Machine Unlearning Without Repair using DiStruction

Zack Dugue

September 2024

1 Abstract

Machine unlearning is the task of teaching a machine learning model to “forget” a subset of its training data, known as the forget set, without compromising performance on the rest of the data, known as the retain set. This task has become increasingly important as privacy concerns around the data used to train machine learning models have grown. At the same time, the size of such models makes simply retraining them on the retain set impractical. In order to prevent the degradation in retain set performance caused by doing approximate unlearning, many algorithms use ”repair”. Repair steps involve tuning the model on a subset of the original dataset in order to maintain performance. For reasons related to data security or licensing, the original data may not always be available at the time of unlearning. Our Distributional Reconstruction (DiStruction) method instead works by representing the parameters of the original model as independent Gaussian distributions. To ensure good performance on the retain set through the forgetting process, we introduce Kullback–Leibler divergence penalty between the distributions of the two models' parameters. This allows the model to forget the required data without compromising its performance on the retain set. Our method acts as a plug in replacement for repair and is compatible with multiple unlearning algorithms. We evaluate it on a wide array of baselines and see competitive performance with repair based methods.

2 Introduction

Machine unlearning takes an already pre-trained trained machine learning model and attempting to remove the influence of certain data points used in training, without compromising the models performance on the remaining training data. These two subsets of the dataset are known as the forget set and the retain set. The unlearning task has become relevant as the cost and amount of data necessary to train machine learning models has grown. Without practical unlearning approaches, an entire model could be compromised due to having been trained on copyrighted, toxic, or otherwise unwanted data. Retraining these models from scratch without the undesired data is often prohibitively expensive for large models, hence the need for unlearning algorithms [5].

A naive approach to Machine Unlearning is to simply maximize (rather than minimize) the pre training objective over the forget set. This approach does succeed in inducing unlearning on the forget set, however it comes with the downside of inducing ”catastrophic forgetting”. Catastrophic forgetting is when performance on the retain set falls off dramatically, IE the model ”over forgets”. This occurs because the neural network doesn’t put any emphasis on ”remembering” past information. This can be countered by also training the model to minimize the loss on its original pre-training objective for samples on the retain set. This obviously requires prior access to the retain set. However this is not always the case, for example in situations where data is acquired under a temporary license.

Working in the case where the dataset is discarded after training, we create an algorithm that manages to balance performance on the forget and retain set without need for access to the retain set directly. Our method uses a Bayesian approach to represent parameters as independent Gaussians, and then our method minimizes the KL divergence between the parameter distributions our ”forget” model and the original pretrain model. We verify the performance of our method by testing on several vision-related forgetting tasks and comparing it against baselines.

3 Method

3.1 Preparation Step

The approach of DiStruction is to remove the reliance on the retain set by representing the parameters of our model as random variables with independent gaussian distributions. This requires performing a preparation step, on the model after pre training, but before the training dataset is discarded. Note that, for this preparation step we do not know what our forget set is.

For each of our original parameters θ_o, the mean of the parameter distribution μ is simply equivalent to the original parameter. Thus $\mu := \theta_o$. We compute the standard deviation by minimizing the following. Let $\mathcal{L}$ be our pretraining objective, then:

$\sigma := \text{argmin}\ \{\mathbb{E}_{\theta\sim\mathcal{N}(\mu,\sigma)}[\mathbb{E}_{x,y\sim D}[\mathcal{L}(\psi_\theta(\S),\dagger)]]\}$

Under the assumption that θ_o was already optimal, this optimization problem has a trivial solution at $\sigma = 0$ for all parameters. In practice, we find an approximate non trivial solution using Bayesian Gradient Descent (BGD). BGD has the following update rule:

$$\sigma_{t+1} \leftarrow \sigma_t \cdot \sqrt{1 + \frac{1}{4} \cdot \sigma_t^2 \cdot \mathbb{E}_\epsilon \left[\frac{\partial \mathcal{L}}{\partial \theta_t} \cdot \epsilon\right]^2} - \frac{1}{2} \cdot \sigma_t^2 \cdot \mathbb{E}_\epsilon \left[\frac{\partial \mathcal{L}}{\partial \theta_t} \cdot \epsilon\right] \qquad (1)$$

This training process typically converges to a loss very close to that of the original, non-stochastic, model. Let $p = \mathcal{N}(\mu, \sigma)$ be the learned parameter distribution. We note then that we can recover the original model, before this preparation step, by simply taking the mean of p, which in practice is stored as a buffer in the BGD optimizer. This means that our preparation step should have no effect on model performance at inference time, and does not require the model to be treated as a Bayesian Neural Network for anything other than forgetting.

