Differential Privacy

Jarin Tasnim Khan Kashfee
School of Data and Sciences
Brac University
Dhaka, Bangladesh
jarin.tasnim.khan@g.bracu.ac.bd

I. Introduction

Differential privacy (DP) is a way to preserve the privacy of individuals in a dataset while preserving the overall usefulness of such a dataset. Ideally, someone shouldn't be able to tell the difference between one dataset and a parallel one with a single point removed. It guarantees to provide the privacy of data when analyzing and sharing sensitive data. To do this, randomized algorithms are used to add noise to the data. The idea of adding noise to the data before analyzing it is that the output of the analysis does not reveal any individual-level information.

II. Related Works

[1] provides a comprehensive survey of differential privacy, its definition, and its applications. It describes the key concepts and techniques used to achieve differential privacy and discusses some of the open challenges in the field. The main problem of [1] would be that it addressed the need for privacy-preserving data analysis in the context of sharing sensitive data while preserving the privacy of individuals. The paper [1] provided a comprehensive survey of the concept of differential privacy, including its definition, its properties, and its applications. It discussed the key techniques used to achieve differential privacy, such as randomized response and the use of noise, and the challenges in implementing and evaluating differentially private algorithms. The result of the paper is an overview of the state-of-the-art in differential privacy research, highlighting its potential for enabling data sharing while preserving privacy. About the work on this paper, I think that the paper provided a strong foundation for understanding differential privacy and its potential for preserving privacy in data analysis.

[2] provides an in-depth analysis of the theoretical foundations of differential privacy. It discusses the mathematical definitions and theorems that underpin the concept and provides examples of differentially private algorithms. The problem they are tracking is they addressed the need for a theoretical foundation for differential privacy, including its formal definition and mathematical properties. [2] presented the mathematical definition of differential privacy, as well as the key theorems that underpin the concept, such as - the composition theorem and the post-processing theorem. It also provided examples of differentially private algorithms and discussed the trade-offs between privacy and utility. As a result of their approach, finally rigorous mathematical foundation for differential privacy and demonstrated its applicability to a wide range of data analysis tasks was provided. About the work on this paper, I think the paper presented a theoretical foundation of differential privacy, making it a valuable resource for researchers.

[3] presents a framework for the design of differentially private algorithms. It proposes a methodology for the analysis of privacy guarantees and provides an implementation of this framework in the form of a software library. The main problem identified in the paper is the need for a practical framework for design- ing differentially private algorithms that guarantee privacy and utility.

[4] addressed the privacy risks associated with gradient descent in federated learning, where multiple clients contribute their data to train a shared model while preserving their privacy. then it analyzed the privacy risks associated with gradient descent in federated learning and proposed a differentially private variant of the algorithm that adds noise to the gradients to protect the privacy of individual clients. It also presented a theoretical analysis of the privacy guarantees of the proposed algorithm. The paper demonstrated the effectiveness of the proposed differentially private gradient descent algorithm on a range of machine learning tasks, showing that it can achieve strong privacy guarantees while maintaining high levels of accuracy as results.

[5] addressed the need for a technique to amplify the privacy guarantees of differentially private algorithms. It proposed a privacy amplification technique based on the idea of mixing differentially private algorithms with noise. It also presented a theoretical analysis of the privacy guarantees of the proposed technique. As a result, the paper shows that it can significantly improve the privacy guarantees of differentially private algorithms.

[6] proposes a new privacy definition called Renyi differential privacy (RDP). RDP is a generalization of the commonly used privacy definition, epsilon-differential privacy (epsilon-DP), that allows for a more flexible trade-off between privacy and utility. [6] aims to address the problem of accurately measuring privacy in situations where the noise added to the data to ensure privacy can be varied. The authors argue that epsilon-DP provides a one-size-fits-all privacy guarantee that can be overly conservative in situations where the privacy risk is low. The authors evaluate the effectiveness of RDP by applying it to two common privacy preserving mechanisms:

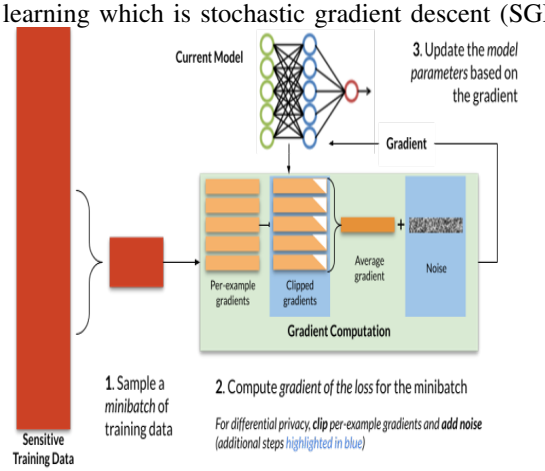
the Laplace mechanism and the Gaussian mechanism. Then they compared the performance of RDP to epsilon-DP on several bench- mark datasets and shows that RDP provides a more accurate measure of privacy while still maintaining good utility.

