SDU
Syddansk Universitet

Machine Unlearning

Dr. Vinay Chakravarthi Gogineni
Assistant Professor, Applied AI and Data Science, TEK, SDU | 21. August 2025

Why Unlearning?

- AI has brought transformative benefits yet raised privacy concerns.
- Personal images in social media may result in:
  - Unwanted surveillance,
  - Leakage of demographic information.
- Simply deleting images is insufficient. AI models trained on them can still reveal personal information.
- Right to Erasure (e.g., GDPR Art. 17) empowers users to have control over their data.
- Users can request deletion of both their raw data and the knowledge derived from it in trained models.

Facial Recognition System

Reconstruction via Model Inversion

What is Unlearning?

- Let M_θ be an already trained model on dataset $\mathcal{D}$.
- A request is made to erase the knowledge of forget set $\mathcal{D}_f$, where $\mathcal{D}_f \subset \mathcal{D}$.

Unlearning

Obtain model $M_{\theta'}$ that behaves as if it has never seen data from $\mathcal{D}_f$.

- However, erasing the knowledge of $\mathcal{D}_f$ from $M_\theta$can inadvertently remove information learned from the retained data $\mathcal{D}_r$ ($\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$) as well.

Goal

Obtain $M_{\theta'}$ that exhibits low utility (e.g., accuracy, F1 score) on $\mathcal{D}_f$ and maintain high utility on $\mathcal{D}_r$.

Types of Unlearning

- Unlearning types:
  - Exact vs. Approximate,
  - Probabilistic (distributional closeness).
- Exact Unlearning:
  - Certifies that the model $M_{\theta'}$ did not use forget data.
  - Retraining from Scratch: Fully retrain the model from scratch on the retain set $\mathcal{D}_r$, ensuring complete removal of the forget set $\mathcal{D}_f$. Considered to be Gold Standard. Guarantees exact unlearning but is computationally expensive.
  - SISA (Sharded, Isolated, Sliced, and Aggregated) is a computationally-efficient exact unlearning approach.

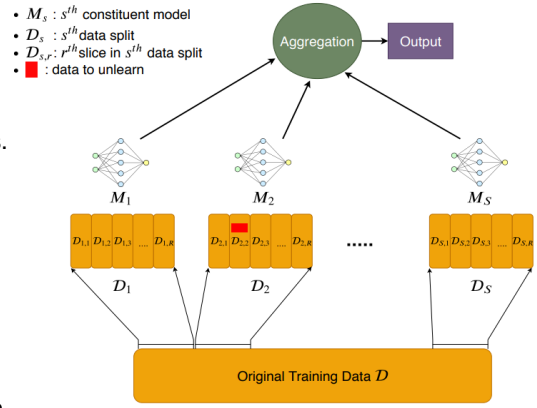
SISA (Sharded, Isolated, Sliced, and Aggregated)

Learning Phase:

- Dataset is split into **Shards**, each further divided into **Slices**.
- Each Shard model is trained by sequentially adding Slices.
- Shard models are aggregated to form the final model.
- Model parameters are saved after each Slice.

Unlearning Phase:

- Only models with Shards containing forget data are retrained.
- Retraining starts from parameters saved before adding the Slice with forget data.



Approximate Unlearning

- Adjust the original model M_θ so that the knowledge of forget data is erased without the need for full retraining.
- These methods may not guarantee complete forgetting but provide a more computationally feasible alternative.
- Some of the existing works are:
  - Amnesia Unlearning,
  - Label Flipping,
  - Optimization-based unlearning.

Amnesiac Unlearning

Learning Phase:

- Maintain a record which training examples appear in each batch. The model updates from each batch are logged and stored.
- The trained model M_θ is obtained by training for E epochs, each comprising B batches, i.e.,

$$M_\theta = M_{\text{initial}} + \sum_{e=1}^{E} \sum_{b=1}^{B} \Delta M_{\theta_{\text{e,b}}}$$

Unlearning Phase:

batches containing the forget data upon a removal request.

- Obtain M'_θ by subtracting the model updates corresponding those batches from M_θ.

$$M'_\theta = M'_\theta - \sum_{e=1}^{E} \sum_{b \in \mathcal{F}_b} \Delta M_{\theta_{\text{e,b}}}$$

Approximate Unlearning

- **Label Flipping:**
  - The D_f is relabeled with randomly selected incorrect labels.
  - The model is then retrained for a few iterations on this modified dataset.
- **Negative Gradient:**
  - Perform **gradient ascent** on the forgotten dataset D_f for a few iterations.
  - This reverses the effect of the data, pushing the model parameters away from what was learned from D_f.
- **Limitations:**
  - May cause *catastrophic forgetting*, where the model unintentionally loses useful information from D_r.
  - A brief retraining phase is required after unlearning to restore model performance.

