Chapter

Beyond Differential Privacy: Synthetic Micro-Data Generation with Deep Generative Neural Networks

Ofer Mendelevitch and Michael D. Lesh

Abstract

Recent advances in generative modeling, based on large scale deep neural networks, provide a novel approach for sharing individual-level datasets (micro-data) without privacy concerns. Unlike differential privacy, which enforces a specific query mechanism on data to ensure privacy, generative models can accurately learn the statistical patterns of such micro-data and then be used to generate "synthetic data" that accurately reflects these statistical patterns, yet contain none of the original data itself, and thus can be safely shared for analysis and modeling without compromising privacy. The successful application of these techniques to various industries including healthcare, finance, and autonomous vehicles is promising and results in continued investment in research and development of generative models in both academia and industry.

Keywords: generative models, synthetic data, deep neural networks, micro-data

1. Introduction

Differential privacy, created more than a decade ago, continues to play an important role in protecting privacy of micro-data while enabling statistical analysis. Initially applied by statistics agencies such as the US census bureau, it is now well recognized that, although useful for some applications, differential privacy comes with significant limitation (e.g., [1]).

To understand some of the limitations of differential privacy, consider the following:

- Differential privacy is defined around the concept of a *mechanism*; as such, it is not intended to create "sharable datasets," but instead allows a user (analyst) to submit various types of queries (via the defined *query mechanism*), requesting some kind of aggregate statistics, like summary statistics of the original data. This limits the usability of differential privacy to queries that are supported by that mechanism.

- An appropriate *privacy budget* needs to be decided upon, and in practice it's often difficult to agree on what that budget needs to be. In fact, practical

Figure 1.
Fake celebrity images created using generative modeling; none of these images are real people.

use-cases demonstrate that due to concerns about risk, most implementations end up with much higher budget than is necessary.

- Many mechanisms of differential privacy require noise to be added to the data in cases where the original data is highly skewed, resulting in reduced utility of the outputs, and in some cases rendering the whole exercise useless.

- In many specific fields of statistical analysis, users of micro-data are highly trained to use specific tools (STATA, SAS, R and Python) and query procedures, which often do not support the complexity of differential-privacy-protected mechanisms. This presents a behavior-change challenge whereby analysts need to be convinced to abandon their familiar methods and tools (which they may have been using for decades) in favor of the interactive system where the privacy-protected data is available.

Fortunately, deep generative models – a recent and novel approach in deep neural networks – provide an alternative for direct sharing of micro-data without privacy risk.

With generative models, a deep neural network algorithm uses the existing micro-data to approximate, with high accuracy, the underlying probability distribution of the data in some high-dimensional latent space. Once the probability distribution is approximated, the trained model can be used to generate any number of *synthetic* records by randomly sampling from that distribution. Those generated records are related to the original data only through the shared underlying probability distribution, and thus does not include any information that can be linked back to the original (private) records.

To further illustrate how synthetic data generation works, consider CelebA,[1] a dataset with more than 200,000 synthetic celebrity face images, each with 40 automatically extracted attribute annotations. Using generative models, researchers have demonstrated the ability to learn the underlying distribution well enough to generate photorealistic celebrity faces as is shown in **Figure 1** above.

[1] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

This same technique can be applied to many other types of data – music, text, videos as well as healthcare, financial or insurance data. In this chapter we will explore synthetic data generation and its application, and how releasing synthetic micro-data can provide an alternative to differential privacy.

In Section 2, we explore synthetic data in more detail, and how generative models can create synthetic data. In Section 3, we discuss using variational auto-encoders as generative models, followed by Section 4, where we discuss generative adversarial networks. In Section 5, we discuss the application of generative models to healthcare data, and in Section 6, we discuss privacy in the context of synthetic data, and some approaches that combine differential privacy with synthetic data generation. Section 7 is a summary and discussion on future directions in synthetic data generation.

2. Generative models for synthetic data

Generative models are a class of mathematical models that approximate a probability distribution of some dataset and can be used to generate samples of data according to the modeled (or approximated) distribution. Such generated data is often called "synthetic data," "fake data," or "realistic but not real."

For a given data domain, consider a dataset A with N data records. For most practical cases, the dataset can be assumed to be drawn from some (usually unknown) probability distribution P(x). A *synthetic* dataset S is a dataset similar to A in terms of fields or structure, where records in S are randomly drawn from some probability distribution Q(x).

In an ideal world where Q(X) = P(X) we can clearly use S for various purposes of analysis and modeling, because they are sampled from the same distribution. The key idea behind generating synthetic data is as follows: can we accurately estimate this probability distribution P(x), such that Q(X) ≈ P(X), with high fidelity?

