theta \leftarrow \Theta - \eta \nabla_\Theta (H(P_\Theta))$
  // note that accuracy goes down in this phase
end
while *accuracy of* $\Theta < \sigma(1-\delta)$ **do**
  $\Theta \leftarrow \Theta - \eta' \nabla_\Theta (\mathcal{L} - \lambda H(P_\Theta))$
end

Let’s call the two while loops phases (a) and (b). (a) does gradient updates corresponding to -H(P) while (b) regains any lost accuracy by retraining with regularization. Note that the accuracy of the classifier can drop arbitrarily in phase (a), and there’s no guarantee that either phase terminates for any given ϵ, δ.
But empirical evidence suggests that for reasonable values of these hyperparameters, phases (a) and (b) should terminate quickly and phase (b) requires far fewer epochs than training from scratch.

Results on unlearning challenge starter kit.

We can apply these ideas to the models provided to us in the unlearning challenge’s starter kit. The results of the experiments are given in the following table. The original model is a ResNet18 pretrained on CIFAR10 (retain+forget sets), the oracle model is trained solely on the retain subset, and the obfuscated model is the original model after undergoing obfuscation. As can be seen in the table, the obfuscated model evades both kinds of membership inference attacks, one between retain and forget sets, and one between forget and test(unseen) sets. And it is just as good at classifying images from the test set. Hopefully this is enough to throw shade on the choice of the oracle model in the unlearning challenge. E.g in the unlearning challenge, had the organizers declared that the oracle model will be trained with regularization the reference distribution would have changed drastically while still performing at comparable accuracy.

Metric	Original model	Oracle model	**Obfuscated model**
Test set accuracy	88.3%	88.0%	88.2%
Average retain set entropy	0.047 ± 0.124	0.043 ± 0.116	$2.293 \pm \mathbf{0.004}$
Average forget set entropy	0.048 ± 0.123	0.145 ± 0.282	$2.293 \pm \mathbf{0.004}$
Average test set entropy	0.137 ± 0.278	0.143 ± 0.280	$2.293 \pm \mathbf{0.004}$
Retain vs. forget MIA acc.	N/A	58.6%	**50.2%**
Forget vs. unseen MIA acc.	57.6%	49.7%	**51.7%**



References

[1] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models, 2017.

[2] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models, 2018.