Using autoencoders on differentially private federated learning GANs

Gregor Schram
Supervisors: Kaitai Liang, Rui Wang
EEMCS, Delft University of Technology, The Netherlands

June 19, 2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Machine learning has been applied to almost all fields of computer science over the past decades. The introduction of GANs allowed for new possibilities in fields of medical research and text prediction. However, these new fields work with ever more privacy-sensitive data. In order to maintain user privacy, a combination of federated learning, differential privacy and GANs can be used to work with private data without giving away a users' privacy. Recently, two implementations of such combinations have been published: DP-Fed-Avg GAN and GS-WGAN. This paper compares their performance and introduces an alternative version of DP-Fed-Avg GAN that makes use of denoising techniques to combat the loss in accuracy that generally occurs when applying differential privacy and federated learning to GANs. We also compare the novel adaptation of denoised DP-Fed-Avg GAN to the state-of-the-art implementations in this field.

1 Introduction

Over the past two decades, machine learning and the broader field of artificial intelligence have grown exponentially. Nowadays, one may find these techniques applied everywhere, from fraudulent transaction detection by banks [1] to self-driving cars and advanced autocorrect and typing aids on smartphones. The expanded usage of these systems however comes at a cost: data. In general, an increase in accuracy of a machine learning model requires an increase in training data. This is problematic for multiple reasons, one of the reasons being the absence of sufficiently large datasets and another being the privacy invasions when using certain datasets.

For this first problem, Ian Goodfellow [2] and his colleagues came up with the technique of Generative Adversarial Nets (GANs) in 2014. This technique makes two machine learning models compete, with one being a discriminator and the other trying to generate examples of data that did not exist in the training set previously to fool the discriminator. This process in the end results in a much more accurate discriminator while requiring a much smaller dataset to begin with.

The second issue is one that has become more relevant in recent years as machine learning gets applied to more and more fields, the data it requires might be very privacy-sensitive. Think for example of medical data, pictures on a user's phone, or all the text input generated on someone's smartphone. Most people do not feel comfortable sharing this data with the large companies that often require it to improve their machine learning models. This is where Differential Privacy (DP) comes in, it was first brought up by Dwork and colleagues [3] but some mathematics underlying the idea stem from Dalenius [4]. Differential privacy ensures that analysis of a dataset as a whole can happen while preserving the privacy of any individuals' data.

Another problem arising with the use of machine learning on private data, especially when this data is coming from phones, is that the dataset is not centralized. Conventional machine learning requires the dataset to be completely accessible at all times. This is not a good requirement when dealing with private and decentralized data, again like pictures or text input on a phone. This is where Federated Learning (FL) comes into the picture. Federated learning allows private and decentralized data to stay on edge devices, like a phone, while still benefiting from all the data available to improve the models globally. The term as used

in this paper was originally put forward by researchers at Google [5].

Now that we understand the importance of differential privacy, federated learning and GANs we can also see why it is important to combine these. Differential privacy inherently makes a model less accurate which can be combatted by GANs, these however rely on centralized datasets which is why we need federated learning to apply it in real world use cases. The first paper combining these three techniques was only published two years ago by Sean Augenstein and his team at Google [6] with a real world use case being researched in the same year [7].

Since the convergence of these three techniques is very novel, this paper answers the question: Does adding a denoising step to DP-FL-GANs increase model accuracy while preserving privacy? This paper will first look at state-of-the-art techniques and add to them with a novel application of denoising for differential privacy.

This paper starts with an enumeration of related works in section 2. It then introduces a novel technique in section 3 and continues with an overview of the experiment setup in section 4. Section 5 gives an overview of the experiment results, while section 6 compares these to state-of-the-art implementations. Section 7 describes the ethics of this research, while section 8 contains the conclusions.

2 Related Work

While the field of differentially private Generative Adversarial Nets (GANs) is quite developed and well researched, the subfield of federated solutions is very new and not well researched yet. A few implementations do exist, and we will be comparing our solution to those.

DP-Fed-Avg GAN [6]. The first implementation and paper that proposed to apply differentially private GANs to a Federated Learning (FL) setting. The paper introduces the DP-Fed-Avg GAN algorithm, an adaptation of the Fed-Avg algorithm [8], to train both the discriminator and the generator in a federated setting while keeping user-level Differential Privacy (DP) guarantees under a trusted server. This trusted server is required because the averaging happens on the server and not on the edge devices.

