Data Interpolation through Deep Generative Models

Juan Ramirez[1]

Advisors:
Jose Gallego[2]
Maria Eugenia Puerta [3]

Research practice II
Final report
Mathematical Engineering
Department of Mathematical Sciences
School of Sciences
Universidad EAFIT

November 2020

[1]jurami28@eafit.edu.co. Universidad EAFIT. CvLAC
[2]jgalle29gmail.com. Université de Montréal/MILA. https://jgalle29.github.io
[3]mpuerta@eafit.edu.co. Universidad EAFIT.

Abstract

Interpolation on the latent space of deep generative models is an efficient tool for data generation. This can be performed well under the Riemannian geometry induced by the model generator's jacobian. Nonetheless, if the jacobian is ill-conditioned, solvers for a geodesic interpolating curve may fail to converge to a good solution. We evaluate the degree of this ill-conditioning across different deep generative models and their architectures. It is shown that variational autoencoders produce a well-conditioned jacobian provided there is enough training data. Furthermore, it is clear the generator is not well defined away from training data. Hence, some intervention which adjusts its jacobian is required for appropriate geodesic interpolation.

Keywords: condition number, deep generative models, geodesics, data interpolation.

1 Introduction

Data representation is a fundamental aspect of machine learning. The performance of many techniques is greatly dependent on the structure of the data presented to them. Also, when dealing with data of high dimensionality, additional challenges arise. Recently, deep generative models have been used to study the representation of data. These methods, while capable of performing dimensionality reduction, enable a suit of other possibilities. In short, they produce a low dimensional (latent) representation Z, while modelling a generator function which can map samples to the (ambient) data space X. Computations, visualisation, sampling or otherwise can then be performed efficiently in the latent space. The main benefit of having the generator is that data can be generated. Goodfellow *et al.* (2016) provide a comprehensive overview of the field.

A natural way for generating unseen data through the latent space is by means of linear interpolation (Verma *et al.*, 2019). That is, a convex linear combination of the latent representation of a pair of data samples, which is then mapped to the X space. It is argued that this approach produces more meaningful data and is less arbitrary than random sampling. Regardless, it may induce some issues. In short, it assumes that Euclidean distances on the latent space are representative of distances on the ambient space. In contrast, Tosi *et al.* (2014); Arvanitidis *et al.* (2017) argue that the manifold generated by Deep Generative Models induces a Riemannian geometry on the latent space. It is shown that distances along the manifold are not coherent with Euclidean distances on the Latent space. More precisely, a metric tensor on the latent space is deduced according to the Jacobian of the deep generative model's generator. This indicates how changes in the latent space are reflected on the ambient space. Subsequently, shortest curves between points, called geodesics, are formulated.

1.1 Problem Definition

Solving for a geodesic is difficult. Naturally, an estimation of g will greatly depend on the estimated jacobian. Unfortunately, a closed, analytical representation of J is unattainable in the context of deep generative models (Tosi *et al.*, 2014). Hence, the geodesic is usually estimated based on a discrete set of values of the jacobian, which are deduced via numeric differentiation of the generator network. Despite this, the Jacobian is said to be numerically unstable (Arvanitidis *et al.*, 2019). This means it can not be reliably used for calculations. This is a significant problem as most methods for approximating geodesics either use the jacobian when solving for them (Arvanitidis *et al.*, 2017; Chen *et al.*, 2019), or optimise their parametrization through an objective function defined via the jacobian (Yang *et al.*, 2018; Chen *et al.*, 2018). An ill-conditioned jacobian may cause convergence issues and may also produce sub-optimal solutions. Furthermore, it is unclear under which conditions the jacobian may or may not be stable.

In this paper, we offer a deeper look into the implications of having an ill-condition jacobian when estimating shortest paths. We experimentally evaluate how the jacobian behaves under the framework of different deep generative models; namely, variational autoencoders, denoising autoencoders and generative adversarial networks. Also, we revise whether certain design choices on the generator network may mitigate these adverse effects. Furthermore, we argue that although geodesics are difficult to solve, the jacobian of variational autoencoders is not ill-conditioned. Nonetheless, it is poorly defined away from data and so adjusting it is recommended for appropriate geodesic interpolation. We also reiterate the fact that considering the latent space of deep generative models as euclidean is misleading. Then, even beyond the use cases of geodesic interpolation, the distance between latent representations must be calculated based on the generator's induced geometry.

