ANITA
Anonymous big data A project funded by FFG

Privacy-Preserving Techniques for Deep Learning

Deliverable D3.2

Author(s): Peter Eigenschink

Reviewer(s): Olha Drozd, Klaudius Kalcher

Document version: 0.4
Date: 31.07.2020

Federal Ministry Republic of Austria
Climate Action, Environment, Energy, Mobility, Innovation and Technology

Programme **"ICT of the Future"** - the initiative of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK)

Disclaimer

Table of contents

1 Summary

In the era of big data Large amounts of high dimensional data make it difficult for humans to extract relevant features that could be the basis for common mathematical models, such as Bayesian models or differential equations. Deep learning models are able to identify and extract relevant features from the data on their own, offering new opportunities in domains such as autonomous driving or speech recognition. Nevertheless, there are risks coming with deep learning that creators of such models should be aware of. One particular risk, when dealing with sensitive data, is the threat of leaking sensitive information during inference. Approaches to mitigate this privacy risk come with trade-offs on different aspects, such as accuracy or efficiency. One approach to, first, quantify and, secondly, mitigate the risk of leaking sensitive information is the implementation of differential privacy into deep learning models.

The techniques that implement differential privacy into deep learning models can be divided into three categories, cf. [1]: (i) perturbation of the learned model parameters, (ii) perturbation of the training dataset; and (iii) mimic learning, i.e. the indirect exposure of the model to the sensitive data. In all of the afore-mentioned categories privacy-preserving techniques implement differential privacy by introducing random noise into the model.

Techniques that perturb the learned model parameters offset the weights, that a deep learning model learns during the training phase, by random noise. Techniques in this category achieve accurate and private results. They are also the most researched ones and are applicable to a wide range of deep learning tasks and architectures.

Techniques that perturb the training dataset are agnostic of the deep learning model itself. The noise is added to the original sensitive data and the models are only trained on this noisy data. While techniques in this category are widely applicable and the reported results are accurate, research about techniques in this category is sparse.

Mimic learning techniques work in two steps. First, an ensemble of models is trained on the sensitive data. Then, these models label a set of insensitive data and inject noise into the labels. The final model is trained on this noisy set. The indirect exposure of the final model to the sensitive data and the noisy labels in the insensitive dataset enforce privacy. While mimic learning techniques achieve accurate and private results, they are restricted to classification tasks.

DP is a successful privacy concept to quantify and mitigate the risk of leaking sensitive information in deep learning models. This concept is implemented into deep learning models by the introduction of random noise. The resulting privacy-preserving models then have a lower privacy risk, while losing some of their accuracy and efficiency.

2 Introduction

Deep neural networks have recently gained a lot of attention in- and outside of the research community. As one of the most successful approaches of machine learning, deep learning is applied to various tasks where the vast amount of data or the high dimensionality makes it difficult to create a mathematical model based on hand-designed features. The translation of text or the transcription of speech are examples of tasks where it is hard to extract the relevant features and to build mathematical models. When given a large amount of data, neural networks are able to identify and extract relevant features and leverage them for classification or generation of previously unseen data. As such, neural networks are models that can be applied to a variety of tasks. They are successful in, among other domains, computer vision, speech recognition, natural language processing and healthcare.

The building blocks of a neural network are neurons. Neurons are affine functions[1] with multiple numbers as inputs and a single number as the output. Each input is contributing to the output with a certain weight. In a neural network, these neurons are arranged in layers. The first layer is the interface to the network where the neurons obtain the data as input. The result of each neuron in this layer is passed over as input to the neurons in the subsequent layer, being further transformed and passed over to the next layer. This way only neurons in consecutive layers of the network are connected. The output of the final layer of neurons is the overall output of the network. Despite this simple architecture, neural networks are able to approximate a huge set of non-affine functions. Figure *1* depicts the architecture of the neural network.