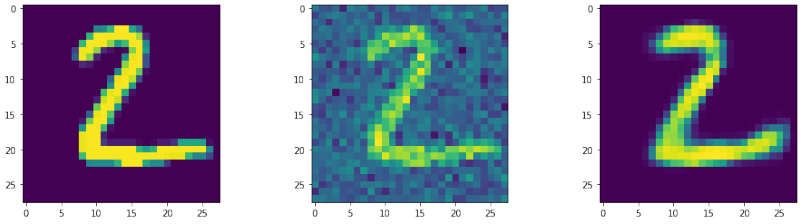
GS-WGAN [9]. This paper introduces a new way to use gradient information, enabling deeper models that can generate more useful samples representing a private dataset. Moreover, it allows this to be done both in a centralized and in a federated setting. In the federated setting, the proposed algorithm can provide user-level DP guarantees under an untrusted server, as gradient sanitization happens before sending the update. It also does not require rigorous hyperparameter tuning, something that DP-Fed-Avg GAN does require.

3 Proposed method

This paper proposes to first apply Gaussian noise to local private training data and denoise it before training. Most Differential Privacy (DP) methods make use of Gaussian noise added to training data before training in order to enable user-level DP guarantees. The addition of

noise inherently makes it more difficult for the Generative Adversarial Net (GAN) to generate and discriminate images reliably, the more noise one adds, the lower the accuracy of the GAN. Recently, there have been significant advances in noise removal on images [10] [11]. This noise removal will never lead back to the exact original image. This should preserve privacy as the original data is still permutated while resulting in higher accuracy due to more distinguishable training data.

The originally proposed denoising algorithm was to be based on the work by Majed El Helou and Sabine Susstrunk [12]. Later, we made the choice to focus on a simpler autoencoder, based on [13]. We made this choice because our implementation works with the EMNIST dataset. This dataset is quite simple and uniform and thus does not benefit much from a very advanced algorithm as presented by [12]. The usage of an autoencoder also makes the experiment run quicker and more reliable. The effectiveness of an autoencoder can be seen in figure 1. As we can see, most of the noise is removed, making it easier to recognize the shape of the sample and thus making the training of the GAN more effective.



(a) Original sample (b) Sample with noise (c) Sample after autodecoding

Figure 1: An illustration of the effectiveness of an autoencoder

As we can see in Figure 2 our modified DP-Fed-Avg GAN algorithm goes through quite a few steps, the new step that we added has been highlighted. As one may understand from the flowchart, the removal of noise does not change the differential privacy level, as this happens after retraining the local GAN. Only when touching the input of the local GAN training session would this change.

4 Experiment Setup

For our experiment, we first looked at the currently existing methods of Differential Privacy (DP) Federated Learning (FL) Generative Adversarial Nets (GANs). As of writing, two exist: DP-Fed-Avg GAN [1] and GS-WGAN [2]. For both of these, the authors provided an implementation. These implementations were released with the intent to reproduce the results in the respective papers. Naturally, in order to compare these implementations against each other and against our implementation, we tried to reproduce the results produced in the respective papers [6] [9].

We used *conda* to create an environment for each of the solutions to run in isolation and to make sure that the setups were as close as possible to the ones used to generate the

[1] https://github.com/google-research/federated/tree/master/gans
[2] https://github.com/DingfanChen/GS-WGAN

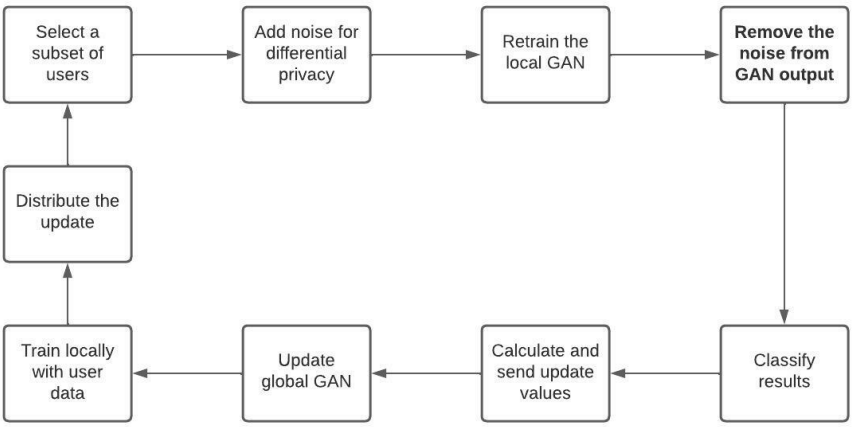


Figure 2: An overview of the steps in our modified DP-Fed-Avg GAN with noise removal

results in [6] and [9]. This ultimately allowed us to recreate some results of the DP-Fed-Avg GAN paper, but unfortunately we were not able to do this for the GS-WGAN paper as the provided code was incomplete and the federated setting was missing from it.

