Machine Unlearing using Forgetting Neural Networks

Amartya Hatua
Filip Cano
Trung T. Nguyen
Andrew H. Sung

Motivation

- 2017: Forgetting Neural are first described, inspired by early findings about memory in humans, with no clear use 📖.
- Early 2020's: Machine Unlearning becomes a popular problem

"The hammer before the nail"

The Machine Unlearning Problem

- A : training algorithm (possibly randomized)
- D dataset, $D = R \cup F$, where R is the retain set and F the forget set
- $\theta = A(D)$ is the original model, $\theta^* = A(R)$ is the perfect unlearning model
- Unlearning produces $\tilde{\theta} = U(\theta, F)$, with the following two goals:
    - Utility: Performance on the distribution of R is retained
    - Unlearning: $\tilde{\theta}$ contains no information about F

Membership Inference Attacks

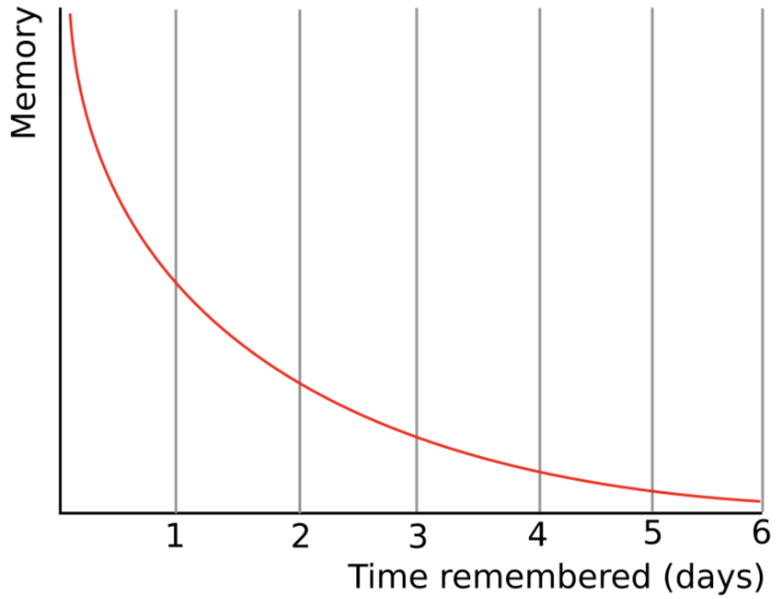
We evaluate unlearning through membership inference attack (MIA)

- Adversary has black box access to trained model f_θ, and has to learn whether $(x, y) \in F$.

- Loss-based MIA: $\Lambda(x, y) = \frac{\Pr[L(f_\theta(x), y) \mid (x, y) \in F]}{\Pr[L(f_\theta(x), y) \mid (x, y) \notin F]}$

  - If $\Lambda(x, y) > threshold$ → $(x, y) \in F$

Forgetting Neural Networks (FNN)



- Inspired on Ebbinghaus forgetting curve (late 1800's)

- Idea: add dampening factors to the NN parameters

$\Sigma_{[\theta_w,\theta_b]}(x;t) = \sigma((\theta_w \cdot x + b) \cdot \boldsymbol{\varphi}(\boldsymbol{t}))$, with $\varphi(t) = e^{-t/\tau}$