Let us look at a simple example – consider a one-dimensional series of values A, where A is drawn from a normal (Gaussian) distribution with mean μ and standard deviation σ. In other words, we know in this case that P(x) is the normal distribution with $P_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$, and that the values in A should fit this distribution. We can then use Gaussian fitting to estimate the values of μ and σ from the data, as is demonstrated in **Figure 2**.

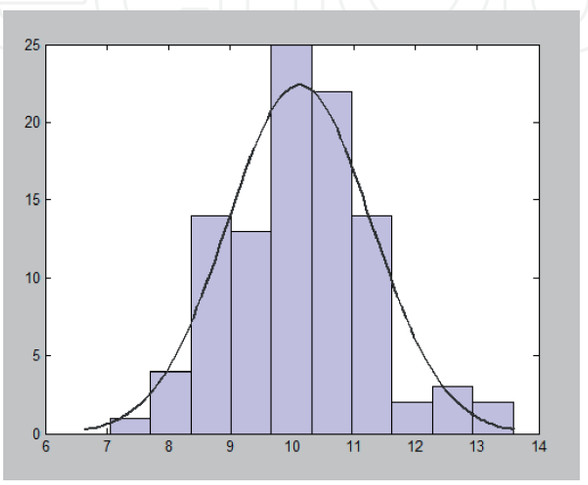

Figure 2.
Sample Gaussian fitting.

Once we have a good approximation for the parameters of the distribution (μ and σ), we can sample from this distribution to generate completely new data points that are fully consistent with the Gaussian distribution describing the original data.

This is of course a simplified example for two reasons. First, with a real generative model we do not know the actual form of the distribution function (e.g., Gaussian in this case); instead we use the neural network to estimate that function. Second, in the real world the data is not one-dimensional, but of much higher dimension.

So how do we approximate an unknown probability distribution from high-dimensional data?

The traditional approach to approximating data distribution is simple frequency counting (histograms), but of course this approach does not work in high dimensions due to the curse of dimensionality, namely the fact that most statistical methods fail in high-dimensional data due to increasing sparseness. This is also the case here with frequency counting, where with many dimensions the amount of histogram needed quickly explodes to make the method unfeasible.

Instead, the approach used in modern generative modeling research is to assume a functional form of the distribution $P_\theta(x)$ and learn the parameters θ of the function from the data. This set of parameters θ is in essence a compressed representation for the original dataset, often called "latent space representation."

To further illustrate this, let us go back to our example of celebrity images. Assume that the images are black and white (so that each pixel is represented by either 0 or 1), and of size 28 × 28 = 784 pixels. If we represent each image as a vector of 784 binary values, the number of possible values for a vector in this space is $2^{784} = 10^{236}$; if we want to approximate P(x) for each possible vector x in this space, we would need to estimate it for 10^{236} such vectors, which is clearly not realistic in practice (thus "the curse of dimensionality"). Instead, we can define some $P_\theta(x)$ with a much smaller set of parameters θ and estimate those parameters in such a way that $P_\theta(x) \approx P(x)$. It turns out that deep neural networks are a good match for this kind of problem, and can be used to accurately estimate the parameters of the distribution $P_\theta(x)$; there are many possible neural network architectures suitable for this task, most common of which are auto-encoders and generative adversarial networks.

Images are a very vivid (pun intended) demonstration of the power of generative models and how they can generate high utility synthetic data; but these techniques can also be successfully applied to many other fields such as music, poetry, cartoon characters, or even synthetic "video miles" for self-driving cars.

The performance of recent techniques in generative modeling is quite impressive, and their success led to a growth in applications of generative models in industry. For example, self-driving car companies use synthetic data to significantly increase the size of training data they have available, covering many more scenarios and edge-cases for improving their self-driving algorithms.

The usefulness of synthetic data generally falls into one of 3 important categories:

- **Replacement**. If access to the real dataset is limited or restricted (e.g., when data access is highly regulated), synthetic data often provides an excellent alternative. A good example comes from healthcare – access to medical records is often heavily restricted because of personal identifiers and the risk of linkage attacks. Synthetic medical records with high fidelity can provide the medical and bio-pharma research community with a replacement dataset that accurately reflects the statistical properties of the original data. This opens up an enormous opportunity to share and aggregate medical data from various clinical care sources and unlock important insights such as how effective are various therapeutics like drugs, medical devices or clinical care protocols.

- **Augmentation**. In many predictive modeling use-cases, the dataset available for training the model is relatively small in size, which often results in lower accuracy of the model. This phenomenon is further exacerbated when using deep learning for predictive modeling, where small datasets tend to overfit quite easily. Creating synthetic training examples and combining the real and synthetic data points ("augmenting" the real dataset with synthetic data), resulting in a much larger training dataset overall, can significantly improve the accuracy of the predictive models.

