Application of Differential Privacy Stochastic Gradient Descent For LLM Privacy Protection

Pals Chinnakannan

Git: https://github.com/pals-ucb/privacy-sdp

1 Introduction

The rapid growth in Generative Artificial Intelligence around Large Language Models (LLMs), and the advancement in the associated techniques for fine-tuning and prompt tuning have spurred many organizations to customize LLMs for solving business critical problems and for competitive advantage. This growth has resulted in deploying fine-tuned or prompt-tuned LLMs using a wide-range of data containing sensitive Privacy information. However, surveys and studies on the security of LLMs and associated privacy indicate potential scope for vulnerabilities and wide-spread privacy violation on the horizon. These concerns are clearly brought out by Yao et al in [1]. Differential privacy is a technique used to protect PII and other sensitive information through fuzzing such information in a dataset. The essence of differential privacy lies in the injection of "noise" to obscure privacy information present in datasets used in domains like LLM Applications, thereby preserving anonymity while maintaining the overall integrity and the utility of the dataset. Several different Differential Privacy Techniques have been proposed, researched and developed [2], [3]. "Differential Privacy applied to Stochastic Gradient Descent" is a very fundamental technique for privacy protection and is discussed in several papers including the "Selective Differential Privacy for Large Language Models", and "Just Fine-Tune Twice: Selective Differential Privacy for Large Language Models". SDP is very appropriate for the sparse amount of privacy information present in very large datasets like those used in LLM Training and Fine-Tuning.

2 Objectives

The objectives of this project are as follows:

- Study the Differential Privacy Stochastic Gradient Descent, a fundamental technology for LLM privacy protection.
- Get a deeper understanding of Stochastic Gradient Descent (SGD), Differential Privacy SGD, the current state of the Privacy Protection in the two major machine learning frameworks, Tensorflow and PyTorch.
- Procure and study the data sets described in the paper, specifically, 1) GLUE (Wang et al., 2018) a widely- used multi-task benchmark dataset for NLU, which contains sensitive information such as name and date. 2) Wikitext-2 (Merity et al., 2017) that contains Wikipedia articles with private information such as name and date. *3)* ABCD (Chen et al., 2021), a human-human customer service dialogue dataset under real-world scenarios with user private information such as name and order IDs.
- Build the foundation required for applying DP-SGD technology on LLMs using different datasets that enables easy training, testing of Differential Privacy using either TensorFlow or PyTorch.
- Uncover the current limitations in privacy protection, especially in DP-SGD, and propose next steps on improving available techniques.

3 Literature Survey

The fundamental Differential Privacy concept, the associated mathematics and the technology was pioneered by Dwork et al in their work on Algorithmic Foundations for Differential privacy [1]. The earlier work from M.Abadi and et al [2], applies the differential privacy concepts and technology to LLMs. These two-research works serve as a good starting point for this project. In addition, most of the authors of this proposal researched Differential Privacy techniques in an earlier course work, which led to their seminal work on "Applied Differential Privacy on Large Language Models". In that study and paper, the authors have clearly brought out the privacy concerns of LLMs and have shown the revealing nature of LLM, specifically, LLama-2. This work is continuing in the same vein as Privacy of LLMs. Finally, the papers on SDM [4] and [5] serve as the foundation for this project.

Martin Abadi, et al described the Differential Privacy Stochastic Gradient Descent in their paper "Deep Learning with Differential Privacy". In this paper they describe the DP-SGD Algorithm as an enhancement to the basic SGD in which they select a random subset batch of the input data with a certain probability, compute the Gradient for each sample in the subset and clip the L2 Norm for this data to a configured limit, then compute the average for gradient for this batch, apply a random noise to this gradient which is used for computing the model parameters. In addition, they also proposed a compute the privacy loss of the mechanism based on the information maintained by a privacy accountant.

4 ML Algorithms

4.1 Introduction

Gradient Descent (GD), Stochastic Gradient Descent (SGD), and Differentially Private Stochastic Gradient Descent (DP-SGD) are fundamental optimization algorithms used in machine learning for training models. Each has its unique characteristics, advantages, and trade-offs concerning complexity, memory, CPU usage, supported model parameters, and scalability.