1.2 Justification

Interpolation is useful for various reasons. When generating data, it is usually desirable to produce samples similar to some reference data point. Also, it may reveal interesting relationships between the underlying classes of the data. Furthermore, extra data can be used to augment a data set. This is useful for regularising and denoising machine learning algorithms. Regardless, linear interpolation is not precise. The dimension of Z is lower than that of X and so the mapping between the ambient and latent spaces is not isometric. Also, interpolation may traverse areas of uncertainty in the Z space where little training data was used. Therefore a suitable geometry to analyse the latent space is relevant, as well as interpolations according to said geometry.

Under a smooth generator, the deduced metric tensor is well defined and geodesics can be computed. Nonetheless, an ill-conditioned jacobian means solutions to geodesics will be numerically difficult. This entails interpolations which are far more complex to deduce than linear interpolations and that may not be significantly better. Hence, it is desirable to know up to what degree geodesics are worth solving under given methodologies and under which conditions their computation may be easier. This may also indicate where future research should focus. In particular, whether new approaches to solve for geodesics should intervene the jacobian, propose a new deep generative model with a well conditioned jacobian, or solve for geodesics whilst avoiding the jacobian altogether.

1.3 Outline

Section 2 mentions some of the methodologies used for estimating geodesics on the latent space of deep generative models. We also reference literary work which either intervenes or avoids the jacobian in favour of alternative solutions. Section 3 provides a brief overview of various frameworks relevant to our work. In particular, section 3.1 describes variational autoencoders, denoising autoencoders and generative adversarial networks; section 3.2 introduces the concepts of a metric tensor and a geodesic; section 3.3 describes notions to assess the numerical stability of the generator's jacobian. Section 4 comprises comparisons of the degree of ill-conditioning of the jacobian across different deep generative models. Finally, section 5 gathers conclusions, as well as our take on how future research should be focused.

2 State of the art

Arvanitidis *et al.* (2017) deduce a boundary value problem for computing geodesics in the latent space. Their approach is computationally taxing, and an ill-conditioned jacobian can hinder a solver's convergence (Arvanitidis *et al.*, 2019). Yang *et al.* (2018) expand on this idea and parametrize geodesics as quadratic splines. They optimise the spline's parameters through gradient descent on the geodesic's length. Both of the aforementioned approaches adjust the metric tensor to prevent geodesics from traversing areas where the generative model has high uncertainty; that is, where little training data fell. This ensures a well behaved geometry away from data. Chen *et al.* (2018) approximate geodesics as neural networks. Optimisation is performed on the length of the geodesic, with an added regularisation term. This term encourages the geodesic to follow the data manifold, similar to the aforementioned adjustment. Chen *et al.* (2019) utilise data to build a graph where every data point is connected to its k nearest neighbours. The notion of distance is measured on a straight line connecting neighbours, although taking into account the jacobian. Then, geodesics consist on the piece-wise linear shortest paths through the graph. They argue this approach is simple and scales well with the dimensionality of the latent space.

Regardless, Arvanitidis *et al.* (2019) argue that the metric tensor based on a variational autoecoder's generator is ill-conditioned. In particular, they say this is due to the fact that deep generative models are trained on limited data. Hence, the generator function can change irregularly fast in between data points. Nonetheless, Arvanitidis *et al.* (2019) model geodesics as the posterior mean of a gaussian process, which is updated via a fixed point algorithm. This procedure does not explicitly use the jacobian, and so it bypasses its associated issues. On a similar note, Chen *et al.* (2020) evade the problem of estimating geodesics, but rather propose a variation of VAEs which produces an approximately Euclidean latent space. They regularise the metric tensor during VAE training in order to obtain a flat manifold. The issues with the jacobian are reflected in the fact that, albeit initial solutions were based on it, later solutions evade it altogether.