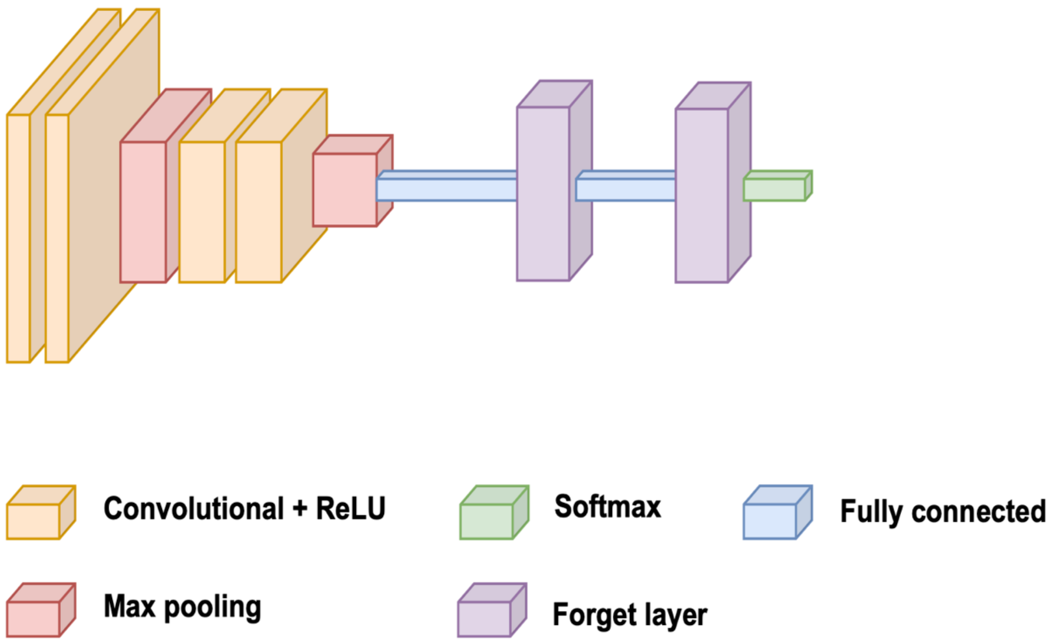
Targetting What to Forget

- If you dampen everything, then $\lim_{\{t \to \infty\}} \Sigma(x;t) = 0$
- Idea: dampen neurons based on their activation on the forget set

Activation level: $A_j(F) = \frac{1}{|F|} \sum_{x \in F} |\sigma(z_j(x))|$

Targeting Where to Forget (Forgetting Layers)

- Forgetting all layers uniformly may be too much
- Forgetting is targeted to the last connected layers



Targeting Where to Forget (Forgetting Rate Variants)

Forgetting neuron is $\Sigma_{[\theta_w,\theta_b]}(x;t) = \sigma((\theta_w \cdot x + b) \cdot \boldsymbol{\varphi}(\boldsymbol{t}))$, with $\varphi(t) = e^{-t/\tau}$

- Fixed forgetting rate (FFR): Forgetting everything at the same time
- Varying forgetting rate (VFR): Apply different forgetting rate (τ) according to activation levels. Four variants:
    - **Rank forget rate**: the j-th most activated neuron forgets proportional to j
    - **Top N**: the N most activated neurons forget with a fixed rate τ
    - **Fixed order forget rate**: pre-defined forget rate, fixed before execution
    - **Random**: forget rates are randomised

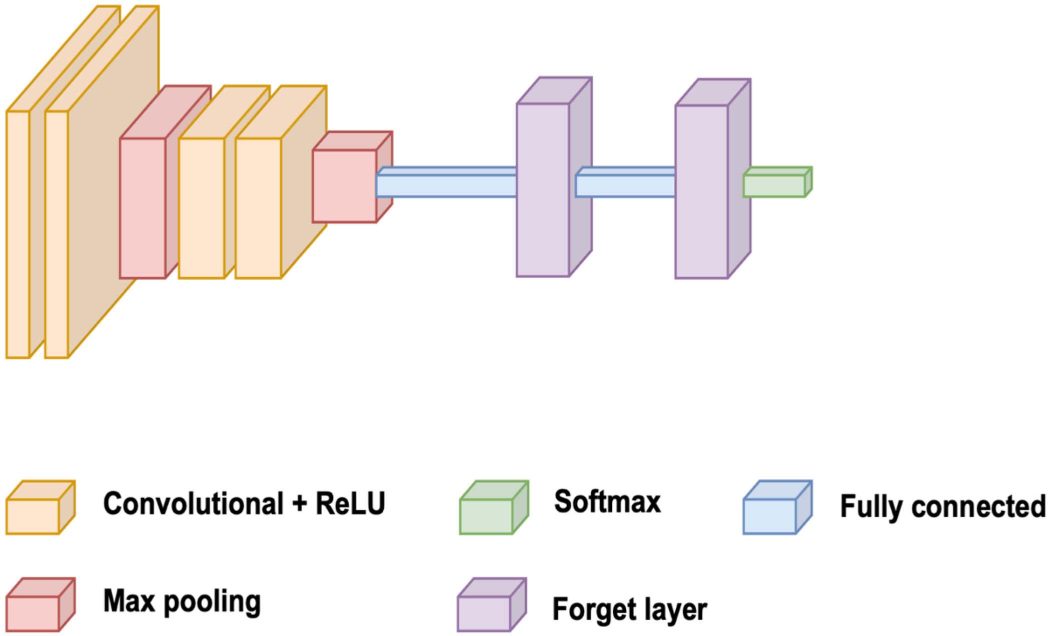
Machine Unlearning Algorithm

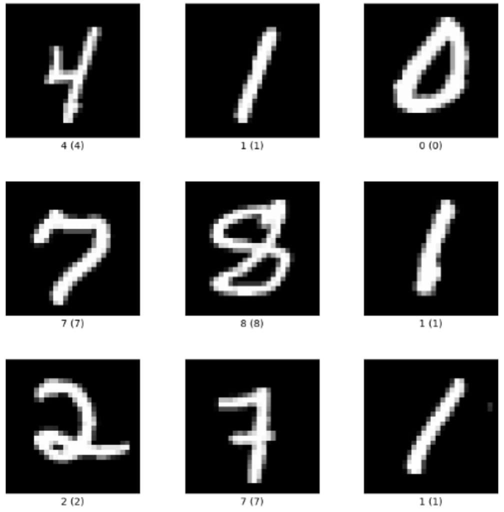
- Apply learning-unlearning bouts
  - Train on retain set
  - Calculate activations and apply forgetting
  - Test MIA on forget set