4.2 Gradient Descent (GD)

Gradient Descent is an optimization algorithm used to minimize the loss function of a machine learning model by iteratively moving in the direction of the steepest descent as defined by the negative gradient. The algorithm updates the model parameters by taking steps proportional to the negative gradient of the loss function with respect to the parameters.

Pros:

- **Convergence**: Provides stable and smooth convergence towards the minimum.
- **Simplicity**: Easy to implement and understand.

Cons:

- **Computationally Expensive**: Requires computing the gradient for the entire dataset in each iteration, leading to high computational cost.
- **Memory Usage**: Needs to store the entire dataset in memory.
- **Scalability**: Not well-suited for large datasets due to high computational and memory requirements.
- **Complexity**: O(n * m), where n is the number of data points and m is the number of parameters.

4.3 Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent is a variation of Gradient Descent where the model parameters are updated incrementally for each training example, rather than the entire dataset. This introduces randomness, which can help to escape local minima and speed up convergence.

Pros:

- **Efficiency**: Faster iterations compared to GD as it processes one or a few training examples at a time.
- **Scalability**: Better suited for large datasets due to lower memory and computational requirements per iteration.
- **Convergence**: Can escape local minima due to the randomness introduced in updates.

Cons:

- **Convergence Stability**: Less stable convergence compared to GD, can oscillate around the minimum.
- **Noise**: Introduces noise in the gradient estimation.
- **Complexity**: O(m), where m is the number of parameters.
- **Memory**: Low, only needs to load a few training examples at a time.
- **CPU Usage**: Lower per iteration but may need more iterations to converge.

4.4 Differentially Private Stochastic Gradient Descent (DP-SGD)

Differentially Private Stochastic Gradient Descent is an extension of SGD that incorporates differential privacy to protect the privacy of individual training examples. It achieves this by adding noise to the gradients and clipping their norm, ensuring that the model does not memorize sensitive information from the training data.

Pros:

- **Privacy**: Provides strong privacy guarantees, preventing the model from leaking sensitive information about the training data.
- **Scalability**: Similar scalability benefits as SGD with added privacy.

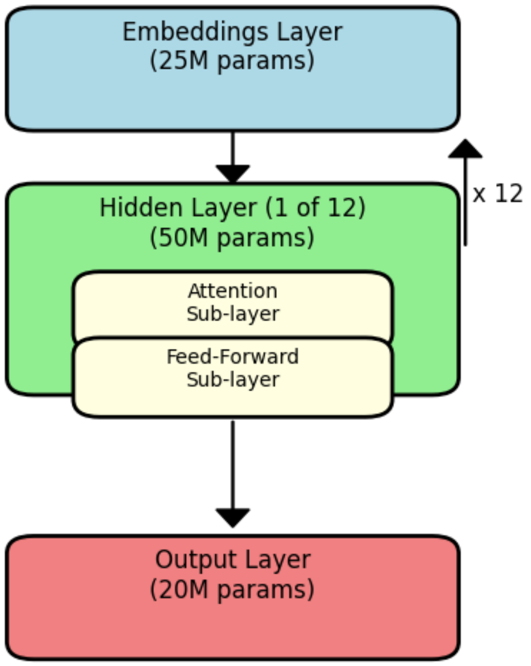
Cons:

- **Complexity**: Adds complexity due to the need for gradient clipping and noise addition.
- **Convergence**: Noise addition can slow down convergence and affect model accuracy.
- **Parameter Tuning**: Requires careful tuning of privacy parameters (e.g., noise scale, clipping norm).
- **Complexity**: O(m), with additional overhead for privacy-preserving operations.
- **Memory**: Low to moderate, depending on the implementation of privacy mechanisms.
- **CPU Usage**: Higher than standard SGD due to additional operations for differential privacy.

5 GPT2 LLM

The Project uses GPT2 LLM which was fine-tuned using wiki2 dataset. This dataset was used in earlier experiments identified in research paper [4] amd [5]. In addition, the wiki2 dataset contains several privacy related samples and the intent was to use those privacy related samples for testing and evaluating the privacy performance.