[7] proposes a method for training an image embedding model that ensures user-level differential privacy (ULDP). The authors' approach is to add noise to the gradients during training to achieve ULDP. They propose a new privacy definition called ULDP, which ensures that an adversary cannot learn any in- formation about a specific user from the trained model. They also introduce a new loss function that balances the trade-off between privacy and accuracy in the embedding model. The authors evaluate their approach on a large-scale image dataset and show that it provides better privacy guarantees compared to existing privacy-preserving methods. They also demonstrate that their method achieves competitive performance on several image retrieval benchmarks.

[8] proposes a provably privacy-preserving algorithm for data distillation, called DP-KIP. The authors aim to address the problem of preserving privacy while distilling knowledge from large datasets. The authors' approach is to use kernel ridge regression (KRR) with neural tangent kernels to estimate the distilled data points, also known as kernel-inducing points. The authors then apply differential privacy to KRR using the differentially private stochastic gradient descent (DP-SGD) algorithm. The authors provide a computationally efficient implementation of DP-KIP using JAX. The authors evaluate their approach on several image and tabular datasets and show that DP-KIP achieves high performance while providing differential privacy guarantees. The authors also compared their approach with other state-of-the-art methods and demonstrate the effectiveness of DP-KIP in terms of both performance and privacy.

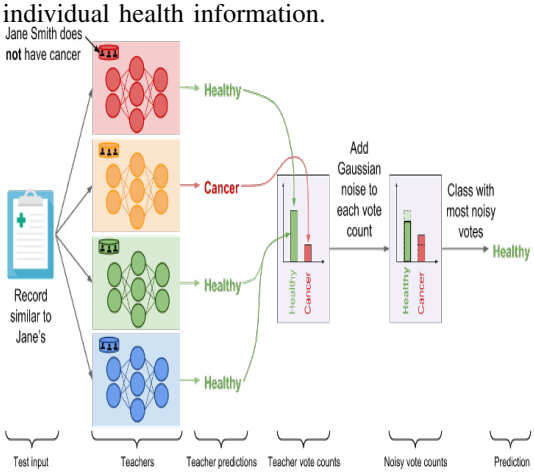
[9] provides valuable insights into the practical implementation of differential privacy in machine learning models. It offers guidance and best practices for deploying differentially private machine learning models while preserving individual privacy. It emphasizes the need for incorporating privacy protections into machine learning systems, especially when handling sensitive data. It highlights the potential privacy risks associated with traditional machine learning techniques that can inadvertently expose sensitive information about individuals. To address these challenges, the post introduces differential privacy as a principled approach to privacy preservation. It explains that differential privacy ensures that the results of computation or analysis remain practically the same, regardless of whether an individual's data is included or excluded from the datasets.

Two approaches were made for training deep neural networks with differential privacy. One approach is DP-SGD which adds noise to the gradient computed by SGD to ensure privacy. The Differential stochastic gradient descent(DP-SGD) is proposed to modify the model updates that are computed by the most common optimizer used in deep learning which is stochastic gradient descent (SGD).



Stochastic gradient descent trains the above process iteratively. In each number of iterations, a small number of training examples (named minibatch in the figure) are sampled from the training set. Then after that, the optimizer computes the average model error on these examples and then differentiates this average error from the model parameters to get a gradient vector. After that, the main two modifications have been made by DP-SGD to obtain differential privacy, and that are gradients (figure-per-example gradients on a per example basis rather than multiple examples) are clipped so that the sensitivity is controlled and next Gaussian noise is added to the summation.

The other approach is namely Model Agnostic Private Learning which trains many non-private models on subsets of sensitive data and finally uses DP to generate results. An example was provided related to medical diagnosis using machine learning. Suppose a healthcare provider wants to develop a machine learning model to predict a patient's disease risk based on their medical records. By incorporating differential privacy, the model can provide accurate predictions while preventing unauthorized access to individual health information.