- **Equalization/reshaping**. An interesting aspect of using generative models is that we can generate as much data as is desired; often many more records than exist in the original dataset. A key characteristic of generative models is that we can direct them to shape the output dataset to certain desired criteria. For example, if the original dataset has 60% male and 40% female, we can control the gender distribution and generate a 50%/50% synthetic dataset. This enables users of the synthetic data to battle bias in the original dataset.

Equipped with a basic understanding of what synthetic data is, and how it's created using generative models, let us look in more detail at two of the most common types of generative models: variational auto-encoders and generative adversarial networks.

3. Variational auto-encoders

An autoencoder is a specific type of deep learning architecture which is split into two distinct neural networks: one is called the "encoder" and the other "decoder," as is shown in **Figure 3**.

In this architecture, the encoder E_θ is a deep neural network that encodes the input data (X) into some intermediate representation (Z, often referred to as "latent representation") in a reduced dimensional space, and the decoder D_θ is also a deep neural network that decodes the vector Z back into the output vector Y. X and Y are of the same dimensionality. The goal of training the auto-encoder is to reconstruct the input X in the output Y, while transitioning through the lower dimensionality representation Z, so that we get as close as possible to $Y = D_\theta(E_\theta(X))$. If you optimize this auto-encoder in such a way that the loss of data between input X and

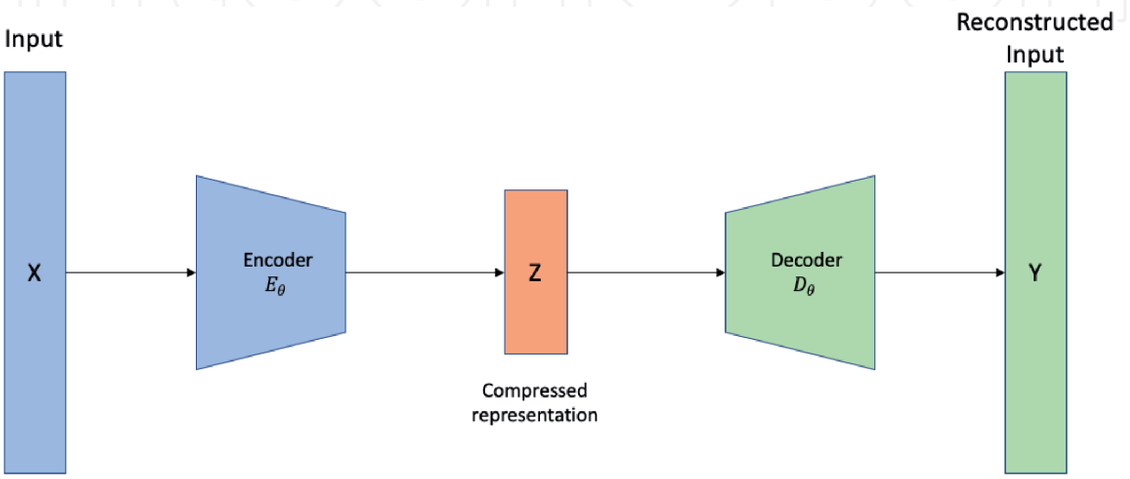

Figure 3.
Auto-encoder architecture.

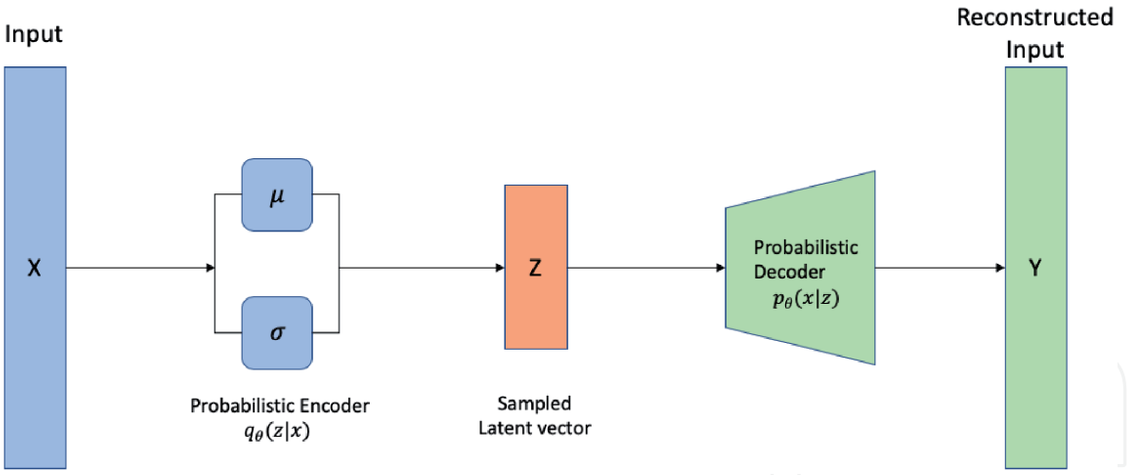

Figure 4.
Variational auto-encoder architecture.

output Y (reconstruction error) is minimized, then it's as if you are trying to find an optimal compressed representation for the input data.