3.2 Forgetting Step

When given a forget set D_f and a retain set D_r we are then able to perform Distruction. Let U be an unlearning objective computed over a given data point. Such unlearning objectives can take many forms, [4],[1], [3].

For a standard unlearning algorithm with repair we are trying to

$$\text{argmin}_{\theta'}\{(1-\lambda)\mathbb{E}_{(x,y)\sim D_f}[U(\psi'_\theta(x),y)] + \lambda\mathbb{E}_{(x,y)\sim D_r}[\mathcal{L}(\psi'_\theta(x),y)]\}$$

Where λ is some balancing hyper parameter.

In DiStruction we replace the fixed model parameters with a bayesian approach, using the standard deviations learned in our preparation step. Let p be our parameter distribution from the preparation step, and q be the parameter distribution of the forget model we're trying to learn.

We replace the repair term with a reverse KL divergence term between the two parameter distributions, yielding:

$$\text{argmin}_q\{(1-\lambda)\mathbb{E}_{\theta\sim q}[\mathbb{E}_{(x,y)\sim D_f}[U(\psi'_\theta(x),y)]] + \lambda\cdot\text{KL}(q||p)\}$$

With p and q as independent normal distributions, we have that the KL term is equivalent to:

$$\text{KL}(q||p) = \frac{1}{2}\{\frac{(\mu_p-\mu_q)^2}{\sigma_p^2} + \frac{\sigma_q^2}{\sigma_p^2} - 1 + \log(\frac{\sigma_p^2}{\sigma_q^2})\}$$

You can split this into two smaller terms. The first terms, depend on the squared difference between the means of the two parameters. This is scaled by the inverse of the variance of the parameter for the original model. This makes it so parameters with a low variance (and thus high importance) in the original model, tend to stay close to their original means. The remaining terms ensure that parameter importance does not drift too significantly.

We compute the unlearning objective using the Bayesian Gradient Descent algorithm, and compute the Kullback-Liebler objective (and its associated parameter updates) analytically using the means and standard deviations for the two models.

This gives us a derivative of

$$\frac{\partial\text{KL}(q||p)}{\partial\mu_q} = \frac{\mu_p-\mu_q}{\sigma_p^2}$$

$$\frac{\partial\text{KL}(q||p)}{\partial\sigma_q} = \frac{\sigma_q}{\sigma_p^2} - \frac{1}{\sigma_q}$$

4 Results

Model	metric	Methods							DiStruction
		baseline	retrain	finetune	teacher	UNSIR	amnesiac	SSD	
ViT	$\mathcal{D}_r$	88.88±0.00	90.07±0.09	80.82±1.37	87.46±0.53	88.47±0.89	87.92±0.89	88.90±0.00	88.86±0.00
	$\mathcal{D}_f$	94.70±0.00	0±0.00	.46±0.72	4.20±5.24	65.32±9.11	0±0.00	0±0.00	0±0.00
	MIA	94.40±0.00	3.23±0.50	19.00±0.09	0.03±0.00	29.13±0.06	1.00±0.01	1.80 ± 0.00	0.00 ± 0.00

Table 1: Results for full class forgetting on CIFAR100 using a Vision Transformer on. $\mathcal{D}_r$ refers to retain set accuracy. $\mathcal{D}_f$ refer to forget set accuracy. MIA refers to membership inference attack accuracy.

Model	metric	Methods							DiStruction
		baseline	retrain	finetune	teacher	UNSIR	amnesiac	SSD	
ViT	$\mathcal{D}_r$	95.73±0.00	94.61±0.13	85.70±3.05	93.60±0.29	93.34±0.45	93.47±0.22	95.13±0.00	95.16
	$\mathcal{D}_f$	94.53±0.00	22.26±8.34	6.25±6.03	3.35±2.89	74.93±10.13	0.85±1.71	5.12±0.00	.08
	MIA	80.40±0.00	3.44±0.01	16.04±0.03	0.02±0.00	27.27±0.14	0.78±0.00	5.40±0.00	.02

Table 2: Results for sub-class forgetting on CIFAR20 using a Vision Transformer. $\mathcal{D}_r$ refers to retain set accuracy. $\mathcal{D}_f$ refers to forget set accuracy. MIA refers to membership inference attack accuracy.