5.1 Simplified Graphical Block View of GPT-2 Layers



5.1.1 Layer Description

- **Input Embedding**: Converts input tokens into dense vectors that can be processed by the model. Each token is represented as a high-dimensional vector.
- **Positional Encoding**: Adds information about the position of each token in the sequence. Since the model processes the entire sequence at once, positional encoding helps it understand the order of tokens.
- **Transformer Block**: The core component of GPT-2, consisting of:
- **Multi-Head Self-Attention**: Allows the model to focus on different parts of the input sequence simultaneously, capturing relationships between tokens.
- **Feed-Forward Network**: Processes the output of the self-attention mechanism, adding non-linearity and complexity to the model.
- **Repeated Transformer Blocks**: The transformer block is repeated multiple times (e.g., 12, 24, or 36 times) to increase the model's capacity and depth. Each block processes the output of the previous block.
- **Output Layer**: Generates the final predictions based on the processed input. This layer maps the high-dimensional vectors back to the vocabulary space, producing probabilities for each token in the vocabulary.

5.1.2 Model Parameters (Weights)

Each layer in the GPT-2 model has associated weights that are learned during training. These weights determine how the input data is transformed at each layer. The weights are adjusted through backpropagation to minimize the loss function, improving the model's performance over time.
The following shows the total Gpt2 Model Parameters and the Frozen parameters during DP-SGD training. Many parameters were frozen to train the model using DP-SGD Opacus library. The per sample gradient computation, clipping the gradient using the given l2 norm, computing the average gradient for backpropagation required the Opacus library to allocate memory proportional to the number of trainable parameters. Many trainable parameters require more GPU memory and after a certain limit the training failed. This allowed only 2 hidden layers to be trained.

Number of parameters	*: 124439808*
Number of Trainable Parameters	*: 14175744*
Number of frozen parameters	*: 110264064*

6 Datasets

The project uses Wiki2 dataset.

6.1 Overview of the WikiText-2 Dataset

The WikiText-2 dataset, often referred to as Wiki2, is a widely used benchmark in the field of natural language processing (NLP). It was created with the intention of providing a high-quality, large-scale corpus for training and evaluating language models. The dataset consists of over 100 million tokens extracted from Wikipedia articles, which are known for their high standard of writing and comprehensive coverage of various topics.

- **Size**: The WikiText-2 dataset contains approximately 2 million tokens.
- **Quality**: Extracted from Wikipedia, the dataset comprises well-written, informative content.
- **Diversity**: Covers a broad range of topics, providing a rich source of diverse language patterns.
- **Structure**: Unlike other datasets, WikiText-2 retains the original formatting, such as article structure, punctuation, and special characters, which helps in modeling real-world text more effectively.

6.2 Rationale for Creating the WikiText-2 Dataset

The creation of WikiText-2 was motivated by the need for a high-quality, large-scale dataset that could address some of the limitations observed in existing corpora used for language modeling. These limitations included issues related to text quality, coherence, and diversity of topics. The white paper titled "Pointer Sentinel Mixture Models" by Merity et al, provides an in-depth explanation of the motivation behind creating the WikiText-2 dataset. The key points are summarized below:

- **High-Quality Text**: Many existing language modeling datasets contained noisy or poorly formatted text, which could negatively impact the performance of language models. WikiText-2 aimed to provide a cleaner and more coherent corpus, derived from Wikipedia articles known for their editorial standards.
- **Long-Term Dependencies**: Traditional language modeling datasets often lacked the ability to capture long-term dependencies effectively. WikiText-2 was designed to include longer context windows, enabling models to learn from and predict long-range dependencies in text.
- **Real-World Relevance**: Wikipedia articles represent a broad spectrum of human knowledge and are closer to the types of text encountered in real-world applications. This makes WikiText-2 a more relevant dataset for training models intended for practical use cases.
- **Research Advancements**: By providing a high-quality dataset, the authors aimed to facilitate research advancements in language modeling. The WikiText-2 dataset has since been used to benchmark numerous language models, contributing to significant progress in the field.
- **Pointer Sentinel Mixture Models**: The primary motivation behind the paper was to introduce a new architecture, the Pointer Sentinel Mixture Model, which combines the strengths of pointer networks and traditional language models. The high-quality and structured nature of WikiText-2 was essential for evaluating the effectiveness of this new architecture.