For the reproduction of the DP-Fed-Avg GAN results, we used the EMNIST dataset [14], this is an extension of MNIST [15], unlike MNIST it has letters as well as numbers and makes a distinction in capitalization, just like the original paper did. Since we are not interested in the practical use case of finding a bug, as introduced in [6] we did not introduce said bug in the experiments we ran. In order to do this, we selected only users with an accuracy over 93%, and we set the image inversion probability to 0. Other than these values, we kept the exact same hyperparameters as used in the original paper.

For the new method proposed in section 3 we will make use of DP-Fed-Avg GAN. The reason we chose this algorithm as a base is that it is the only algorithm that has working code available, and implementing all the layers of GANs, FL and DP seemed unfeasible in the short timeframe of this research. Our method adds an extra step after each training round to remove noise on the intermediate results, which are also used in the next round, of the DP-Fed-Avg GAN. This extra step is based on the autoencoding algorithm described in 3. The code and setup instructions for our experiment can be found on GitHub[3].

As the original algorithm uses the EMNIST dataset, we also use this for our implementation. This requires some slight modification to the denoising algorithm, as it is designed for the MNIST dataset. We will use the same hyperparameters described earlier. However, this time we will use different noise levels as this affects the effectiveness of the denoising algorithm and the overall accuracy of our GAN.

[3]https://github.com/gregor160300/federated

5 Results

Our implementation has a very similar approach for applying differential privacy to a federated learning Generative Adversarial Net (GAN) as the DP-Fed-Avg GAN algorithm, we actually reuse most of the code from the DP-Fed-Avg GAN paper [6]. The only thing we added on top of the DP-Fed-Avg GAN algorithm is a step to denoise the intermediate outputs of the GAN. This did not touch the Differential Privacy (DP) level of the GAN, this was also not the intent. The intent was to reach a higher accuracy on classification of generated output, as well as reduce the Frechet Inception Distance (FID). Unfortunately, our experiment failed.

(a) No DP and no noise removal

(b) DP but no noise removal

(c) DP and noise removal

Figure 3: An illustration of the impact of DP and denoising on the accuracy of the GAN

Let us first take a look at the GAN output after 1000 rounds of training in figure 3. We see in figure 3a that the generated images when not using differential privacy nor noise removal are quite readable. In 3b we see a slight degradation in the readability due to the addition of DP guarantees. In the results of our implementation in 3c we see that all outputs have some sort of gray layer on top of them making it less readable and thus increasing the FID and lowering the accuracy of classification based on these generated results.

The way the autoencoder works for noise removal is to train a network on noisy input and non-noisy input. It then learns to remove noise from a noisy image by recognizing the differences. It is however trained on one specific noise level, in our case 20%. This means that it will learn to reduce the noise best at 20% noise, however our results might have more or less noise. The autoencoder will still try to remove the noise, but white noise on a black background then just becomes a gray filter on top of the entire image as it is unsure how to remove the noise. Our concept of noise removal might work with an algorithm that is agnostic of the noise level.

6 Discussion

The current implementations of DP-Fed-Avg GAN and GS-WGAN both have corresponding papers [6] [9] with results. We will focus on the model accuracy of the results and the differential privacy, as these are integral to our problem of increasing accuracy while preserving privacy. Differential privacy is measured by a value, ϵ this value ideally is between 0 and 1, the lower, the better. The accuracy is measured by the Frechet Inception Distance (FID) as Generative Adversarial Nets (GANs) do not do classification, we check the similarity of

the generated results to the training samples. The lower the distance between the two, the better.

We will first take a look at the claimed performance of the DP-Fed-Avg GAN. [6] claims that the DP-Fed-Avg GAN reaches an epsilon of 9.99×10^6 in the simulation run in code. This is far above any useful privacy, as this basically means the chance that our GAN generates an actual training input (thus breaking privacy) is enormous. However, the researchers note that this is due to the small sample size of users (around 10 of a total of 3000) in the simulation. In a realistic scenario, with millions of users, they note an epsilon around the 1.4 mark. This is a much more practical privacy preservation level. As for the FID, we see a claim around the 200 mark, which is not amazing, but considering the noise added for differential privacy, it is definitely acceptable.