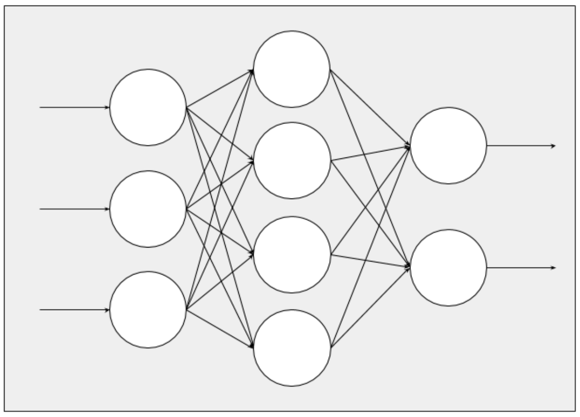

Figure 1: The basic structure of a neural network. The input layer followed by a so-called hidden layer and the output layer.

The structure of the network, i.e., the number of neurons, the number of layers, and the weight the neurons assign to each of their inputs

[1] An affine function is a function of the form $f(x_1, x_2, x_3) = w_1*x_1 + w_2*x_2 + w_3*x_3 + b$. Each w_i can be considered as the weight of the input x_i for the output, quantifying the impact that input has on the final output.

determine the output of the network when exposed to input data. Neural networks model a task by approximating a certain function. For example, when a dataset of images of cats and dogs should be classified as cats or dogs, the function should output the label "cat" for all cat images and the label "dog" for all dog images. While the creator of the neural network model fixes the structure, the adjustment screws, to most accurately approximate the desired function, are the weight parameters of each neuron.

In a first phase the network is exposed to training data with known output. The weight parameters are adjusted, such that the transformation, applied by the neural network to the input, reconstructs the output as well as possible. The adjustments are determined by minimizing the error between the predicted and the real output. This error is called the loss. There are specific optimization algorithms to calculate these adjustments for each weight, most popular is the stochastic gradient descend (SGD) algorithm. After the learning phase, the network is set up for inference and, when exposed to data that it has never seen before, it predicts the output based on what was learned from the training data in the first phase.

The purpose of a neural network model is to efficiently infer accurate predictions on unseen data. So, to some extent they will be exposed to the outside world. Either they are deployed so the model users are only able to submit their input and receive a predicted output (i.e., black box), or the model is shared in its entirety, together with the weight parameters learned during the training phase (i.e., white box). While the training data is never directly accessible, both cases still pose threats to leaking sensitive information from the original training data. Adversaries are able to gain that information from black box models by constructing certain inputs and collating the outputs. White box models are even more prone to leaking sensitive information, because adversaries can retain information about the training data directly from the observed weight parameters of each neuron. The survey [2] conducted by Mireshghallah et al. describe adversary attacks on neural network models in detail.

To quantify and mitigate the privacy risk of adversary attacks on the network, differential privacy (DP), being one of particularly successful concepts of privacy, is implemented into the neural network models. Rather than quantifying the amount of anonymity in a dataset, as with k-anonymity[2], DP quantifies the maximum impact of a single datapoint in the input data on the output of a process. This makes differential privacy particularly useful in the case of deep learning models. When training a differential private deep learning model, each access to the training data leaks information into the model, resulting in an accumulated privacy loss. The goal is to keep this loss within certain predefined bounds - the privacy budget ε.

[2] In a k-anonymous dataset there are k "similar" data points for each single data point.

The aim of this deliverable is to review published privacy-preserving techniques for deep learning. In section 3 techniques to introduce DP into deep learning models are reviewed together with applications of them to neural network models for certain tasks. Section 4 concludes the deliverable and sums up the insights from the published research.

3 Privacy-Preserving Techniques for Deep Learning

Differential privacy is applied to deep learning models to mitigate the risk of leaking sensitive information about the training data. During the training phase each access to the data yields a certain privacy loss. This allows the creator of a model to assess the total privacy loss of a model before releasing it. And the other way around, the creator is also able to quit training the model as soon as it exceeds a certain privacy budget. However, this can result in a less accurate model.

A privacy-preserving technique is a way to minimize the privacy loss as quantified by DP for a deep learning model. Models that are trained with applying privacy-preserving techniques are called privacy-preserving (PP) models. The ones that are trained without applying any particular technique to minimize the loss are called non-private.

Keeping the privacy loss for each access to the training data low is important to create an accurate model within certain privacy bounds. This loss is highly dependent on the training procedure. To mitigate the impact of a single data point on the optimization of the network, usually, DP is implemented by injecting perturbations of random noise into the model.