7 Approach

7.1 Introduction

The project was accomplished in few different phases starting with research and specification during which the research papers on DP-SGD were reviewed and understood. During this phase the project GIT repository was created and the researched paper and the associated experiments using Jupyter notebooks were committed. In this phase, the Language model for the project and the datasets for fine-tuning and analysis were also identified. Next, the analyzed Jupyter Notebook code was converted into main code in the src directory of the repository. Finally, the analysis was done using this code.

7.2 Layout of the Repository

The repository is organized into several key directories and files:

- **notebooks/: Contains Jupyter notebooks used for experimental work and analysis.**
- **papers/: Includes research papers and literature relevant to the project.**
- **src/: Holds the source code for the project, including scripts for fine-tuning the GPT-2 language model.**
- **"README.md: Provides an overview of the project, setup instructions, and usage guidelines." (“MadhumitaChaudhary/airbnb-price-prediction-linear-regression”)**

- Docs/: Project documentation

The src/ directory was used in the end to implement the main fine-tuning and DP-SGD Optimizer application code.

7.3 Experimental Work Using Jupyter Notebooks

The notebooks directory contains several Jupyter notebooks that document the experimental work conducted during the project. These notebooks include:
data_preprocessing.ipynb: Details the steps taken to preprocess the data, including cleaning, normalization, and splitting into training and test sets.
model_training.ipynb: Describes the process of training various machine learning models using the preprocessed data. It includes hyperparameter tuning, model evaluation, and performance metrics.
privacy_analysis.ipynb: Focuses on the application of selective differential privacy techniques to the trained models. It includes experiments to measure the impact of privacy mechanisms on model performance and data utility.

These experiments helped in identifying problems with TensorFlow as follows:

- TensorFlow 2.13, Keras 2.13 and TensorFlow_privacy 0.9 were compatible, however, had certain defects that required upgrades to newer version

- The Tensorflow_privacy library did not support Tensorflow >= 2.16.0
- Upgrading Tensorflow to 2.15 pulled in Keras library 3 dependencies and made incompatible software

These issues caused the Project to abandon Tensorflow ML infrastructure and use Torch Infrastructure.

A separate set of issues caused Tensorflow ML infrastructure unusable on the MAC M1/M2/M3 Metal GPU. The Project had earlier determined to use the Mac Book M3 Pro for all fine-tuning and training to reduce project cost.

7.4 Main Software

The final project involves fine-tuning the GPT-2 language model using the code in the **src** directory. This process includes:

- **data_loader.py**: Script for loading and preprocessing the dataset for fine-tuning.
- **train_gpt2.py**: Main script for fine-tuning the GPT-2 model. It includes configurations for training parameters, model checkpoints, and evaluation metrics.
- **evaluate_model.py**: Script for evaluating the performance of the fine-tuned model on various tasks, including text generation and privacy preservation.
- **gpt2_main.py**: This script orchestrates the entire fine-tuning process, integrating data loading, model training, and evaluation. It ensures that the workflow is seamless and efficient.
- **llm_ops.py**: This script is responsible for freezing different layers of the GPT-2 model and applying DP-SGD only to certain layers. By selectively applying DP-SGD, the script aims to balance privacy preservation with model performance.

7.5 Fine-tuning

The Gpt2 model was first fine-tuned using the Wiki2 dataset and tested for the overall fine-tuning work.