Traditional auto-encoders have been around since the early days of neural networks and in their basic form they cannot be used to generate synthetic data; In 2013 the idea of *variational* auto-encoders (VAE) started to take shape, primarily with the work of [2, 3], as a way to use auto-encoders as generative models.

With VAEs, instead of mapping the input vector X to a fixed vector Z, we want to map it into a distribution $q_\theta(z|x)$, often assumed to be a multivariate normal distribution with mean μ and standard deviation σ; then to generate synthetic outputs Y we just randomly sample this learned distribution and decode the sampled vector to arrive at a synthetic output Y, as shown in **Figure 4**.

VAEs, being one of the first deep neural network architectures for practical generative models, created a lot of excitement about synthetic data, and was used primarily to generate synthetic images. Although elegant and theoretically pleasing, the synthetic images generated by VAEs tend to be blurry, which very quickly became a limiting factor for their use in synthetic imaging. Various improvements to the basic VAE approach have been proposed such as beta-VAE [4] and VQ-VAE [5] to address these issues; however, this also led researchers to the idea of generative adversarial networks, which we discuss next.

4. Generative adversarial networks

The idea of a generative adversarial network is inspired by game theory: we build two models, a generator and a discriminator, that compete with each other in an adversarial manner to collaboratively optimize the whole system. The generator G is a generative neural network that outputs synthetic samples given a noise variable Z. The discriminator D is a different neural network that is trained to discriminate between real and synthetic samples. During training, the generator is trying to generate samples that mimic as much as possible the real data, so that it can fool the discriminator, whereas the discriminator is trained not to be fooled and be able to distinguish between real and synthetic data samples. This is shown in **Figure 5**.

As you can see from **Figure 5**, a key idea in this architecture is that the discriminator D shares gradient updates with the generator, such that the generator can "understand" how its generated data fails to fool the discriminator and improve its generation over time resulting in better and better synthetic samples.

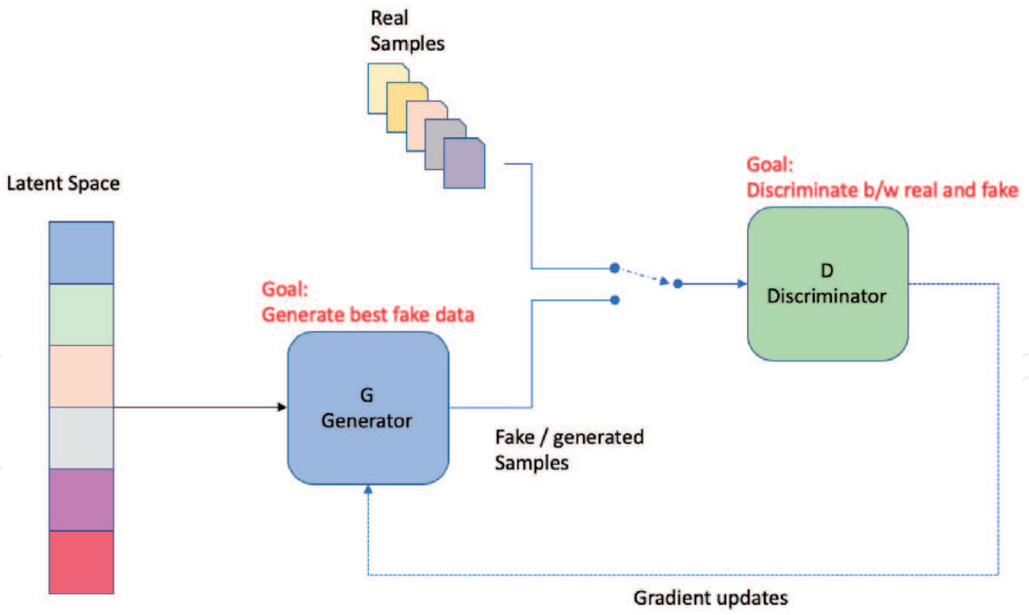

Figure 5.
Generative adversarial networks.