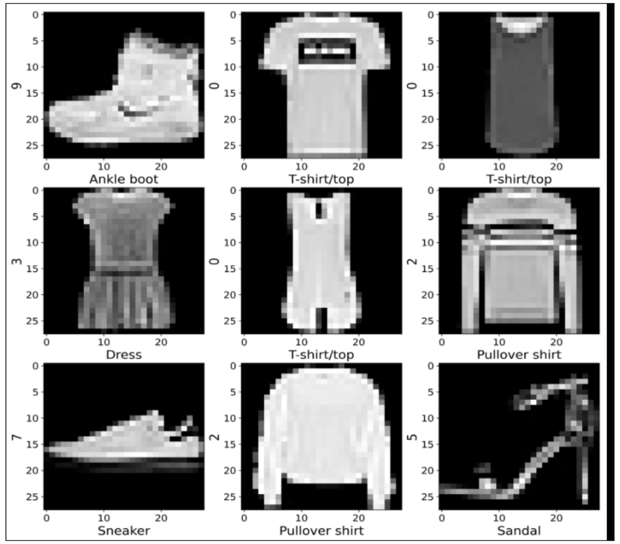
```
Input: training_data, testing_data,
       retain_data, forget_data
Train model once on training_data;
for turn ← 1 to N_turns do
    /* Learning bout                          */
    for epoch ← 1 to training_epochs do
        I) Train model on retain_data;
        II) Evaluate accuracy on testing_data;
        III) Evaluate MIA on forget_data;
    end
    /* Unlearning bout                        */
    for epoch ← 1 to forget_epochs do
        I) Present forget_data and apply
         forgetting functions φ(t = epoch);
        II) Evaluate accuracy on testing_data;
        III) Evaluate MIA on forget_data;
    end
end
```

