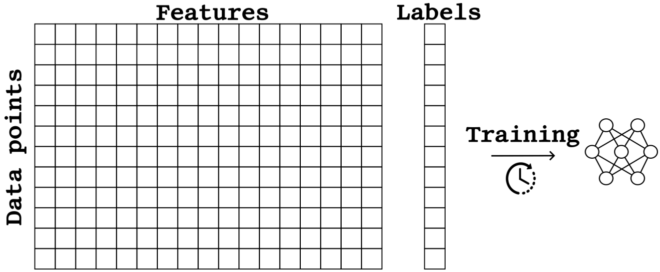
Machine Unlearning of Features and Labels

Alexander Warnecke[1], Lukas Pirch[1], Christian Wressnegger[2], Konrad Rieck[1]

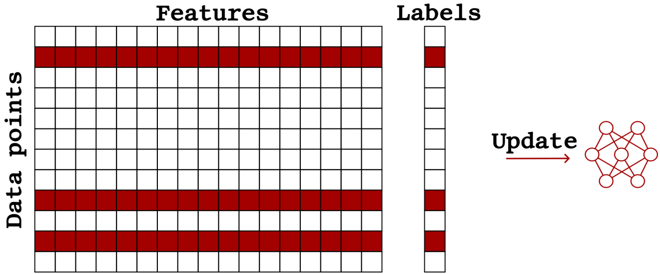
[1]Technische Universität Berlin
[2]Karlsruhe Institute of Technology

Machine Learning



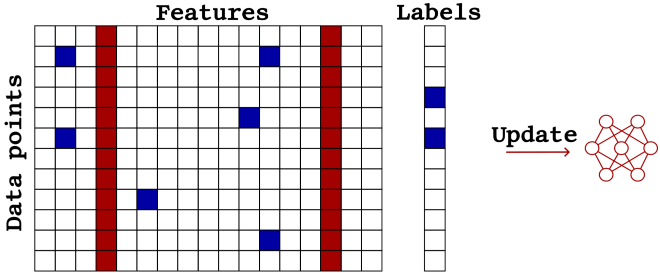
Machine Unlearning

- Algorithms to remove information from ML models
  - Necessary to fulfill privacy policies like GDPR or CCPA
  - So far, removal of entire datapoints



Machine Unlearning

- Algorithms to remove information from ML models
  - Necessary to fulfill privacy policies like GDPR or CCPA
  - So far, removal of entire data points
- We extend the concept of Unlearning to Features and Labels



Approach

- Input given by model and its parameters θ^*
- Framework for unlearning: $\theta = \theta^* + \mathcal{U}(Z, \tilde{Z})$
  - Z contains the datapoints to be fixed, $z = (x, y)$
  - $\tilde{Z}$ contains the corrected datapoints $\tilde{z} = (x + \delta_x, y + \delta_y)$

Approach

- Input given by model and its parameters θ^*
- Framework for unlearning: $\theta = \theta^* + \mathcal{U}(Z, \tilde{Z})$
  - Z contains the datapoints to be fixed, $z = (x, y)$
  - $\tilde{Z}$ contains the corrected datapoints $\tilde{z} = (x + \delta_x, y + \delta_y)$
- Difference in gradients of loss used as basis

$$\Delta(Z, \tilde{Z}) = \sum_{\tilde{z} \in \tilde{Z}} \ell(\tilde{z}, \theta^*) - \sum_{z \in Z} \nabla \ell(z, \theta^*)$$

Approach

- Input given by model and its parameters θ^*
- Framework for unlearning: $\theta = \theta^* + \mathcal{U}(Z, \tilde{Z})$
  - Z contains the datapoints to be fixed, $z = (x, y)$
  - $\tilde{Z}$ contains the corrected datapoints $\tilde{z} = (x + \delta_x, y + \delta_y)$
- Difference in gradients of loss used as basis

$$\Delta(Z, \tilde{Z}) = \sum_{\tilde{z} \in \tilde{Z}} \ell(\tilde{z}, \theta^*) - \sum_{z \in Z} \nabla \ell(z, \theta^*)$$

- $\mathcal{U}(Z, \tilde{Z}) = -\tau \Delta(Z, \tilde{Z})$ (First-Order)