The GS-WGAN paper [9] claims quite different results. The epsilon value they manage to reach is 5.99×10^2, which is significantly lower than the value reached by the DP-Fed-Avg GAN. Note that this claim is also in the same simulated setup as the DP-Fed-Avg GAN, no claim is made however about the privacy level in a more realistic scenario. As for the FID, it is also significantly lower at just over 60, compared to the more than 200 with Fed-Avg GAN. Overall, GS-WGAN seems to be the more performant algorithm, both in terms of privacy preservation and GAN performance.

There is however one major sidenote to take on the GS-WGAN, that is that all these results should be taken at face value as there is no code available at the time of writing to reproduce these results. On the contrary, we were able to reproduce the results found in the DP-Fed-Avg GAN paper within a margin of error. The sidenote here being that we were only able to verify the performance of the simulation and not for the realistic setting. In summary, we thus have a quite performant unverified algorithm in GS-WGAN and a less performant, partially verified algorithm in DP-Fed-Avg GAN.

	GS-WGAN*	DP-Fed-Avg GAN	Ours
Frechet Inception Distance ↓	60	0.2 * 10^3	2.5 * 10^3
Generator loss ↓	Unknown	-0.6	-0.5
Classifier accuracy ↑	Unknown	60%	18%
Epsilon ↓	5.99 * 10^2	9.99 * 10^6**	9.99 * 10^6**

Table 1: A performance comparison of our algorithm against GS-WGAN and DP-Fed-Avg GAN. ** claimed, but unverified values used, ** experimental setting value, lower in real world scenarios*

When we compare the performance of our method of applying an autoencoder to the DP-Fed-Avg GAN, we unfortunately see worse performance than both DP-Fed-Avg GAN and GS-WGAN. As we can see in Table 1 the Epsilon value, which indicates the differential privacy level, stays the same from DP-Fed-Avg GAN to our modification of it, however the FID as well as the classifier accuracy is significantly lower. These lower values are most likely caused by the issue described in the Results section.

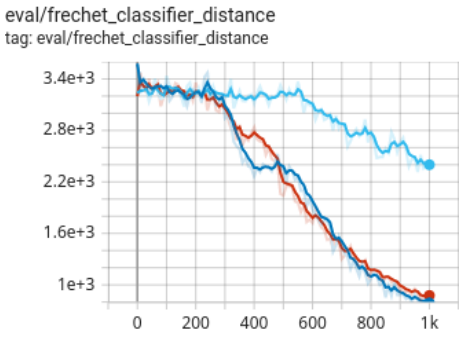
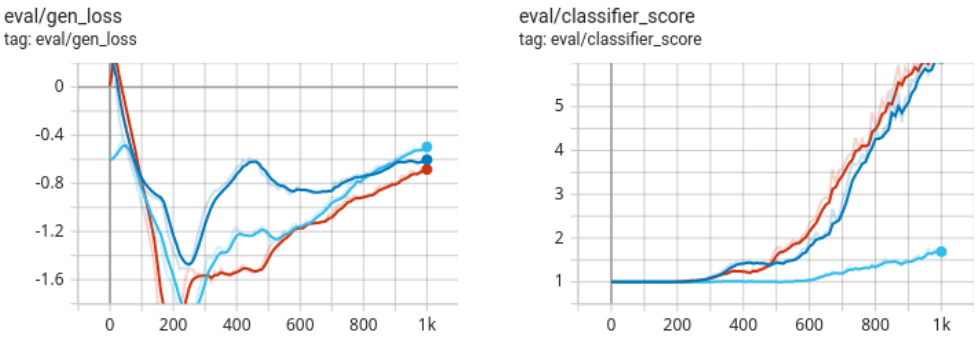


Figure 4: Frechet Inception Distance: Red = Fed-Avg GAN without DP, Light Blue = Ours, Blue = Fed-Avg GAN with DP

When we look into more detail at the performance over the 1000 iterations, we can see in Figure 4 that the further we proceed, the lower our FID becomes. This makes sense intuitively, as the GAN will get better at creating results similar to its input with more training. What we also see is that this improvement goes much slower with our implementation than the others. This is logical, considering a gray blur is added in our results due to the failing denoising algorithm. In future research, one might try to run the algorithm for more iterations to see if the FID slowly keeps declining or plateaus.