7.5.1 Baseline Fine-Tuning

The following figure shows the Baseline fine-tuning.

```
(/Users/pals/MICS/pt_3.10) bash-3.2$ python gpt2_main.py  -p ./gpt2_baselined -e 3 --save_model -o 1
world_size: 1, device_type: mps
optimizer: Stochastic Gradient Descent
Skip DDP setup.
Loading model from huggingface.
Robert went on a trip to Las Vegas, and  he and his wife had a beautiful time.   The girls had fun with the girls, they had their own things, a lot of fun. But their relationship with me got a little out of hand. I really needed to speak to them. When I told them I was leaving, I said, "You know, we got to talk about this. You're not going to come back." They said it was because I had been out for three days with a band. It was not because of anything I did or what happened. That was a bad relationship. The truth is, it's a good relationship, not just with my wife. We were married for a year. There was no talk of quitting or anything. She is very happy and very nice. Her life is happy, but she's not happy. The whole time she was there, she wasn't happy that I could go. If you can't get a job in a big city, you don't have any chance of finding your way out. So, I just kept her busy. And I gave her a very good job. Now, when I'm talking about my relationship as a musician, the time when she would be away from
llm_ops: starting model fine-tuning.
llm_ops: pushing llm to device mps
Training:
100%|████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████| 743/743 [15:57<00:00,  1.29s/it]
Evaluating:
100%|████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████| 77/77 [00:47<00:00,  1.61it/s]
Epoch 1, Train Loss: 2.507805629662197, Validation Loss: (2.15608494157915, 0.4287880172763659)
Training:
100%|████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████| 743/743 [15:57<00:00,  1.29s/it]
Evaluating:
100%|████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████████| 77/77 [00:47<00:00,  1.61it/s]
Epoch 2, Train Loss: 2.3250041222989477, Validation Loss: (2.122762203990639, 0.4293428823331841)
```

```
Training:
100%|████████████████████████████████████████████████████████████████████████████████
██████████████████████████████████████████████| 743/743 [15:57<00:00,  1.29s/it]
Evaluating:
100%|████████████████████████████████████████████████████████████████████████████████
██████████████████████████████████████████████| 77/77 [00:47<00:00,  1.61it/s]
Epoch 3, Train Loss: 2.2896304395285463, Validation Loss: (2.105672011901806, 0.429570841671798)
llm_ops: training total time: 50.25375297864278 mins
Robert went on a trip to Las Vegas and  made a visit to the White House to meet with President Obama.
Saved model to: ./gpt2_baselined
```

7.5.2 Baseline Testing

The following shows the testing of the Baselined model

```
/Users/pals/MICS/pt_3.10) bash-3.2$ python gpt2_main.py  -p ./gpt2_baselined --eval_model 3
world_size: 1, device_type: mps
optimizer: Stochastic Gradient Descent
Skip DDP setup.
Loading model from path: ./gpt2_baselined
Successfully loaded from checkpoint.
Robert went on a trip to Las Vegas, and  unknowingly, he was in the process of purchasing a house in a town called Largo, a
city of about 400 people. In the middle of the day, the house was rented to a friend, who had just moved to the city in his late
teens. The couple had recently been married, but she was the son of a local rancher. It was a good time to be an amateur
photographer, and they were both interested in wildlife, so they had spent a while, spending time at various places, including
the Lunkunk National Wildlife Refuge, which had been established in 1843.
prompt : The drama "Romeo and Juliet" written by
The drama "Romeo and Juliet" written by  John Wycliffe, directed by Christopher Nolan, and starring Bradley Cooper, was
directed and produced by Paul Verhoeven.
------------------------------------------------------------
prompt : Jurassic Park movie directed by
Jurrassic Park movie directed by  Brad Pitt, the film has been shown in theaters and on DVD, and it features several other
films including The Hobbit, The Lord of the Rings, Blade Runner, Ghostbusters, Alien, Ghost Rider, Jurassic Park, Avatar,
Star Wars, Spider-Man, Doctor Who, X-Men, Pirates, Buffy the Vampire Slayer, Mad Max, Sherlock, Dracula, Godzilla, Thor,
King Kong, Captain America, Batman, Superman, Black Panther, Guardians of The Galaxy, Marvel, Netflix,
------------------------------------------------------------
prompt : The Gpt2 model trained with Wiki2 dataset sucks in text completion because
The Gpt2 model trained with Wiki2 dataset sucks in text completion because  it is too slow, and a lot of data is lost during the
training run (e
------------------------------------------------------------
```