Random noise can be injected into the model in different ways. Based on the way the noise is injected into the model, Boulemtafes et al. [1] categorize PP models into three categories: (i) differential private model parameters, (ii) differential private input data, and (iii) differential private mimic learning. The authors also propose three performance metrics to evaluate how well a privacy-preserving model performs. These are the effectiveness (or the prediction accuracy), the training efficiency and the privacy of a model.

The three categories introduced in [1] are categories of privacy-preserving techniques that introduce DP into deep learning models. In this section, techniques that belong to these categories, are reviewed alongside examples of their applications. The techniques are applied to deep learning models for different classification tasks and are evaluated according to the performance metrics suggested in [1].

3.1 Differential Private Model Parameters

Neural networks are able to learn from data by adjusting their internal weight parameters according to the error between the real output in the training data and the predicted output. One category of privacy-preserving techniques to introduce DP into a deep learning model is the injection of noise into the weight parameters. There are three main ways to do that: (i) the direct injection into the parameters, (ii) the perturbation of the loss function and (iii) a combination of both.

The direct injection of noise into the weight parameters can be inflicted at different stages of the training phase. The most straightforward way is to add random noise to the weight parameters either after each step of the optimization procedure, or after the training is completed (cf. [3]). More complex is the technique to inject noise with a modified optimization, as proposed by Abadi et al. in [4]. The authors suggest a modified version of the SGD optimization algorithm that reduces the impact of single datapoints and injects randomness into the optimization procedure. In the following paragraphs we discuss the techniques proposed in [3] and [4].

In [3], Sei et al. construct two PP models, called "AnonymizedLearning" and "LearningFirst", and evaluate them on the ADULT[3] dataset for salary estimation. Both models preserve privacy by the injection of noise into the model parameters at different stages of the training. In the "AnonymizedLearning" model, Laplace noise is added after each training step whereas in "LearningFirst" model, the noise is added to the parameters after the training is completed. While both models achieve high accuracy in the salary estimation task, "AnonymizedLearning" outperforms "LearningFirst" on a small privacy budget (i.e., ε=1) and vice versa on larger privacy budgets (i.e., ε=10 or ε=100). Sei et al. [3] report that for most values of the privacy budget models constructed with both techniques perform better than a baseline model.

Abadi et al. [4] approach the private classification tasks of MNIST[4] and CIFAR-10[5] datasets by incorporating DP into the optimization algorithm. At each step the influence of the training data is controlled by randomly sampling a set from the training data, called a lot. The gradient is computed on each element of the lot, clipped and averaged on this lot, then noise is added to the gradient. The authors also introduce a technique to tightly estimate the privacy loss during training. Such techniques are called privacy accountants and commonly only give loose bounds on the privacy loss. The so-called moments accountant introduced in [4] gives better bounds on the privacy loss.

Compared to non-private models, the authors achieved an accuracy of 1.3% less for MNIST and 7% less for CIFAR-10, both with a privacy budget of ε=8. In case of MNIST, a reduction of the privacy budgets to ε=4 or ε=0.25 results in a drop of the accuracy by 3.3% and 8.3% respectively, when compared to the baseline. The prediction accuracy of CIFAR-10 drops by 10% for a privacy budget of ε=4 and by 13% for ε=2, both again compared to the baseline accuracy.

As reported by Abadi et al., the accuracy seems to be more dependent on the parameters, such as noise or batch size, rather than on the structure of

[3] The ADULT dataset contains 1994 census data together with the information if a person's income is above or below $50k/year. Given the census data of a person, the task is to predict whether or not the income is above $50k/year.
[4] The MNIST dataset contains images of handwritten digits from 0 to 9. Given an image, the task is to predict the number in that image.
[5] The CIFAR-10 dataset contains 32x32 pixel images of 10 different classes (e.g., airplanes, cats, dogs, cars, ships, etc.) Given an image, the task is to predict the class that image belongs to.

the network. Also, the injection of noise into the gradients, in every step of the training procedure, results in a dependence of the overall privacy cost on the number of training epochs. On the one hand, high accuracy can only be ensured by a large number of epochs, on the other hand, a large number of epochs leads to a bigger overall privacy loss.