GANs were first formulated by Ian Goodfellow and colleagues [6], and since then have been an active area of research; they have demonstrated the ability to generate significantly better synthetic images than VAEs, and have been used in a variety of applications like generating synthetic celebrity faces, fake Pokémon characters, time-series medical events [7] and electronic medical records [8].

Due to the impressive realism in synthetic data generated by GANs, they have also initiated an active and important discussion of malicious use of generative models, and privacy implications. We will discuss this important aspect of generative models in Section 6.

One difficulty with GANs is that they are quite difficult to train, and often require significant time and effort to manually tune until they reach the desired outcome; some of the most common issues when training GANs are:

- **Nash Equilibrium:** the Generator and Discriminator work against each other in a competitive manner, and it is often rather difficult to reach the Nash equilibrium of this 2-player minimax game. Training GANs to achieve this equilibrium tends to require extensive experimentation and good intuition about how GANs work.

- **Vanishing gradient:** when the Discriminator is doing very well in its role to discriminate between real and synthetic data, its gradients are very close to 0 and thus learning in the Generator slows down significantly or sometimes even stops completely.

- **Mode collapse:** a common failure mode in GANs where the Generator generates samples that "fool" the Discriminator but fails to generate the full breadth of such possible samples and thus gets stuck in a local sub-space of synthetic samples possible. For example, consider an image face generator that generates excellent photorealistic images of faces but only focuses on faces of people with gray hair. Since the images are of great quality, the discriminator will consider them of great quality and indistinguishable from real images, however they only represent a fraction of the types of images in the training set, which include many more hair colors.

Various approaches and hacks have been proposed to address the vulnerabilities in GANs with varying levels of success. One important improvement over the basic GAN approach is Wasserstein GAN (WGAN [9]) which uses a different loss function based on Wasserstein distance, and has been shown to be more robust to mode collapse.

5. Industry example: applications of generative models to healthcare data

Healthcare is one of the most popular area of application for analytics and machine learning, driving improved outcomes for patients, lower cost of care, and improved patient experience. There are a vast number of applications for data in healthcare, such as measuring quality of care metrics, developing predictive models for better diagnosis, or analyzing data to understand the differences in clinical care protocols.

Due to the highly regulated nature of healthcare data, and various regulations that govern health data privacy (such as HIPAA, GDPR, CCPA), most healthcare data are locked down in silos. Many healthcare organizations have used de-identification as a way to reduce privacy risks, typically through the modification of potentially identifiable attributes (e.g., dates of birth) via generalization, suppression or randomization. However, this approach is susceptible to linkage attacks, as was demonstrated in [10], and it is accepted by many risk experts that the risk of re-identification is high and in fact they treat de-identified medical data the same way they do fully identifiable medical data.

This presents an enormous challenge to realizing the promise of understanding and using data in healthcare to drive better outcomes and achieving the vision of precision medicine.

There are many types of medical data that is useful, and herein we focus on three types of data that are quite common:

- **Tabular data:** large amounts of medical data are collected in table format, including clinical trial data and other data used for observational studies. In clinical trials, for example, the researchers review the individual patient records from the trial, and perform statistical analysis to understand whether the hypothesized outcome of the trial is confirmed or rejected with statistical significance given the data. Being able to share the vast amount of clinical trial data that is currently locked down in medical centers and biopharma companies to the research community, as well as combining these datasets, can unlock advances in design and speed-to-market for many necessary drugs and medical devices.

- **Electronic Medical Records:** electronic medical records (EMR) are now mandated by regulatory bodies; a vast number of such records is collected every day around the world, and stored in EMR systems by vendors like EPIC, Cerner and Allscripts. EMR are difficult to access due to privacy regulations, yet they represent a gold-mine of aggregated knowledge about health outcomes and can open up enormous opportunities for precision medicine.

- **Medical imaging**: medical imaging diagnostics using MRI, CT and other types of scanning are critical in diagnosis and following the response of treatment, and where advanced AI and machine learning are poised to provide significant gains in the near future (see e.g., [11]). Yet many diagnostics providers are

starving for highly quality and diverse labeled medical images to improve their diagnostics models, leaving a huge gap in advancing the state of the art.

By providing synthetic EMR, clinical trial or medical imaging data that accurately mimics the statistical properties of the real data, one can perform the same analysis or modeling on the synthetic data, achieving near- identical results, without the risk of exposing patient privacy. Even more exciting is the ability to augment small medical datasets with synthetic data, which is useful for example in the case of relatively rare medical conditions where the number of patients available is limited.

It's interesting to note that there is previous work on synthetic data generation in the healthcare domain, notably the work done on Synthea described in [12]. These early techniques, while recognizing the importance of high fidelity synthetic data, used domain-specific knowledge to drive simulated data, but have unfortunately failed to achieve the kind of fidelity that is required for any meaningful analytics (see [13]), and thus have proven to be of limited use in practice where patient-level analysis is required.