Model	metric	Methods						DiStruction
		baseline	retrain	finetune	teacher	amnesiac	SSD	
ViT	$\mathcal{D}_r$	98.88±0.00	98.61±0.08	97.28±0.33	97.58±0.36	97.62±0.35	98.01±1.56	98.353
	$\mathcal{D}_f$	100.00±0.00	98.80±0.76	97.19±0.98	86.75±3.57	73.49±5.11	98.07±2.35	96.36
	MIA	90.76±0.03	91.77±0.02	86.14±0.02	33.53±0.06	10.44±0.05	85.54±0.11	.81

Table 3: Results for random class forgetting on CIFAR10 using a Vision Transformer. $\mathcal{D}_r$ refers to retain set accuracy. $\mathcal{D}_f$ refers to forget set accuracy. MIA refers to membership inference attack accuracy.

We ran exclusively vision-based forgetting experiments using a vision transformer model. The 3 tasks we evaluate are full class forgetting on CIFAR10, subclass forgetting on CIFAR20, and random forgetting on CIFAR100.

It is worth noting that, especially in **??**, the resulting Membership Inference Attack score is extremely low for our method. We believe that this is because of our entropy-based loss function 'hacking' this metric, more so than the exact advantages of our method. Using the D_r and D_f metrics, we see that our method consistently outperforms the other methods, including the state of the art method SSD, [2].

We also do some investigations into how our method works under the hood. In figure 1 we see how the KL divergence penalty changes over the course of training iterations during the forgetting process, where each curve represents a different λ value. While not shown in the figure, we found empirically model achieves the unlearning objective very quickly (usually only a handful of iterations), while the KL divergence penalty changes substantially over the course of the unlearning process. As is evident in the diagram, halting training too early can cause the model to have a larger KL divergence, and thus a worse performance on the retain set.

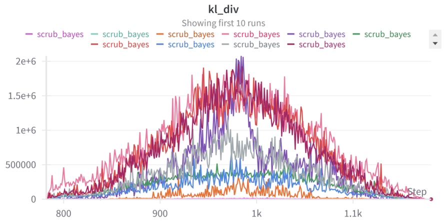

Figure 1: A Plot of the KL divergence penalty between the original model and the forgetting model vs training iterations on the full class forgetting task. Each line represents a different unlearning run with a different λ value.

5 Discussion

We show a promising method for replacing repair in machine unlearning algorithms. Our method shows minimal degredation in performance when compared to repair, without requiring access to the training dataset during unlearning. This independence from training data allows previously repair-based machine unlearning algorithms to be usable on models in cases where the weights have been distributed externally or where the data is on a temporary license.

Our method also provides empirical backing to the hypothesis that the representation of model parameters as independent gaussians learned by the BGD algorithm is relatively accurate. Many algorithms which require determining parameter importance focus on measuring curvature through squared gradients, or through non adaptive monte carlo methods. We believe our results show a promising new direction for using the standard deviation learned by BGD to estimate these importances instead. While some methods for determining parameter importance like Synaptic Intelligence [6], peturb parameters by a fixed amount and then compute how strong the gradient is in pushing the parameter back to its original position. The Standard Deviations from BGD can be thought of as an adaptive version of this; it uses a running estimation of parameter importance to only perturb important parameters slightly, and unimportant parameters a lot. This dynamic importance estimations helps us find accurate importances for each of the parameters, which aids in preserving retain set performance during the unlearning phase.

Our investigation was limited to the vision domain and to non generative applications, as is standard for unlearning research. However we see extending our method in generative applications as an exciting future direction.

6 Conclusion

Many machine unlearning algorithms require access to the training data when unlearning is performed. We have created a novel method that circumvents this requirement and performs repair by minimizing the KL divergence between the distribution of model parameters trained on the original data and the model parameters tuned with the unlearning algorithm. We can do this with a minimal reduction in performance compared to repair methods.

References

[1] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher, 2023.

[2] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening, 2023.

[3] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning, 2020.

[4] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning, 2023.

[5] Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning, 2024.

[6] Friedemann Zenke et al. Continual learning through synaptic intelligence. In *Proceedings of Machine Learning Research*, volume 70, pages 3987–3995, 2017.