Approach

- Input given by model and its parameters θ^*
- Framework for unlearning: $\theta = \theta^* + \mathcal{U}(Z, \tilde{Z})$
  - Z contains the datapoints to be fixed, $z = (x, y)$
  - $\tilde{Z}$ contains the corrected datapoints $\tilde{z} = (x + \delta_x, y + \delta_y)$
- Difference in gradients of loss used as basis

$$\Delta(Z, \tilde{Z}) = \sum_{\tilde{z} \in \tilde{Z}} \ell(\tilde{z}, \theta^*) - \sum_{z \in Z} \nabla \ell(z, \theta^*)$$

- $\mathcal{U}(Z, \tilde{Z}) = -\tau \Delta(Z, \tilde{Z})$ (First-Order)
- $\mathcal{U}(Z, \tilde{Z}) = -H_{\theta^*}^{-1} \Delta(Z, \tilde{Z})$ (Second-Order)

Certified Unlearning

- How can we guarantee that information has been removed?

Certified Unlearning

- How can we guarantee that information has been removed?
- Guarantee that unlearning is indistinguishable from retraining
  - Add random noise to parameters
  - Bound the difference between retraining and unlearning

$$e^{-\epsilon} \leq \frac{P\left(\text{"Model after unlearning"}\right)}{P\left(\text{"Retrained model"}\right)} \leq e^{\epsilon}$$

Certified Unlearning

- How can we guarantee that information has been removed?
- Guarantee that unlearning is indistinguishable from retraining
  - Add random noise to parameters
  - Bound the difference between retraining and unlearning

$$e^{-\epsilon} \leq \frac{P\left(\text{"Model after unlearning"}\right)}{P\left(\text{"Retrained model"}\right)} \leq e^{\epsilon}$$

- Inspired by the concept of differential privacy (DP)

Certified Unlearning

- How can we guarantee that information has been removed?
- Guarantee that unlearning is indistinguishable from retraining
  - Add random noise to parameters
  - Bound the difference between retraining and unlearning

$$e^{-\epsilon} \leq \frac{P\Big(\text{"Model after unlearning"}\Big)}{P\Big(\text{"Retrained model"}\Big)} \leq e^{\epsilon}$$

- Inspired by the concept of differential privacy (DP)
- Theorem
  - Both update strategies are certified for convex loss functions with bounded derivatives.

Evaluating Unlearning

- We propose three criteria for evaluation

Evaluating Unlearning

- We propose three criteria for evaluation
- Efficacy
    - We require measure that information has been removed

Evaluating Unlearning

- We propose three criteria for evaluation
- Efficacy
  - We require measure that information has been removed
- Fidelity
  - Classification performance should be close to the original model

Evaluating Unlearning

- We propose three criteria for evaluation
- Efficacy
  - We require measure that information has been removed
- Fidelity
  - Classification performance should be close to the original model
- Efficiency
  - The Unlearning algorithm must be faster than retraining

Evaluating Unlearning

- We propose three criteria for evaluation
- Efficacy
  - We require measure that information has been removed
- Fidelity
  - Classification performance should be close to the original model
- Efficiency
  - The Unlearning algorithm must be faster than retraining
- All criteria must hold at the same time! We don't need
  - Fast algorithms with low fidelity or efficacy
  - Algorithms with high fidelity or efficacy that are slow

Case Study: Generative Language Models

- **Learning Model**
  - Character based language model based on LSTM
  - Trained on the novel¸ "Alice in wonderland"
  - Insertion of a canary sentence to induce memorization[1]
  - "’My telephone number is 0123456789’, said Alice."

[1] The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks, Usenix Security, 2019

Case Study: Generative Language Models

- **Learning Model**
  - Character based language model based on LSTM
  - Trained on the novel¸ "Alice in wonderland"
  - Insertion of a canary sentence to induce memorization[1]
  - "'My telephone number is 0123456789', said Alice."
- **Task**
  - Unlearn the memorized number by changing features and labels
  - "'My telephone number is not here ', said Alice." ´

[1] The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks, Usenix Security, 2019