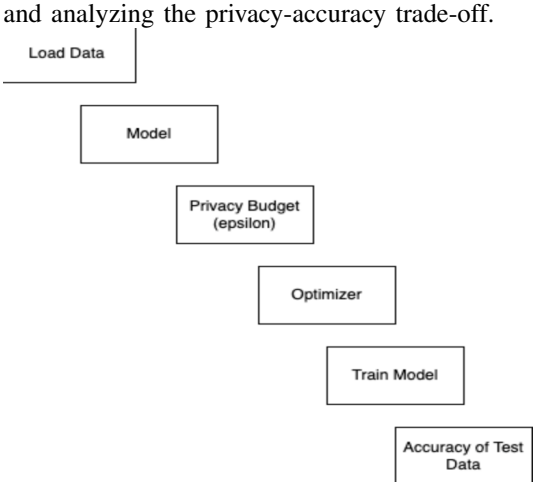
Here in the example provided is a PATE Framework. In DP generally, noise is added to the gradient. But instead of adding noise to the gradient, PATE trains non-private models which can be seen in teachers on subsets of data, and then

ask the models to on the correct prediction (Healthy / Cancer) using a DP aggregation mechanism.

III. Research Methodology

The method used is a Differentially Private stochastic gradient descent(DP-SGD) optimizer implementing a Convolutional Neural Network (CNN) on the MNIST dataset by adding noise to the gradient updates during the time of training for ensuring privacy. By adding noise to the gradients, it becomes difficult for an attacker to distinguish between the training data of a single individual and a group of individuals, thereby preserving the privacy of the individuals. The randomness is introduced in the form of noise added to the gradients computed during training. The optimizer DPKerasSGDOptimizer, which is a class by TensorFlow Privacy library and it extends the Keras implementation of the SGD optimizer to incorporate DP. The optimizer uses a Gaussian mechanism for adding noise to the gradients with the amount of noise controlled by the privacy budget parameter (epsilon). To train a convolutional neural network(CNN) to recognize handwritten digits with the DP-SGD optimizer provided by the TensorFlow library, 'tf.keras is used. Specifically, the optimizer is used to implement the Stochastic Gradient Descent (SGD) algorithm with differential privacy guarantees. The privacy budget is controlled by the value of epsilon. It is a hyperparameter that determines the amount of noise added to the gradients during training. The privacy budget controls the tradeoff between privacy and model. Higher values of epsilon provide better model accuracy, but with less privacy, whereas lower values of epsilon provide better privacy but with reduced model accuracy. accuracy. Finally, the accuracy of the trained model is evaluated on the test data, and the test accuracy is recorded for each privacy budget. The accuracy is then plotted as a function of the privacy budget to understand the relationship between model accuracy and privacy.

So the workflow goes by loading and preprocessing the data, defining a model architecture, applying differentially private optimization, training the model, evaluating the performance, and analyzing the privacy-accuracy trade-off.



The implementation part of the figure begins with importing the necessary packages including TensorFlow, NumPy, and scikit learn. After that, it loads the data which is the MNIST dataset, and then performs data pre-processing steps like normalizing pixel values and converting labels to one-hot encoding vectors. Next comes the model part, here CNN model is defined by the Tensorflow keras. The convolution neural network model consists of convolutional layers, max pooling, a flatten layer and a dense layer. Then the CNN model is trained with different privacy budgets (epsilon). The privacy budget is defined as an array consisting of epsilon values that control the amount of privacy protection provided. For each epsilon values a differentially private optimizer, DPKerasSGDOptimizer is created. The model is then compiled with a DP optimizer and trained using the fit () function. The process of training includes specifying batch size, number of epochs, and a validation split. During training, the model's performance is evaluated on the test dataset using the evaluate() function, providing the test accuracy and loss. The test accuracy is recorded for each epsilon value and stored in the accuracy list. Finally, implementation is completed by importing matplotlib library to plot the test accuracy as a function of the privacy budget (epsilon). The resulting plot helps visualize the relationship between privacy and model performance. The plotting to test accuracy as a function of epsilon is done extra to show the relationship.

Between the code and the changed code, the changes that have been made in the validation split ratio, the validation split ratio is set to 0.2 which means that 20 percent of training data(60000 samples) is used for validation, resulting in 48000 samples for training and 120000 samples for validation. In the modified code that has been implemented, the validation split ratio is set to 0.1 which means 10 percent of training data (54000 samples) is used for validation, resulting in 48600 samples for training and 5400 samples for validation. The validation split ratio is flexible and can vary depending on the specific requirements of model training. In the first implemented code a larger validation set is used which can provide more reliable validation performance but was taking more training time. In the modified code, as the validation set used is smaller, it is faster to train but the validation performance can vary.