How Well Does it Work? Experimental Setup

- MNIST HDR and MNIST Fashion
- Convolutional + fully connected network

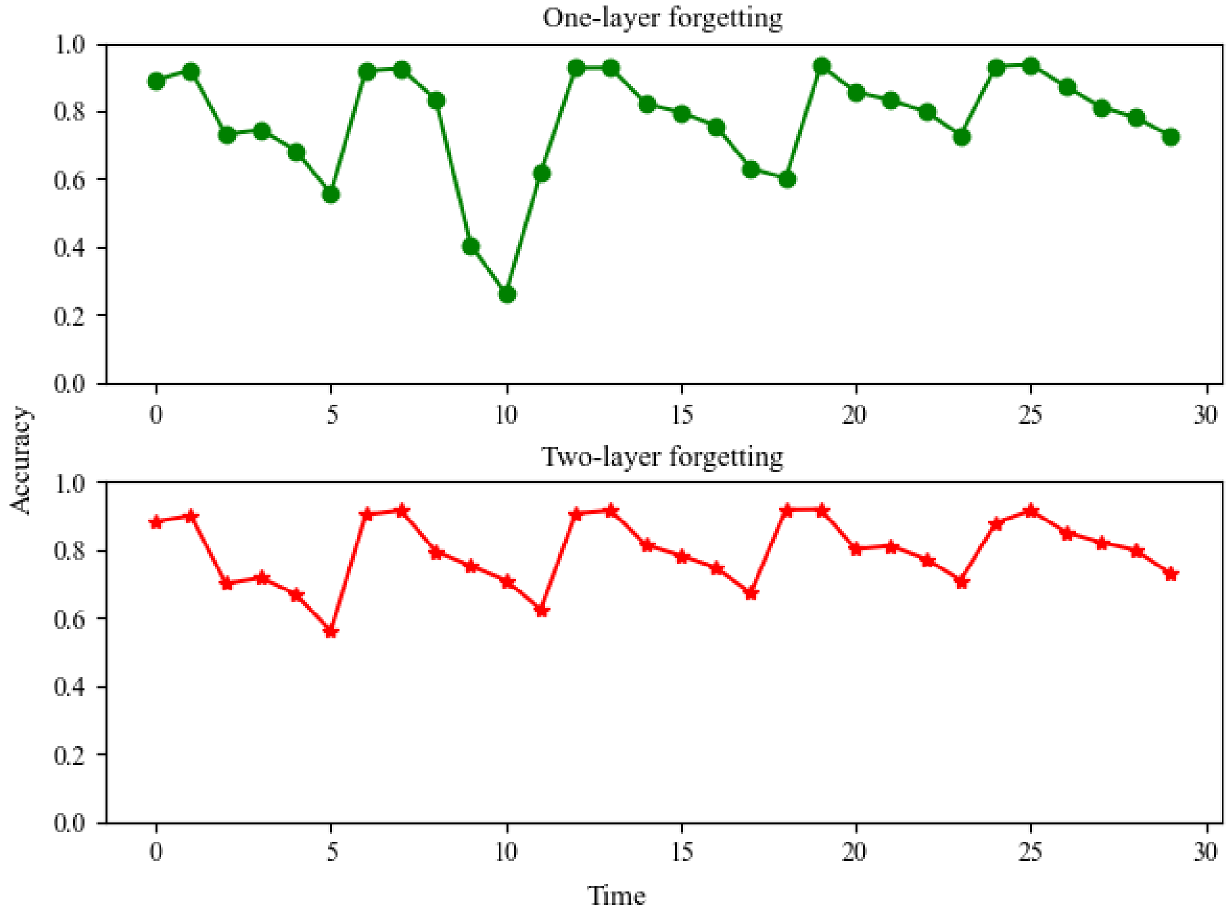




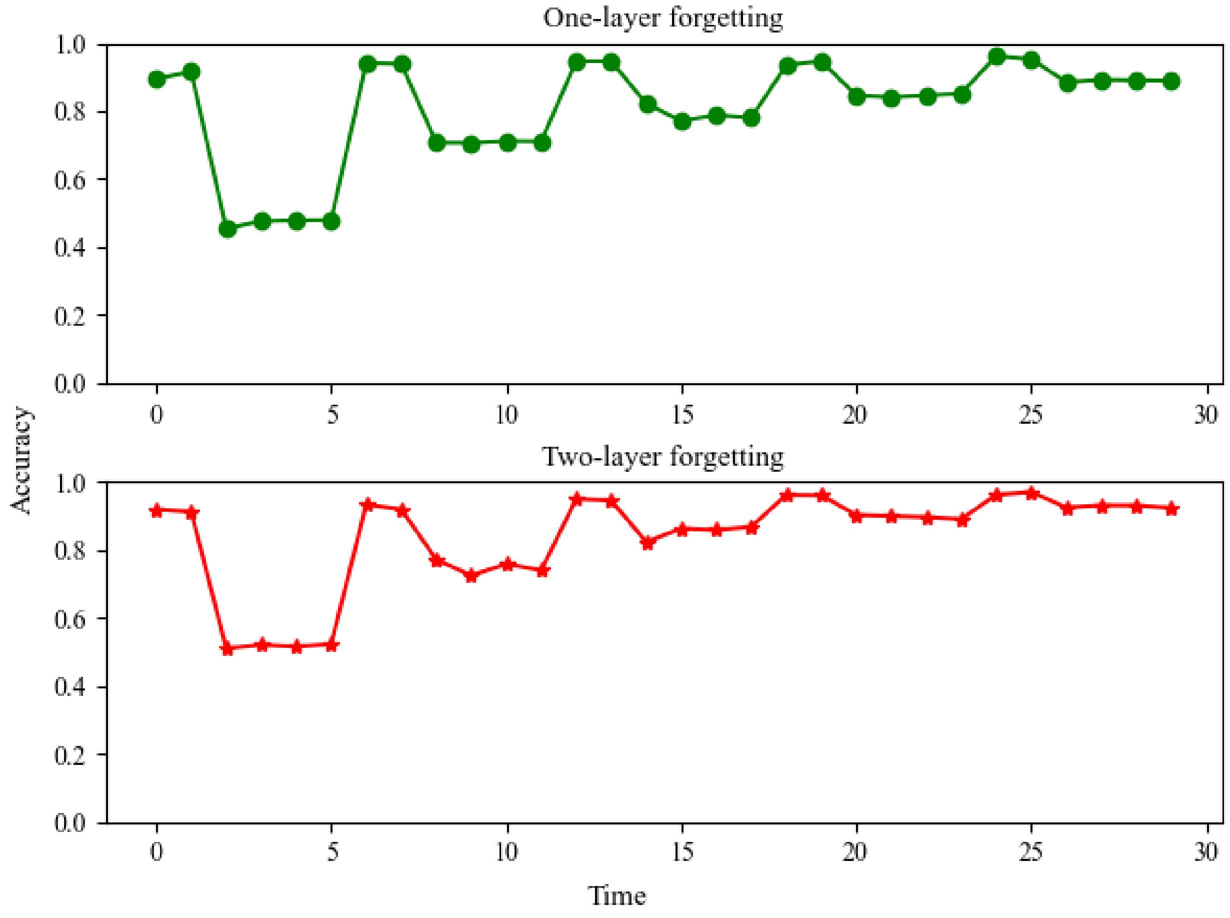


Accuracy: Fixed Forget Rate