The direct injection of noise into the weight parameters is a straightforward approach. That is also the reason that, especially, the differential private version of SGD is widely applied to construct privacy-preserving models. Another technique to construct privacy-preserving models is the injection of noise into the loss function. The loss function calculates the error between the real output in the training dataset and the output predicted by the neural network. First, this function is approximated with polynomials, such as Taylor or Chebyshev polynomials, and then the coefficients of the polynomials are perturbed. This way random noise is injected into the loss function. In the following paragraphs we discuss applications of this technique to different models.

Phan et al. [5] apply the perturbation of the loss function to an auto-encoder by approximating the reconstruction function with Taylor polynomials and injecting Laplace noise into the coefficients. The results were accurate and robust against changes in the privacy budget (between $\varepsilon=0.1$ and $\varepsilon=6.4$). However, in [6] Rahman et al. criticised this approach for the lack of meaningful privacy guarantees and that the approach cannot be easily transferred to other models.

A similar approach is conducted in [7] to privately classify data in the MNIST dataset with a convolutional deep belief network. Laplace noise is injected into the coefficients of a polynomial approximation (i.e., Chebyshev expansion) of a nonlinear loss function. This approach shows better accuracy than [4] and [5] on datasets of different sizes and for various values of the privacy budget (between $\varepsilon=0.2$ and $\varepsilon=8$). It is also robust against changes in the privacy budget and, unlike in [4], the overall privacy loss is independent of the number of training epochs, which makes this approach suitable for larger datasets.

As previously discussed, models trained with direct noise injection into weight parameters or trained with a perturbed loss function yield acceptable results for various privacy budgets. Still, there are situations where higher accuracy or better mitigation of the privacy risk is necessary. To achieve this, more complex approaches, that combine the two techniques with each other, are applied to the models.

Phan et al. [8] combine both the perturbation of weight parameters and of the objective function in their Adaptive Laplace Mechanism to privately classify images in MNIST and CIFAR-10. Not only do the authors combine both techniques, but also, they directly inject noise into the weight parameters in a more differentiated way, which they call adaptive redistribution of noise. The magnitude of the noise injected into the parameters of each neuron is determined by their relevance for the final output. This relevance is calculated by a mechanism called Layer-wise Relevance Propagation (LRP). More noise is injected into the weight parameters of less relevant neurons, leading to higher prediction accuracy. Additionally, the loss function is Taylor-approximated and noise is injected into the polynomial coefficients. For different privacy budgets, this model predicted results more accurately than the models constructed in [4]. Small to large privacy budgets were tested, with ε between 0.2 and 8 for in case of MNIST and between 2.5 and 8 for CIFAR-10. Also, the consumption of the privacy budget is independent from the number of training epochs, being an advantage over [4] for large datasets. This approach is applicable to other networks and activation functions, as well. Still, in [9] Shen et al. reported potential flaws. While noise is injected during the training phase, for humans the noisy data remained recognisable. The noise injection mechanism also has minor downsides in terms of efficiency.

In [10] Adesuyi et al. extend the technique of [8] and apply it to the private classification in the Wisconsin Diagnosis Breast Cancer (WDBC) dataset. In addition to the relevance of the neurons, the amount of noise injected into the weight parameters varies with the given privacy budget. A small budget yields large noise and vice versa. Also, noise is injected into the loss function similarly to [8]. The authors report a loss of accuracy of 4.5% for a small privacy budget of ε=1.1 and an accuracy-loss of 0.5% for a large privacy budget of ε=11, both, when compared to a non-private version.

The injection of noise into model parameters, either directly or indirectly, via the loss function, is a widely applicable approach. The above reviewed PP models only minimally lose in effectiveness compared to the non-private versions. The models keep track of the consumed privacy budget during training by means of a privacy accountant. It can be beneficial to use the moments accountant, as proposed in [4], for privacy accounting to get tighter bounds on the overall privacy loss. The independence of the privacy budget consumption from the number of training epochs, as is the case in [7], [8], is also an important feature of efficient PP models. Adaptive redistribution ([8], [10]) is a very promising approach to close the gap in prediction accuracy between private and non-private models even more. Still, the complexity of the noise injection mechanism and the points in the training phase where noise is injected can impact the training efficiency.