What is interesting to see is that the generator loss (see Figure 5a for our algorithm does not seem to be markedly higher or lower than the regular DP-Fed-Avg GAN. Unfortunately, we could not find a clear reason for this behavior. When we look at the classifier score (see Figure 5b), remember a GAN has usually trains against a classifier, we see a notable decrease in accuracy with our implementation due to previous described issues with our experiment. Here again, it might be interesting to see how the accuracy progresses after 1000 rounds when doing future research into optimization of the DP-Fed-Avg GAN algorithm.



(a) Generator Loss: Red = Fed-Avg GAN without DP, Light Blue = Ours, Blue = Fed-Avg GAN with DP

(b) Classifier score: Red = Fed-Avg GAN without DP, Light Blue = Ours, Blue = Fed-Avg GAN with DP

Figure 5: Generator loss and classifier score

7 Responsible Research

During our research, at some points we had to make choices that could be considered ethically questionable. The main choice that was made was to include the results of GS-WGAN without being able to verify them. We made this choice to still be able to compare our results to the state of the art, we considered the inclusion necessary because our algorithm is based on DP-Fed-Avg GAN and thus comparing exclusively to that would be putting our results in a vacuum. To make sure there is no confusion about the results of GS-WGAN we stated that these results are unverified at multiple points in the paper.

Continuing from this point, we noticed that reproducing experiment results was impossible for GS-WGAN as the code was not published yet. And for DP-Fed-Avg GAN we were eventually able to get the code working, but it took a lot of effort to figure out how to set it up, mostly due to dependencies on specific Python package versions, without this being stated. These experiences were quite frustrating, that is why we will have clear instructions, including those required for a proper installation of the dependencies, in the readme file of the GitHub repository that hosts the code.

Another thing to consider is the impact of this research on society. We believe that this research can not be used to cause harm to anyone, as it just considers how to make private machine learning algorithms more efficient. Doing so to our knowledge can not cause anyone to be harmed as their privacy is being preserved while having better functionality in end products that use these techniques.

8 Conclusions

This paper introduced the idea of using autoencoders on Differentially Private (DP) Federated Learning (FL) Generative Adversarial Net (GANs). This novel approach applied an autoencoder on top of the DP-Fed-Avg GAN. The idea is to reduce the noise on the GAN output and thus improve the readability of the output, making it easier to classify or

spot weird outputs. Unfortunately, due to the autoencoder being trained at a specific noise level and the differential privacy calculations adding arbitrary levels of noise, this technique failed. The concept could still work when tried with a noise removal algorithm that does not require knowing the noise level upfront.

This paper also compared the performance of the novel approach to the existing GS-WGAN and DP-Fed-Avg GAN algorithms. These results showed that both in terms of privacy and generated results, the GS-WGAN algorithm is currently the best. The paper also addressed some questions with regard to the ethics of this research and the societal impact thereof.

References

[1] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *2017 international conference on computing networking and informatics (ICCNI)*, pp. 1–9, IEEE, 2017.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[3] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*, pp. 265–284, Springer, 2006.

[4] T. Dalenius, "Towards a methodology for statistical disclosure control," *statistik Tidskrift*, vol. 15, no. 429-444, pp. 2–1, 1977.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[6] S. Augenstein, B. McMahan, D. Ramage, S. Ramaswamy, P. Kairouz, M. Chen, R. Mathews, and B. Aguera-Arcas, "Generative models for effective ml on private, decentralized datasets," 2019.

[7] S. Ramaswamy, O. Thakkar, R. Mathews, G. Andrew, H. B. McMahan, and F. Beaufays, "Training production language models without memorizing user data," *arXiv preprint arXiv:2009.10031*, 2020.

[8] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[9] D. Chen, T. Orekondy, and M. Fritz, "Gs-wgan: A gradient-sanitized approach for learning differentially private generators," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12673–12684, 2020.

[10] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[11] J. W. Soh and N. I. Cho, "Deep universal blind image denoising," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 747–754, IEEE, 2021.

[12] M. El Helou and S. Süsstrunk, "Blind universal Bayesian image denoising with Gaussian noise level learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 4885–4897, 2020.

[13] "Intro to autoencoders: Tensorflow core."

[14] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926, IEEE, 2017.

[15] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.