7.5.3 DP-SGD Fine-Tuning

The DP-SGD Fine-Tuning of the GPT2 model is shown below. This fine-tuning uses the Opacus library to fine-tune the model.

```
python gpt2_main.py  -p ./gpt2_dpsgd -e 3 --save_model -o 2
world_size: 1, device_type: mps
optimizer: Differential Privacy Stochastic Gradient Descent
Skip DDP setup.
Loading model from path: ./gpt2_dpsgd
Successfully loaded from checkpoint.
Robert went on a trip to Las Vegas, and  unknowingly decided to take a break from the show to watch an episode of'The Walking
Dead '. In that episode, Dr. Robert is played by David Morrissey, and is described as having been the "mastermind" of the
group at the time of their formation in the hospital room where the zombies had been released from.
Number of parameters           :  124439808
Number of Trainable Parameters :   14175744
Number of frozen parameters    :  110264064
/Users/pals/MICS/pt_3.10/lib/python3.10/site-packages/opacus/privacy_engine.py:95: UserWarning: Secure RNG turned off.
This is perfectly fine for experimentation as it allows for much faster training performance, but remember to turn it on and
retrain one last time before production with ``secure_mode`` turned on.
  warnings.warn(
/Users/pals/MICS/pt_3.10/lib/python3.10/site-packages/opacus/accountants/analysis/rdp.py:332: UserWarning: Optimal
order is the largest alpha. Please consider expanding the range of alphas to get a tighter privacy bound.
  warnings.warn(
Enabled differential privacy
llm_ops: starting model fine-tuning.
llm_ops: pushing llm to device mps
DP Training:   0%|                                  | 0/2970 [00:00<?, ?it/s]/Users/pals/MICS/pt_3.10/lib/python3.10/site-
packages/torch/nn/modules/module.py:1373: UserWarning: Using a non-full backward hook when the forward contains
multiple autograd Nodes is deprecated and will be removed in future versions. This hook will be missing some grad_input.
Please use register_full_backward_hook to get the documented behavior.
  warnings.warn("Using a non-full backward hook when the forward contains multiple autograd Nodes "
DP Training:  34%|███████████▍                     | 1000/2970 [08:49<17:13,  1.91it/s]Epoch: 1 | Step: 1000
| Train loss: 2.310 | Eval loss: 2.106 | Eval accuracy: 0.430 | ε: 4.93
DP Training:  67%|██████████████████████▍          | 2000/2970 [18:24<08:32,
1.89it/s]Epoch: 1 | Step: 2000 | Train loss: 2.288 | Eval loss: 2.105 | Eval accuracy: 0.430 | ε: 5.46
DP Training: 3000it [28:01,  1.85it/s]Epoch: 1 | Step: 3000 | Train loss: 2.287 | Eval loss: 2.105 | Eval accuracy: 0.430 | ε:
5.84
DP Training: 3278it [31:18,  1.75it/s]
DP Training:  34%|███████████▍                     | 1000/2970 [08:36<16:38,  1.97it/s]Epoch: 2 | Step: 1000
| Train loss: 2.282 | Eval loss: 2.105 | Eval accuracy: 0.430 | ε: 6.23
DP Training:  67%|██████████████████████▍          | 2000/2970 [18:02<08:13,
1.96it/s]Epoch: 2 | Step: 2000 | Train loss: 2.279 | Eval loss: 2.105 | Eval accuracy: 0.430 | ε: 6.50
DP Training: 3000it [27:29,  1.90it/s]Epoch: 2 | Step: 3000 | Train loss: 2.287 | Eval loss: 2.105 | Eval accuracy: 0.430 | ε:
6.74
DP Training: 3294it [30:50,  1.78it/s]
DP Training:  34%|███████████▍                     | 1000/2970 [08:38<16:34,  1.98it/s]Epoch: 3 | Step: 1000
| Train loss: 2.282 | Eval loss: 2.105 | Eval accuracy: 0.430 | ε: 7.04
DP Training:  67%|██████████████████████▍          | 2000/2970 [18:03<08:23,
1.93it/s]Epoch: 3 | Step: 2000 | Train loss: 2.302 | Eval loss: 2.105 | Eval accuracy: 0.430 | ε: 7.25
DP Training: 3000it [27:29,  1.96it/s]Epoch: 3 | Step: 3000 | Train loss: 2.292 | Eval loss: 2.105 | Eval accuracy: 0.430 | ε:
7.44
DP Training: 3284it [30:44,  1.78it/s]
llm_ops: DP training total time: 92.89597291549047 mins
Robert went on a trip to Las Vegas, and  he met with his fiancee and a number of friends and acquaintances to talk about his
experiences in the business. He also interviewed a friend in his early twenties who was working in a fashion business at the
time.
Saved model to: ./gpt2_dpsgd
```