Learning-forgetting curves



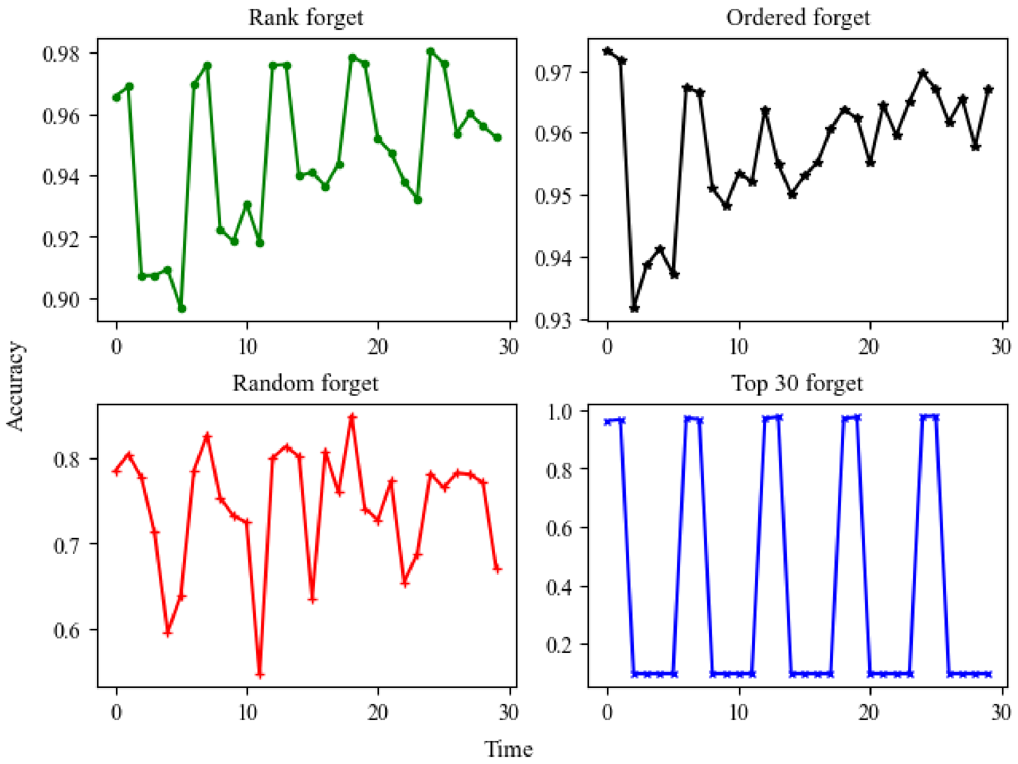
Digit



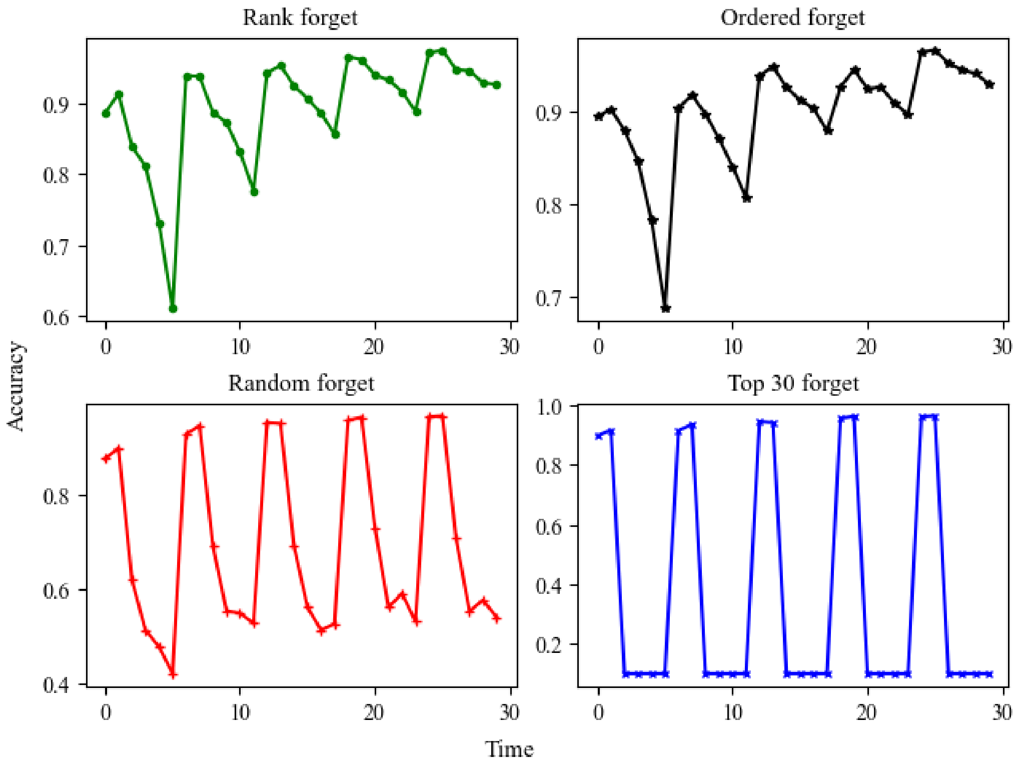
Fashion

Accuracy: Varying Forgetting Rate

Learning-forgetting curves for MNIST Fashion



1 Forget Layer