More recently, generative adversarial networks and variational auto-encoders have been applied to medical datasets, which have demonstrated the potential to provide much higher fidelity synthetic data and thus more useful in practice. We now quickly review two of these more recent techniques: generating medical records with discrete values (MedGAN), and work by Nvidia to generate synthetic medical imaging.

5.1 MedGAN: generating discrete medical variables with GANs

Electronic medical records include vast amounts of structured data about patients such as diagnoses, drugs, lab results, and procedures. Most of this data is encoded in commonly shared data dictionaries such as ICD9 or ICD10 for diagnosis codes, NDC for drug codes, and similar dictionaries for procedure codes and labs. Although some variables in this data are continuous (like lab results), most of it is represented as discrete variables with very large dictionary sizes.

MedGAN [8] was developed with the recognition of the potential that generative adversarial networks have to model electronic medical records, while trying to adapt the GAN approach to deal with discrete variables, which it's not typically very good at. MedGAN aims to learn the probability distribution of data that include high-dimensional, multi-label discrete variables, and specifically supporting both binary (e.g., variables that represent whether you have a certain diagnosis or not), and count variables (i.e., variables that represent how many times a patient took a medication over time, or total number of risk factors for some disease). This approach proposes combining an auto-encoder within a generative adversarial network architecture and demonstrates how to deal with situations of overfitting and mode collapse in this scenario.

It is noteworthy that in addition to MedGAN, several researchers proposed additional similar approaches to modeling medical records and other tabular data, for example EhrGAN proposed in [14] and TableGan proposed in [15].

5.2 Medical image synthesis with GANs

It is widely recognized in AI and machine learning that insufficient data volume as well as imbalanced or non-diverse data often leads to poor predictive performance and lack of model generalization. This often proves to be a critical issue in the development of medical imaging algorithms where abnormal findings are by definition rare, and high-quality training images are hard to find.

In [16], Nvidia researchers demonstrate generation of synthetic MRI images with brain tumors using generative adversarial networks, trained on two publicly available datasets of brain MRI: ADNI and BRATS. Two distinct benefits of synthetic data are highlighted in this work: improved performance leveraging synthetic images as a form of data augmentation, and the value of synthetic data as a tool for reducing privacy risk while achieving comparable tumor segmentation results when trained on the synthetic data versus when trained on real data.

The results from [16] are quite impressive, and some synthetic images taken from that paper are shown in **Figure 6**.

Clearly more work remains in this area, especially in generating higher resolution synthetic images, tackling all imaging modalities as well as addressing many other clinical use-cases; nonetheless, this work demonstrates excellent initial results for synthetic image generation in medical research with the potential to improve medical imaging diagnostics and significantly reduce privacy risks.

5.3 Other approaches

Recently, neural language models with attention (i.e., Transformers [17]) have been used to for a variety of language tasks, including synthetic text generation, sequence to sequence translation, question answering and many others. One potential application of language models in medicine is the generation of free-text clinical notes based on structured data. Instead of generating synthetic versions of the structured medical EMR record, the goal is to translate the input structured data into a clinically correct and useful text summary of the patient information, in a form physicians are used to reading. Although early experiments with human-like language generation with models like GPT2 are showing good initial results, there's still a lot of work to do in this area.

It's worth mentioning one other generative modeling approach called flow-based generative models; this technique is quite complex mathematically, and is in early stages of research, but can potentially provide an additional set of