3.2 Differential Private Input Data

The goal of privacy in deep learning is to minimize the risk of leaking information about the training data. One obvious, but naive, privacy-preserving technique is the injection of noise directly into the original training dataset to create a private version of that dataset. The neural network is then trained on that dataset, resulting in a PP model. However, to obtain sensible privacy guarantees this way, the amount of noise that has to be injected is large, making the data rather useless.

The noise has to be injected carefully in order to retain the utility of the training data after the perturbation. To achieve this, Soria-Comas et al.

propose, in [11], a combination of the perturbation of the input data with microaggregation-based k-anonymity. Microaggregation modifies the original data by, first, creating clusters of k records, that are as similar as possible, and then, replacing all of them by a record, typically, the centroid record, that represents the cluster. DP is then enforced by injecting noise into the centroid records and training a neural network on this private input data. This technique increases the utility of the input data, when compared to the above-mentioned naive approach of perturbing each single record, and results in a more accurate model.

In [3] Sei et al. apply this technique to a model called "AnonymizingFirst". The model achieved high accuracy when evaluated on the task of salary prediction on the ADULT dataset. The authors compare this model to two other models that preserve privacy via the perturbation of the weight parameters, namely "AnonymizedLearning" and "LearningFirst" that were described in section 3.1. For small privacy budgets (i.e., ε=1) "AnonymizingFirst" outperformed these approaches. However, for large budgets (i.e., ε=100) it performed worse than "AnonymizedLearning" and "LearningFirst" models.

The naive injection of noise into the training dataset usually leads to either inaccurate or non-private results. By combining this naive perturbation of the input data with microaggregation-based k-anonymity ([11]), in [3] Sei et al. report both accurate and private results for small privacy budgets. As this technique is applied to a dataset and not directly to a model, it is applicable to a wide range of deep learning tasks. However, compared to the other privacy-preserving techniques, published research on this technique is scarce.

3.3 Differential Private Mimic Learning

In mimic learning the original training data are protected by introducing an extra layer between the data and the model. A teacher model is trained on the original labelled dataset, and is then used to annotate a large unlabelled dataset. A so-called student model is the final product of this privacy-preserving technique. The student model is trained on the dataset annotated by the teacher. This way the student has only indirect access to the original labels by means of the teacher model.

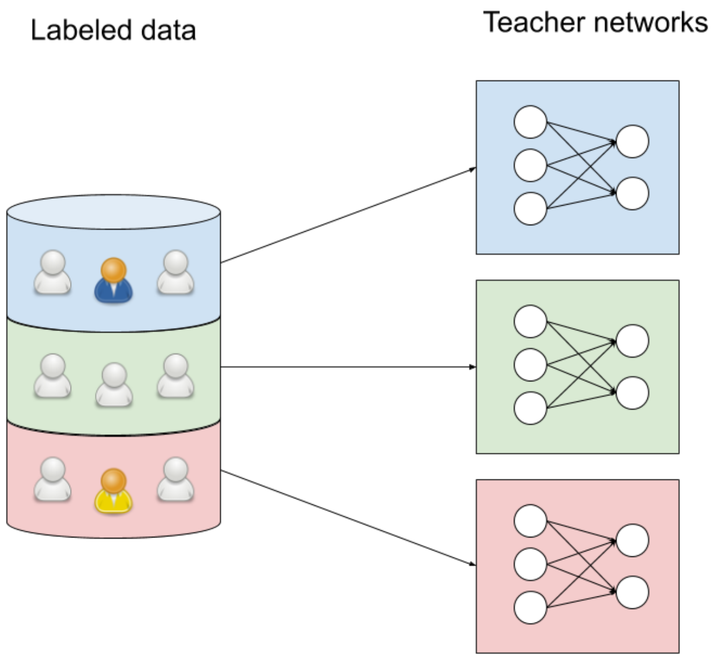

Figure 2: The first step in the training of the PATE scheme. An ensemble of teacher networks is trained on disjoint subsets of the original dataset. One particular individual is known by only one teacher network.