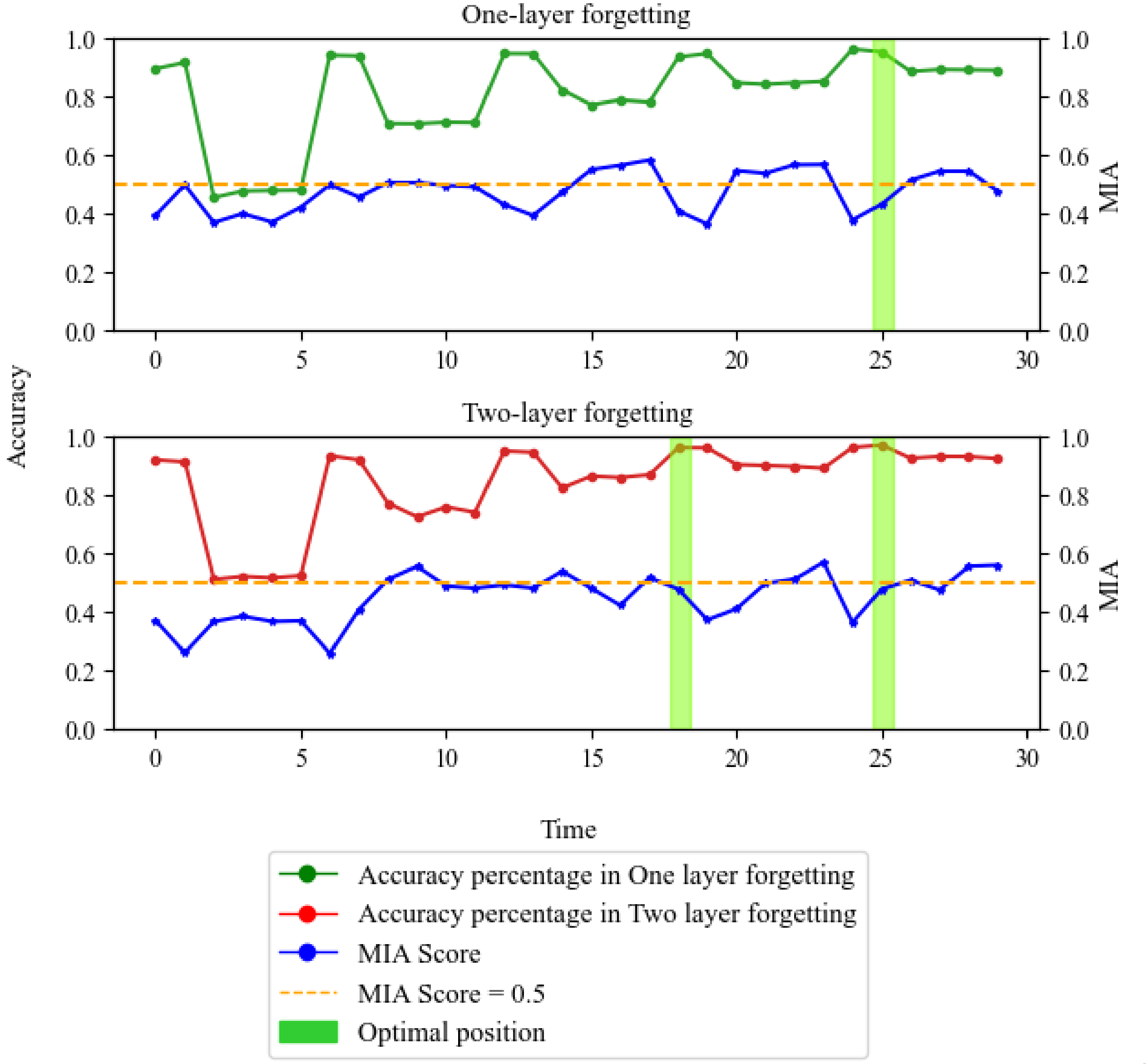


2 Forget Layer

Rank Forgetting works best

Experimental Results: MIA on Fixed Forgetting Rate

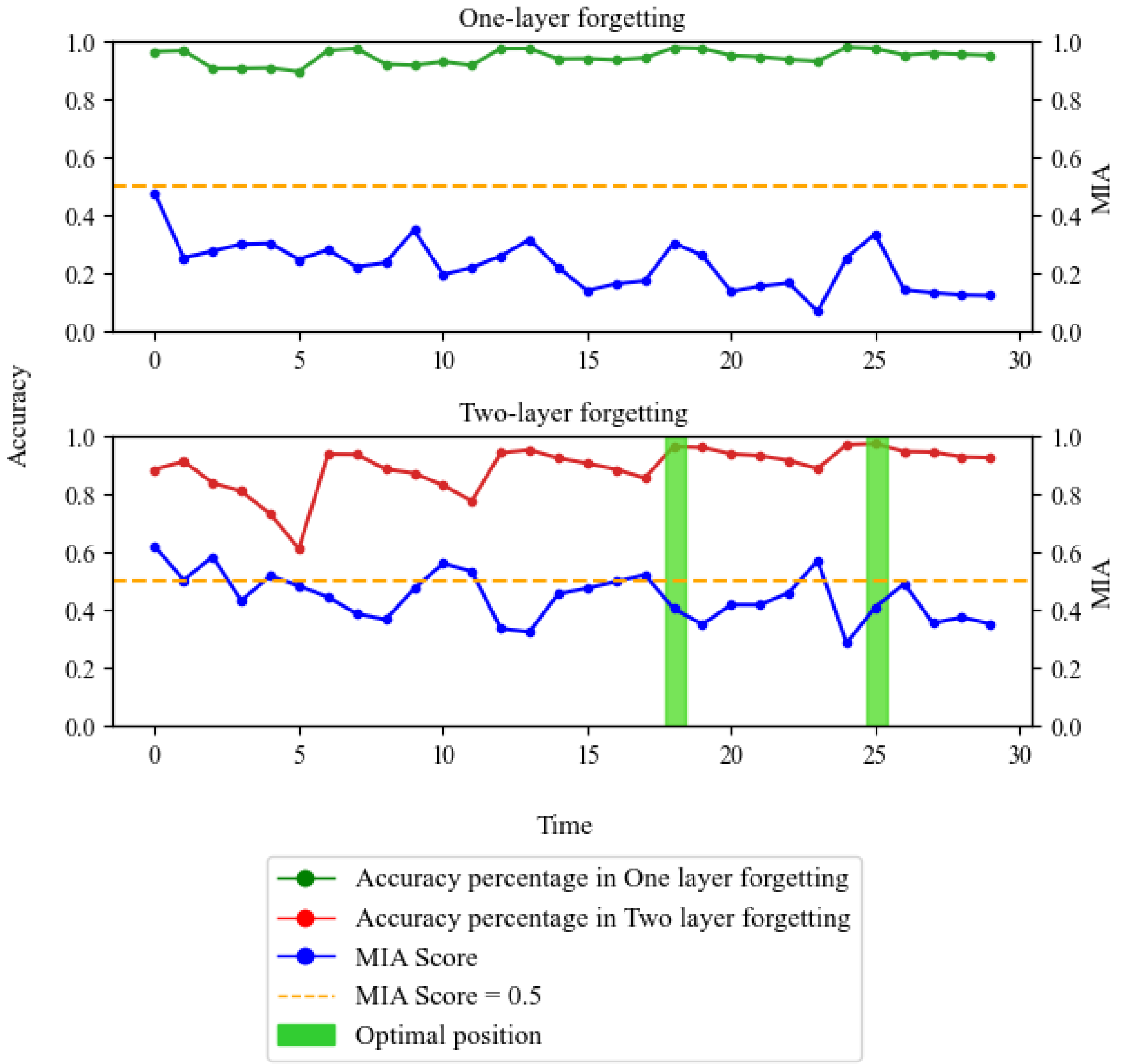
- Accuracy vs MIA score over time
- MIA score = 0.5 is “ideal”
- Optimal positions: high accuracy, MIA score close to 0.5 on forget set



MIA on Varying (Rank) Forgetting Rate

- Accuracy vs MIA score over time
- MIA score = 0.5 is "ideal"
- Optimal positions: high accuracy,