7.5.4 DP-SGD Fine-Tuned Model Testing

The following shows the test results of the fine-tuned Gpt2 model using DP-SGD:

```
(/Users/pals/MICS/pt_3.10) pals-mbp-m3:src pals$ python gpt2_main.py  -p ./gpt2_dpsgd/ --eval_model 3 -o 2
world_size: 1, device_type: mps
optimizer: Differential Privacy Stochastic Gradient Descent
Skip DDP setup.
Loading model from path: ./gpt2_dpsgd/
Successfully loaded from checkpoint.
Robert went on a trip to Las Vegas, and  sunk the head of a horse which he had captured for the purpose, which was to be his new home
in a "wilderness town".
Number of parameters          :  124439808
Number of Trainable Parameters :   14175744
Number of frozen parameters    :  110264064
prompt : Romeo and Juliet in Shakespeare ballad met in
Romeo and Juliet in Shakespeare ballad met in vernacular by a young Henry VII., who was a member of the Royal Court of England
and was the heir to the throne of Charles VII. They had been married in 1729 and had two children; and in their first year of marriage
they married two sons of Philip II., and also two daughters of Henry VIII.
-----------------------------------------------------------
```

prompt : Steve Jobbs founded Apple Inc., however, was
Steve Jobbs founded Apple Inc., however, was a business partner of the Clinton Foundation, the main fund of which was run by the wealthy and powerful Clinton family..
----
prompt : The list of US presidents assassinated during their term indicates that
The list of US presidents assassinated during their term indicates that Obama is the first president assassinated since the end of the Cold War (1936-42).
----

8 Setbacks, Results and Conclusion

8.1 Setbacks

The following setbacks impacted the project:

- Tensorflow: Tensorflow, Tensorflow_privacy and Keras ML infrastructure software packages version incompatibilities.
- Keras: NLP Support in Keras 3 injected dependencies breaks Tensorflow_privacy library.
- Opacus: PyTorch library for privacy supports only tiny LLMs with few 100 Million parameters and failed training a full Gpt2 model. The training was possible only by freezing almost all the layers except for 2 hidden layers.
- GCP: Google Colab and GCP does not have a good GPU support, even for a single GPU instance.
- AWS: supports single GPU instance; however, multiple GPUs are supported in 12XLarge or greater instances. These instances are expensive and requires larger quota from AWS.

8.2 Results

The Project provided the following:

- Provided an avenue for a solid hands-on understanding of the ML Gradient Descent and the derived algorithms.
- Provided scope to learn Tensorflow, PyTorch, Keras, Opacus and other libraries
- Enabled a solid understanding of Differential Privacy and get hands-on development in DP
- Provided a platform to learn LLM Fine-Tuning, Evaluation and the fundamentals of NLP features

8.3 Next Steps

The Team intends to continue this effort with the help of community:

- Bring awareness on the need for scalable privacy enabling software for NLP/LLM
- Work on the Opacus software for making it scalable.

9 References

[1] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, Yue Zhang, 2024, A Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly, https://doi.org/10.48550/arXiv.2312.02003
[2] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3- 4):211–407.
[3] Martin Abadi, Andy Chu, Ian Goodfellow, H Bren- dan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pages 308–318.*
[4] Weiyan Shi, Aiqi Cui, Evan Li, Ruoxi Jia, Zhou Yu, 2022, Selective Differential Privacy for Language Modeling, https://doi.org/10.48550/arXiv.2108.12944
[5] Weiyan Shi, Ryan Shea, Si Chen, Chiyuan Zhang, Ruoxi Jia, Zhou Yu , 2022, Just Fine Tune Twice, Selective Differential Privacy for Large Language Models https://ar5iv.labs.arxiv.org/html/2204.07667