Case Study: Generative Language Models

- **Learning Model**
  - Character based language model based on LSTM
  - Trained on the novel¸ "Alice in wonderland"
  - Insertion of a canary sentence to induce memorization[1]
  - "'My telephone number is 0123456789', said Alice."
- **Task**
  - Unlearn the memorized number by changing features and labels
  - "'My telephone number is not here ', said Alice." ´
- **Evaluation**
  - Exposure metric for efficacy of unlearning
  - Accuracy on training data for fidelity

[1] The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks, Usenix Security, 2019

Unlearning unintended memorization - Efficacy

- Start sentence "'My telephone number is "
  - Induces probability distribution over 36^{10} possible completions

Unlearning unintended memorization - Efficacy

- Start sentence "'My telephone number is "
  - Induces probability distribution over 36^{10} possible completions
  - To measure how surprised the model is we use log-perplexity

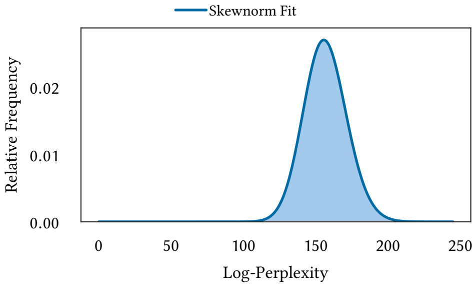
$$Px(x_1 \ldots x_{10}) = -\log(\mathbf{Pr}(x_1 \ldots x_{10}))$$

Unlearning unintended memorization - Efficacy

- Start sentence "'My telephone number is "
  - Induces probability distribution over 36^{10} possible completions ¸
  - To measure how surprised the model is we use log-perplexity

$$Px(x_1 \dots x_{10}) = -\log(\mathbf{Pr}(x_1 \dots x_{10}))$$

  - Sample 10^7 random completions to approximate distribution

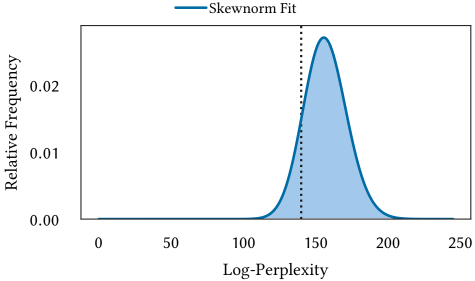


Unlearning unintended memorization - Efficacy

- Start sentence "'My telephone number is "
  - Induces probability distribution over 36^{10} possible completions
  - To measure how surprised the model is we use log-perplexity

$$Px(x_1 \dots x_{10}) = -\log(\mathbf{Pr}(x_1 \dots x_{10}))$$

  - Sample 10^{10} random completions to approximate distribution
- "'My telephone number is 8584881081"

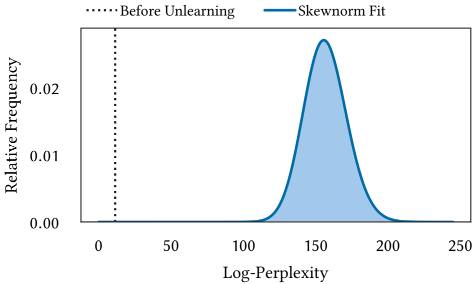


Unlearning unintended memorization - Efficacy

- Start sentence "'My telephone number is "
  - Induces probability distribution over 36^{10} possible completions ¸
  - To measure how surprised the model is we use log-perplexity

$$Px(x_1 \dots x_{10}) = -\log(\mathbf{Pr}(x_1 \dots x_{10}))$$

  - Sample 10^{10} random completions to approximate distribution
- "'My telephone number is 0123456789"

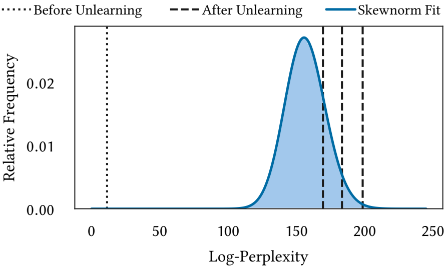


Unlearning unintended memorization - Efficacy

- Start sentence "'My telephone number is "
  - Induces probability distribution over 36^{10} possible completions ¸
  - To measure how surprised the model is we use log-perplexity

$$Px(x_1 \dots x_{10}) = -\log(\mathbf{Pr}(x_1 \dots x_{10}))$$

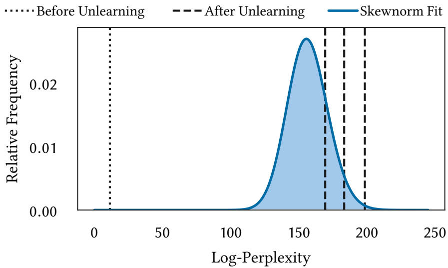
  - Sample 10^{10} random completions to approximate distribution
- "'My telephone number is 0123456789"