Solution: Layer-wise Partial Unlearning

Layer-wise Partial Amnesia Unlearning

Motivation: Early layers in DNNs capture generic features that are shared across classes.
So they must be handled with care.

Learning Phase:

- Store selected updates per layer instead full update; pruning percentage decreases with network depth.
- Let $\Delta M_{\theta_{\text{l,e,b}}}$ denote the update corresponding to the lth layer in the eth epoch in bth batch. then just store:

$$\widehat{\Delta M_{\theta_{\text{l,e,b}}}} = \mathbf{P}_l \odot \Delta M_{\theta_{\text{l,e,b}}},$$

- Here, P_l is the pruning matrix whose elements are either 0 or 1.

Unlearning Phase:

- Subtracts the stored pruned updates $\widehat{\Delta M_{\theta_{\text{e,b}}}}$ corresponding to the forget data.
- Let $\mathcal{F}_b$ denotes the set of batches containing the data samples of forget data, then

$$M'_\theta = M_\theta - \sum_{e=1}^{E} \sum_{b \in \mathcal{F}_b} \widehat{\Delta M_{\theta_{\text{e,b}}}}$$

Evaluation Metrics

Unlearning Efficiency

- **Unlearning time:** Time taken for unlearning.
- **Recovery time:** Time taken for repairing degradation on D_r.
- **Relearn time:** Time taken after unlearning to recover on D_f.

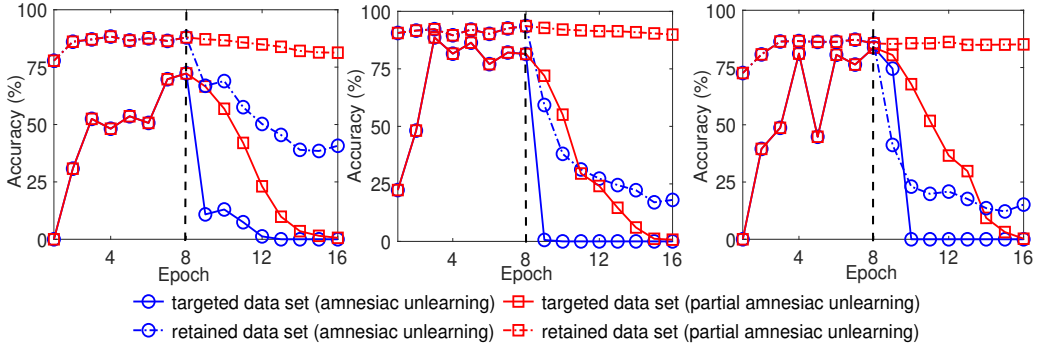
Unlearning Efficacy

- **Membership Inference Attack:** Determines if information about forgotten samples remains.
- **Layer wise distance:** Weight difference between original and unlearned model.

Model Utility

- **Retain Accuracy:** Accuracy on D_r.
- **Forget Accuracy:** Accuracy on D_f.
- **Any Performance metrics:** F1 Score, MSE, etc.

Experimental Results



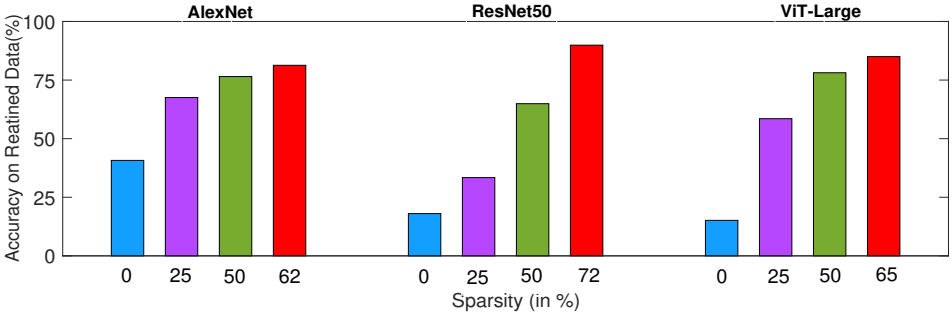
Comparison of model behavior between partial amnesiac unlearning and conventional amnesiac unlearning during training and unlearning phases. AlexNet (left), ResNet50 (middle), ViT-Large (right) architectures.