IV. Results

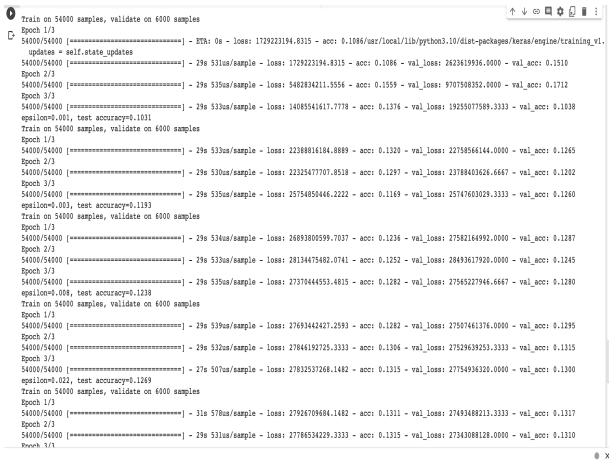
For the first implementation of the raw code, the results of the output that is trained on 60000 samples, validate on 10000 samples, and epoch = 3 is -

```
model.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])

model.fit(train_data, train_labels,
          epochs=epochs,
          validation_data=(test_data, test_labels),
          batch_size=batch_size)
```

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/3
60000/60000 [==============================] - 3571s 60ms/sample - loss: 0.9281 - acc: 0.7071 - val_loss: 0.3851 - val_acc: 0.8891
Epoch 2/3
60000/60000 [==============================] - 3511s 59ms/sample - loss: 0.3811 - acc: 0.8908 - val_loss: 0.3433 - val_acc: 0.9150
Epoch 3/3
46000/60000 [============================>.....] - ETA: 13:38 - loss: 0.3489 - acc: 0.9154
```

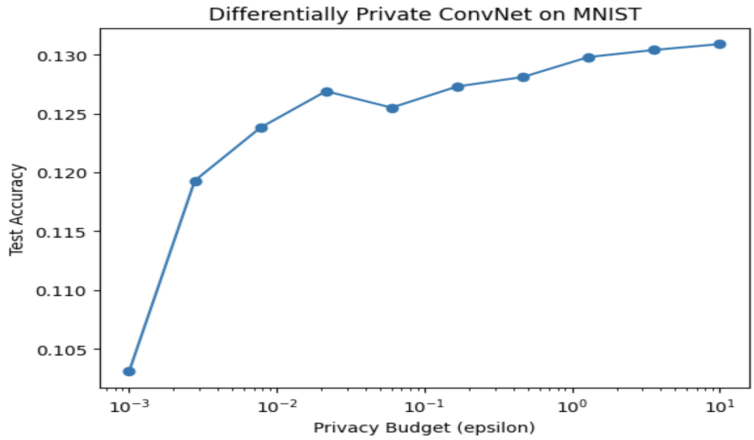
This code particularly, generates the accuracy of the model in terms of training epochs.
Next, for the modified code part, the results of the output that is trained on 54000 samples, validate on 6000 samples with epoch = 3 is -



This code particularly, generates the accuracy of the model in terms of privacy budgets (epsilon)

A. *Discussion*

For understanding the relationship between model accuracy and privacy, the accuracy is then plotted as a function of the privacy budget epsilon. The relationship between model accuracy and privacy can be seen below in the figure.



V. Conclusion

Differential privacy is a powerful concept that provides a principled approach to balancing privacy and utility in machine learning and data analysis. By incorporating differential privacy techniques into the training process, we can ensure that the models we build respect the privacy of the individuals whose data is used. Applying differential privacy techniques, organizations can responsibly handle and analyze sensitive data while minimizing the risk of privacy leakage. By adopting and understanding these techniques, we can ensure that privacy concerns are adequately addressed

Acknowledgment

References

1. ”Differential privacy: A survey of results” by Cynthia Dwork (2008) 2. ”The Algorithmic Foundations of Differential Privacy” by Cynthia Dwork and Aaron Roth (2014) 3. ”A firm Foundation for private data Analysis” by Cyn- thia Dwork et al. (2015) 4. Differential Privacy: What is all the noise about? (https://arxiv.org/pdf/2205.09453v1.pdf) 5. Opacus: User-Friendly Differential Privacy Library in PyTorch (https://arxiv.org/pdf/2109.12298v4.pdf) 6. Renyi differential privacy (https://arxiv.org/pdf/1702.07476v3.pdf) 7. Learning to Generate Image Embeddings with User-level Differential Privacy” by Xu et al. from Google Research (https://arxiv.org/pdf/2211.10844v1.pdf) 8. Differentially Private Kernel Inducing Points (DP-KIP) for Privacy-preserving Data Distillation (https://arxiv.org/pdf/2301.13389v1.pdf) 9. Deploy Machine Learning. (https://www.nist.gov/blogs/cybersecurity-insights/how-deploy-machine-learning-differential-privacy)