In [12] Papernot et al. describe "Private Aggregation of Teacher Ensembles" (PATE). An ensemble of teacher models is trained on disjoint subsets of the sensitive original data (see Figure *2*). Then, the teachers vote together on the label for each datapoint of an unlabelled non-sensitive dataset. To enforce privacy, Laplace noise is added to the count of votes for each class. Finally, each datapoint is annotated with the label of the class with the majority of votes. The student model is then trained on that non-sensitive dataset with the labels assigned by the teacher ensemble (see Figure *3*).

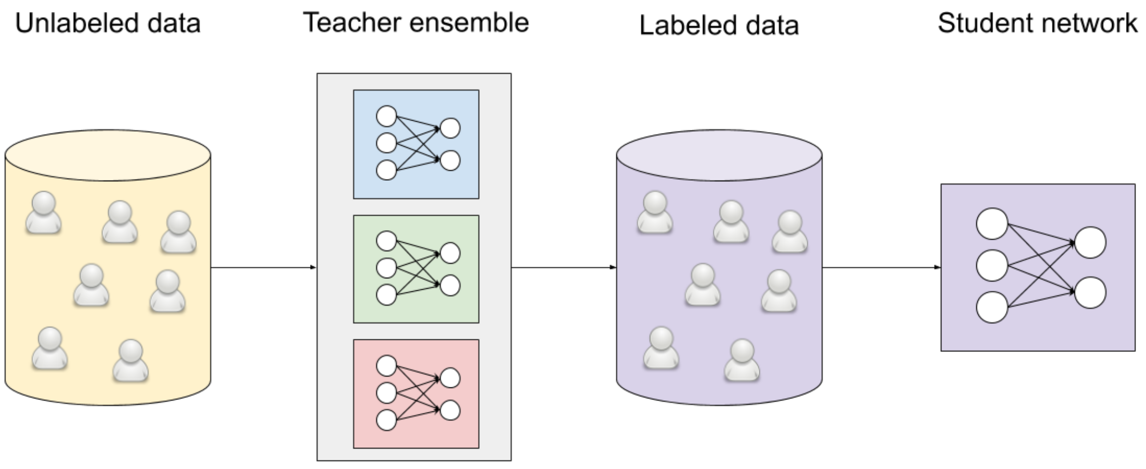

Figure 3: The second step in PATE is the training of a student model. The student model is only indirectly exposed to the original sensitive data through non-sensitive data labeled by the teacher ensemble.

The authors compared PP models trained with PATE on the task of labelling MNIST and SVHN[6] datasets. On the MNIST dataset the PP model was around 1% less accurate than the non-private baseline model for both privacy budgets with ε between 2.04 and 8.03. The reported accuracy on labelling the SVHN dataset was 2% below the baseline result for a large privacy budget (i.e., ε=8.19) and almost 11% more inaccurate for a modest privacy budget (i.e., ε=5.04). In both cases the non-private baseline is a model directly trained on the entire original dataset. The results show that the accuracy depends on the privacy budget, however, this dependency is minimal in the case of MNIST.

When compared to the perturbation of model parameters, described in section 3.1, PATE protects the individual privacy even if the model is released as a white box model. The privacy increases with the number of teachers, but, as having more teachers reduces the size of the training set for each teacher, a higher number of teachers can negatively influence the accuracy. In view of this trade-off the number of teachers has to be chosen sensibly.

Dehghani et al. [13] transfer PATE to a core task in the domain of Information Retrieval (IR), namely document ranking. This task is, for example, highly relevant for search engines, because documents have to be ranked by their relevance for a given query. Large labelled datasets for IR tasks are crucial, however, scarce, due to the documents being very sensitive. Dehghani et al. apply PATE to document ranking with acceptable accuracy and low privacy risk guarantee. Instead of perturbing the total vote count of the teachers' predictions, they inject noise into every single teacher's prediction.

Mimic learning achieves good results in terms of accuracy for different privacy budgets. Additionally, this technique yields a decent privacy protection, because the student model never observes the sensitive training data. While accuracy and privacy are reported to be acceptable, effectiveness and efficiency as well as privacy are sensitive to the chosen number of teachers. Due to its architecture, mimic learning is most suitable for classification tasks.