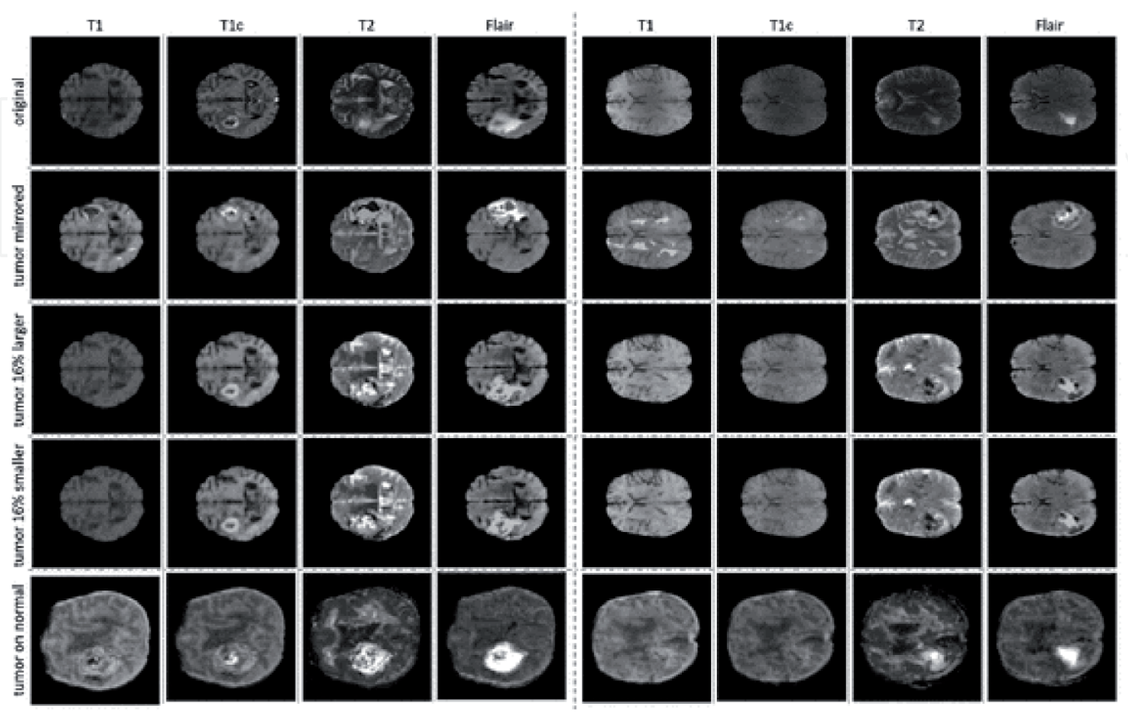

Figure 6.
Examples of synthetic abnormal brain MRI images.

methods for synthetic data generation. The interested reader is referred to [18, 19] for more details.

Another recent area of research in deep learning and privacy aims to integrate differential privacy into training procedures of deep neural networks [20]. This is particularly important for generative models and can be used to constrain the learning process around certain privacy guarantees, ensuring that the learning process does not just memorize the input data.

6. Privacy of synthetic data

With differential privacy, our goal is to define a query mechanism that guarantees certain privacy levels if the users are restricted to access micro-data through the specified mechanism only. Synthetic data generation is different in that it assumes synthetic data is published directly to users, and thus access to the data is virtually unlimited. We now want to inspect those differences in more detail to better understand the implications of privacy for synthetic data generation.

We start with an important, fundamental recognition. With real datasets (either de-identified or available through differential privacy mechanisms), an attacker knows for sure that each row in the datasets represents a real instance or person, only the privacy mechanisms attempt to conceal the privacy information in different ways. With synthetic datasets that is not the case, as the samples are randomly chosen from a probability distribution, and thus by definition do not reflect real people. In fact, as described at the beginning of this chapter, if we assume the real data and synthetic data are both sampled from a theoretical (unknown) distribution P(X), and that distribution is very high dimensional (as it often is for micro-data), then the only hypothetical risk is that by a stroke of luck a synthetic record exactly matches the values in one of the original values, which is very unlikely. And its occurrence could not be recognized with any assurance by an attacker.

Nonetheless, there is an important privacy consideration – unintended memorization [21]. A deep generative learning model might unintentionally memorize the training set (of real data) and thus instead of approximating a distribution and then sampling from that distribution, it instead just copies one or more of the original data records into the synthetic dataset.

It is possible to test for memorization pro-actively as part of training the generative model (as proposed in [21]) and optimize the generative model in such a way as to remove any memorization or minimize it to a level which presents minimal risk.

To further enhance privacy guarantees, we can apply a k-anonymity [22] to the synthetic dataset. It's common to use generalization or obfuscation of variables to achieve the desired levels of k-anonymity; however both techniques result in reduced utility. With synthetic data, however, one can instead generate additional records in a way that improves the privacy guarantees without compromising utility.

7. Summary and conclusion

In this chapter we provided an overview of synthetic data and how it may provide an alternative to differential privacy as a method for sharing micro-data for the purpose of analysis and machine learning applications.

We discussed two of the most common techniques used in deep generative modeling, namely variational auto-encoders and generative adversarial networks, and highlighted some of the remarkable success in the space of modeling medical

data. We then discussed why synthetic data provides privacy by design and some areas of research in privacy of synthetic data generation.

As research in the space of generative models continues at a neck-break pace at companies like OpenAI, Google, Facebook, Microsoft and others, we expect to see tremendous prosgress in this field on the research side as well as in applications of synthetic data across many areas of industry.

Acknowledgements

We would like to thank Dr. Johannes Otterbach and Dr. Marlene Grenon for many hours of discussions on the topic of synthetic data generation.