Unlearning unintended memorization - Efficacy

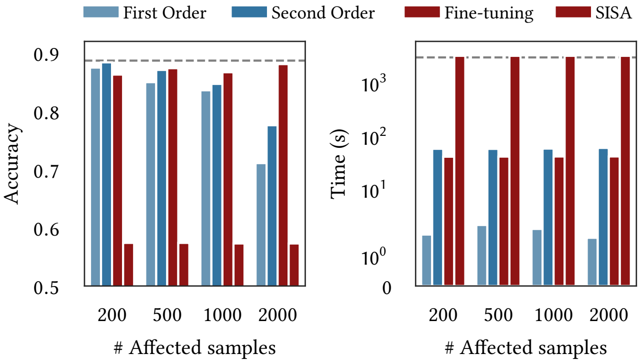
Result

Removing unintended memorization is surprisingly simple and renders extraction of memorized information infeasible.



Unlearning unintended memorization - Fidelity & Efficiency

- Performance is close to retraining for small number of canaries
- Substantial speedup compared to retraining (up to 100×)



Unlearning unintended memorization

- How is the canary completed after unlearning?
  - Prediction of replacement?
  - Gibberish caused by unlearning?

Unlearning unintended memorization

- How is the canary completed after unlearning?
  - Completions preserve structure of the dataset and punctuation

Length	Replacement	My telephone number is ...
5	taken	'... mad!' 'prizes! said the lory confused . . .
10	not there␣	'... it,' said alice. 'that's the beginning . . .
15	under the mouse	'... the book!' she thought to herself 'the . . .
20	the capital of paris	'... it all about a gryphon all the three of . . .

Case Study: Poisoning Attacks

- **Model**
  - Convolutional network (VGG) for image classification (CIFAR-10)
  - Flipping of image labels to reduce performance

Case Study: Poisoning Attacks

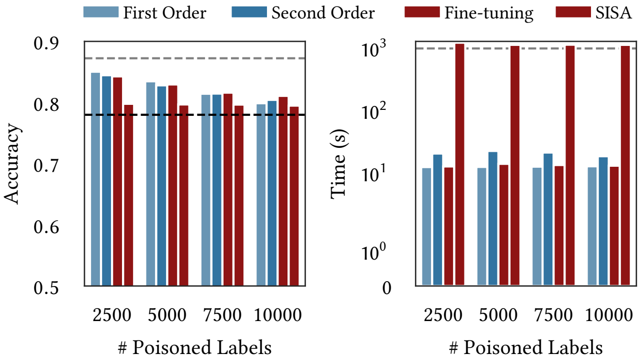
- **Model**
  - Convolutional network (VGG) for image classification (CIFAR-10)
  - Flipping of image labels to reduce performance
- **Task**
  - Unlearn the poisoned samples by correcting the labels

Case Study: Poisoning Attacks

- **Model**
  - Convolutional network (VGG) for image classification (CIFAR-10)
  - Flipping of image labels to reduce performance
- **Task**
  - Unlearn the poisoned samples by correcting the labels
- **Evaluation**
  - Accuracy on test data after unlearning for Efficacy & Fidelity

Unlearning Poisoning

- No approach can remove poisoning effect completely
- Great speedup compared to retraining



Limitations

- **Size of changes matters**
  - Our approach can fix defects caused by few erroneous samples
  - Retraining is inevitable at some point
- **Certification only for convex loss functions**
  - Modern neural networks have usually non-convex loss
  - Could be mitigated by application to final layers only
- **Unlearning requires detection**
  - Finding data to be removed is a hard problem in the real world

Conclusion

- We propose two unlearning updates $\theta = \theta^* + \mathcal{U}(Z, \tilde{Z})$
  - First order update uses gradient information
  - Second order update includes Hessian matrix
- We derive conditions to enable *certified unlearning*
- We show that our approach can solve security problems