[6] The Street View House Numbers (SVHN) dataset contains real world images of house numbers obtained from Google Street View. The images are centered around single numbers and belong to one of 10 classes, one for each number from 0 to 9. Given an image of a house number, the task is to predict the number in the image.

4 Conclusion

This document provides an overview and evaluation of the existing privacy-preserving techniques for deep learning. Differential privacy is implemented into deep learning models either by imposing it on the model parameters, on the input data, or by only indirectly exposing the model to the original data via mimic learning.

The majority of research is carried out in the perturbation of deep learning model parameters. This technique is most appealing due to the accuracy for moderate privacy budgets, the training efficiency and the wide applicability to deep learning models for different tasks. The recent advances in adaptive redistribution of noise in [8] and [10] further improve the accuracy and the privacy of models with perturbed weight parameters.

The direct perturbation of the input data has been researched very little, still, [3] reported good results in an application of this technique.

Mimic learning is able to produce PP models that can be as accurate as models with perturbed weight parameters. This technique can also help in situations with scarcely labeled data. However, the applicability of mimic learning is restricted to classification tasks.

Overall, DP methods for Deep Learning are only in their early stages. One of the biggest challenges remains finding a good trade-off between accuracy and privacy. In the papers discussed here, ε values that provide strong guarantees (ε below 1) lead to limited accuracy of the models, and the best published results in terms of accuracy used high privacy budgets (ε values of 8 and above), which might be too weak for practical purposes. Any application of the PP methods currently available thus must evaluate whether an acceptable trade-off can be achieved.

5 References

[1] Amine Boulemtafes, Abdelouahid Derhab, and Yacine Challal, "A review of privacy-preserving techniques for deep learning," Neurocomputing, vol. 384, pp. 21 - 45, 2020.

[2] Fatemehsadat Mireshghallah et al. (2020) Privacy in Deep Learning: A Survey. [Online]. https://arxiv.org/abs/2004.12254

[3] Y. Sei, H. Okumura, and A. Ohsuga, "Privacy-Preserving Publication of Deep Neural Networks," in 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016, pp. 1418-1425.

[4] Martin Abadi et al., "Deep Learning with Differential Privacy," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016.

[5] NhatHai Phan, Yue Wang, Xintao Wu, and Dejing Dou, "Differential Privacy Preservation for Deep Auto-Encoders: an Application of Human Behavior Prediction," in AAAI Conference on Artificial Intelligence, 2016.

[6] Md Atiqur Rahman, Tanzila Rahman, Robert Laganière, Noman Mohammed, and Yang Wang, "Membership Inference Attack against Differentially Private Deep Learning Model.," Trans. Data Priv., vol. 11, no. 1, pp. 61-79, 2018.

[7] NhatHai Phan, Xintao Wu, and Dejing Dou, "Preserving differential privacy in convolutional deep belief networks," Machine learning, vol. 106, no. 9-10, pp. 1681-1704, 2017.

[8] NhatHai Phan, Xintao Wu, Han Hu, and Dejing Dou. (2017) arXiv. [Online]. http://arxiv.org/abs/1709.05750

[9] Juncheng Shen, Juzheng Liu, Yiran Chen, and Hai Li. (2018) Morphed Learning: Towards Privacy-Preserving for Deep Learning Based Applications. [Online]. http://arxiv.org/abs/1809.09968

[10] T. Adesuyi and B. Kim, "A layer-wise Perturbation based Privacy Preserving Deep Neural Networks," in 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2019, pp. 389-394.

[11] Jordi Soria-Comas, Josep Domingo-Ferrer, David Sánchez, and Sergio Martínez, "Enhancing Data Utility in Differential Privacy via Microaggregation-Based k-Anonymity," The VLDB Journal, vol. 23, no. 5, pp. 771–794, 2014.

[12] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. (2016) Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. [Online]. https://arxiv.org/abs/1610.05755

[13] Mostafa Dehghani, Hosein Azarbonyad, Jaap Kamps, and Maarten de Rijke. (2017) Share your Model instead of your Data: Privacy Preserving Mimic Learning for Ranking. [Online]. http://arxiv.org/abs/1707.07605