Conflict of interest

The authors are co-founders of Syntegra.io, a startup with a mission to enable completely secure data sharing of even the most sensitive medical information in a way that fully maintains statistical fidelity while preserving privacy, via its synthetic data engine based on generative models.

Author details

Ofer Mendelevitch[1*] and Michael D. Lesh[2]

1 Syntegra.io, San Carlos, USA

2 Syntegra.io, Mill Valley, USA

*Address all correspondence to: ofermend@gmail.com

References

[1] Garfinkel S, Abowd J, Powazek S. Issues encountered deploying differential privacy. In: WPES'18: Proceedings of the 2018 Workshop on Privacy in the Electronic Society; 2018. pp. 133-137. DOI: 10.1145/3267323.3268949

[2] Kingma D, Welling M. Auto-encoding variational bayes. In: ICLR; 2014. arXiv:1312.6114

[3] Rezende D, Mohamed S, Wierstra D. Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31st International Conference on Machine Learning (ICML); 2014. arXiv:1401.4082v3

[4] Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, et al. β-VAE: Learning basic visual concepts with a constrained variational framework. International Conference on Learning Representations. 2017;**2**(5):6

[5] Van den Oord A, Vinyals O, Kavukcuoglu K. Neural Discrete Representation Learning. In: NIPS; 2017. arXiv:1711.00937v2

[6] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Advances in Neural Information Processing Systems. 2014. pp. 2672-2680. arXiv:1406.2661v1

[7] Yu L, Zhang W, Wang J, Yu Y. SeqGAN: Sequence generative adversarial nets with policy gradient. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017. San Francisco, California: AAAI Press; 2017. pp. 2852-2858. arXiv:1609.05473v6

[8] Choi E, Biswal S, Malin B, Duke J, Stewart W, Sun J. Generating multi-label discrete patient records using generative adversarial networks. In: Machine Learning for Healthcare Conference. PMLR; 2017. arXiv:1703.06490v3

[9] Arjovsky M, Chintala S, Bottou L. Wasserstein GAN; 2017. arXiv:1701.07875v3

[10] Barth-Jones D. The "Re-identification" of governor William Weld's medical information. In: A Critical Re-Examination of Health Data Identification Risks and Privacy Protections, Then and Now. 2012. DOI: 10.2139/ssrn.2076397

[11] Pesapane F, Codari M, Sardanelli F. Artificial intelligence in medical imaging: Threat or opportunity? In: Radiologists Again at the Forefront of Innovation in Medicine. 2018. DOI: 10.1186/s41747-018-0061-6

[12] Walonski J, Kramer M, Nichols J, Quina A, Moesel C, Hall D, et al. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. Journal of the American Medical Informatics Association. 2018;**25**(3):230-238. DOI: 10.1093/jamia/ocx079

[13] Chen J, Chun D, Patel M, Chiang E, James J. The validity of synthetic clinical data: A validation study of a leading synthetic data generator (Synthea) using clinical quality measures. BMC Medical Informatics and Decision Making. 2019;**19**(1). DOI: 10.1186/s12911-019-0793-0

[14] Che Z, Cheng Y, Zhai S, Sun Z, Liu Y. Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In: International Conference on Data Mining. IEEE; 2017. arXiv:1709.01648v1

[15] Park N, Mohammadi M, Gorde K, Jajodia S, Park H, Kim Y. Data synthesis

based on generative adversarial networks. In: International Conference on Very Large Data Bases. 2018. arXiv:1806.03384v5

[16] Shin H, Tenenholtz N, Rogers J, Schwartz C, Senjem M, Gunter J, et al. Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In: Workshop on Simulation and Synthesis in Medical Imaging - SASHIMI2018. 2018. arXiv:1807.10225v2

[17] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A, et al. Attention is all you need. In: NIPS17: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017. pp. 6000-6010. arXiv:1706.03762v5

[18] Dinh L, Sohl-Dickstein J, Bengio S. Density estimation using real NVP. In: ICLR. 2017. arXiv:1605.08803v3

[19] Kingma D, Dhariwal P. Glow: Generative flow with invertible 1x1 convolutions. In: Advances in Neural Information Processing Systems. 2018. pp. 10215-10224. arXiv:1807.03039v2

[20] Abadi M, Chu A, Goodfellow I, McMahan B, Mironov I, et al. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM; 2016. pp. 308-318. arXiv:1607.00133v2

[21] Carlini N, Liu C, Erlingsson U, Kos J, Song D. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks and Extracting Secrets; 2018. arXiv:1802.08232v3

[22] Holohan N, Antonatos S, Braghin S, Aonghusa P. (k, ε)-Anonymity: k-Anonymity with ε-Differential Privacy; 2017. arXiv:1710.01615v1