Experimental Results

Network Architecture	Method	Performance Metrics				
		$\mathcal{D}_t \downarrow$	$\mathcal{D}_r \uparrow$	Class 0 $\uparrow$	Class 6 $\uparrow$	Class 10 $\uparrow$
AlexNet	amnesiacML	0	40.73	1.25	99.93	1.27
	Proposed	0	81.30	36.67	98.78	74.52
ResNet50	amnesiacML	0	18.03	37.45	0	0
	Proposed	0	89.89	83.88	95.98	92.03
ViT-Large	amnesiacML	0	15.15	3.37	0	0
	Proposed	0	85.00	83.78	97.19	65.23

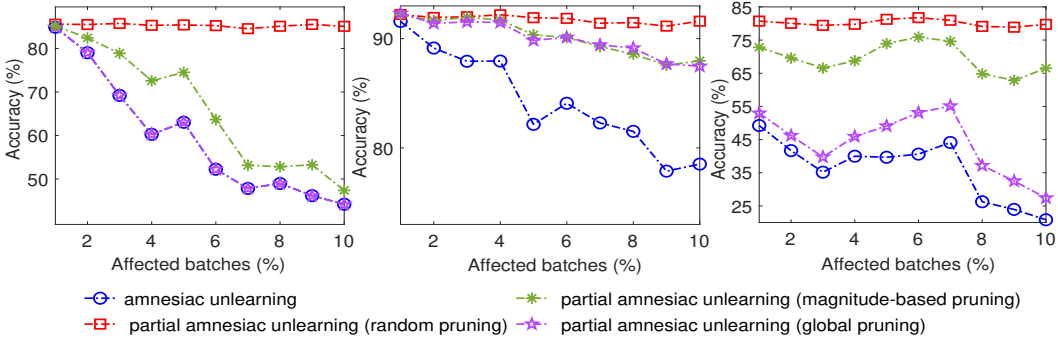
Tabelle: Comparison of model efficacy between partial amnesiac unlearning and conventional amnesiac unlearning on targeted and retained classes.

Experimental Results



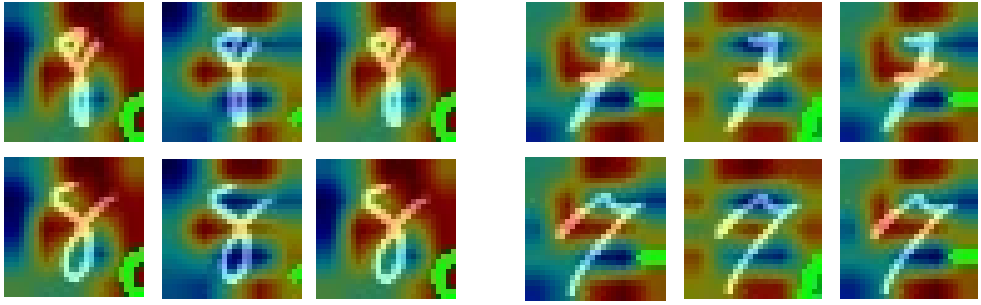
Performance of partial amnesiac unlearning with random pruning on retained data set versus amount of pruning applied to model updates. Zero sparsity corresponds to conventional amnesiac unlearning. The sparsity associated with the red colour bar represents the maximum amount of pruning that results in zero accuracy on targeted data.

Experimental Results



Comparison of model efficacy between partial amnesiac unlearning between conventional amnesiac unlearning against percentage of affected batches. AlexNet (left), ResNet50 (middle), ViT-Large (right) architectures. The vertical dashed line marks the unlearning request initiation.

Experimental Results



First, second, and third column images represent the class activation maps before unlearning, after applying conventional amnesiac unlearning, and proposed partial amnesiac unlearning, respectively.

Reference

1. L. Bourtoule et al., *Machine Unlearning,* IEEE Symposium on Security and Privacy (SP), 2021.
2. Graves, Laura Nagisetty, Vineel Ganesh, Vijay. *Amnesiac Machine Learning*, in AAAI Conference on Artificial Intelligence, 2021.
3. A. Golatkar, A. Achille and S. Soatto, *Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
4. Vinay Chakravarthi Gogineni and Esmaeil S. Nadimi, *Efficient Knowledge Deletion from Trained Models through Layer-wise Partial Machine Unlearning*, under review in Journal of Machine Learning Research (JMLR).

Thank You!