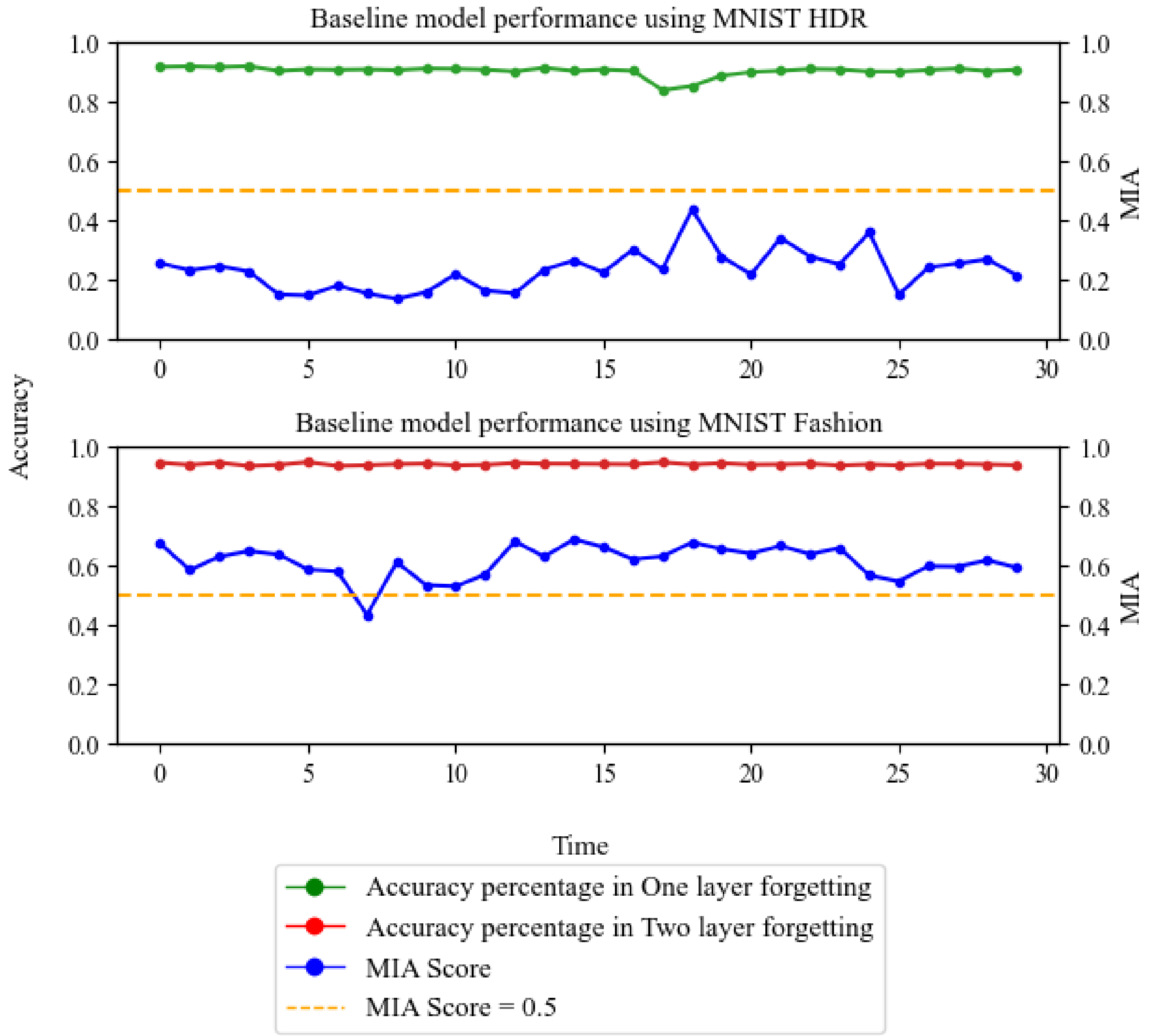
  MIA score close to 0.5 on forget set



MIA on Baseline (full retraining)

- Accuracy vs MIA score over time
- MIA score = 0.5 is “ideal”
- Optimal positions: high accuracy,

  MIA score close to 0.5 on forget set



Discussion / Limitations / Future Work

- ML inspired in nature can be useful, but machines do not work as brains
- The over-forgetting phenomenon: forgetting too much
- No formal guarantees on accuracy or required epochs

Future Work:

- Other Varying Forget Rate methods
- Scale beyond MNIST

Concluding Remarks

- Machine Unlearning is the problem of efficiently “forgetting” training data.
- Forgetting Neural Networks dampen the signal of concrete neurons
  - By targeting the neurons most activated by the data to forget, FNNs can be an efficient method for machine unlearning
  - After ~20 epochs, accuracy over 95% with MIA score close to 0.5 on MNIST Digit and Fashion.

This work has been supported by the European Research Council under Grant No.: ERC-2020-AdG 101020093.