3 Methodology

3.1 Deep Generative Models

AutoEncoders have been used as a basis for generative modelling. They consist on a pair of funcitons which are trained to be inverses. An encoder maps data points to a latent space, while a decoder reconstructs data from their latent representation. More precisely, let $X \subseteq \mathbb{R}^N$ and $Z \subseteq \mathbb{R}^n$ be Euclidean spaces. For some submanifold $\mathcal{M}$ embedded in X, a pair of functions $e : X \longrightarrow Z$ and $f : Z \longrightarrow X$ that satisfy $f \circ e|_{\mathcal{M}} \approx id_{\mathcal{M}}$ are referred to as an AutoEncoder on $\mathcal{M}$. These are usually modelled as neural networks optimised to minimise the reconstruction error of some dataset. When constrained to reduce the dimensionality of data ($n < N$), AutoEncoders are forced to develop a manifold which captures the data's intrinsic characteristics. From a machine learning point of view, we can interpret this embedded manifold as the underlying support of the data distribution. This support, in practice, does not require the full dimensionality of the ambient space.

3.1.1 Denoising AutoEncoders

Consider an encoder $e : X \to Z$, decoder $f : Z \to X$ pair. Similar to traditional autoencoders, these functions are designed as to $f \circ e|_{\mathcal{M}} \approx id_{\mathcal{M}}$. Nonetheless, when using more hidden and latent dimensions than those of the input space, regular autoencoders may learn an identity function (Goodfellow *et al.*, 2016). Therefore, they may fail to extract relevant characteristics from data. This can also be the case when the encoder and decoder functions are given too much capacity. Denoising autoencoders propose an stochastic extension to autoencoders. Let L be a function to quantify the difference between data samples. Also, let $C(\tilde{x} \mid x)$ be a corruption process where $E_{C(\tilde{x}|x)}[\tilde{x}|x] = x$. Then, f, e are optimised based on $L(x, (f \circ e)\tilde{x})$, where $\tilde{x}$ results from a random variation of x. In other words, the network is asked to reconstruct data from some noisy variation of itself. When measuring the mean squared error ($L = ||(f \circ e)(\tilde{x}) - x||^2$), a reconstruction is an estimation of $E_{p(x)C(\tilde{x}|x)}[x|\tilde{x}]$, where $p(x)$ is the data distribution on X (Goodfellow *et al.*, 2016). Therefore, the DAE is indeed designed to denoise its inputs.

An advantage of DAEs in the context of our work, is that they induce a locally smooth metric tensor. This entails a geometry which is, at least locally and around training data, well behaved. This is due to the added noise when training. In fact, noisy inputs to the encoder will produce different latent representations. Nonetheless, these are all required to be reconstructed as the same sample. Therefore, the reconstruction function is insensitive to perturbations of the input and so it's jacobian varies smoothly around data points. Regardless, DAEs do not impose conditions on the global structure of data. This means there can be some areas of the latent space which are more densely packed with data than others. This, in turn, entails a tensor which changes too rapidly in some areas but no so in others. Also, the latent space may have gaps. These gaps will have a poorly estimated generator and jacobian. These factors make sampling and interpolation difficult in the context of DAEs.

3.1.2 Variational AutoEncoders

Autoencoders can produce widely varying and complex latent spaces. This, in turn, may render data generation not very straight forward and data interpolation difficult. In response, Variational AutoEncoders were introduced (Kingma & Welling, 2013). These models regularise the distribution of points on the latent space, as to induce some desired properties. VAEs approach the autoencoder in a probabilistic manner. They model the distribution of data in the ambient space under a Bayesian framework: $p(x) = \int p(x|z;\theta)p(z)dz$, while assuming a $\mathcal{N}(0, \mathbf{I}_n)$ prior distribution on the latent space. In addition, the likelihood $p(x|z;\theta)$ of generated data from some $z \in Z$ is considered to be as a $\mathcal{N}(\mu(z), \sigma(z)\mathbf{I}_N)$, where $\mu(z) : Z \longrightarrow X$ and $\sigma(z) : Z \longrightarrow X$ are trained as Neural Networks which model, respectively, the mean and variance of the values that are generated by some z. In other words, each point in Z produces an entire distribution of points on X.

The training of the encoder e and mean and variance networks μ, σ considers both the reconstruction error of samples as well as the distribution of data on the latent space. Note that a reconstruction for some $x \in X$ is modelled as $\mu(e(x)) + \sigma(e(x)) \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I}_N)$ and $\odot$ is an element-wise product. Furthermore, as the posterior distribution of data on Z is intractable, its difference with respect to a standard normal is measured via a Kullback-Leibler divergence. Hence, the autoencoder is motivated to reconstruct images properly, but also to distribute latent representations according to a normal distribution. Kingma & Welling (2019) present a comprehensive overview of Variational Autoencoders and mention some of its extensions.

This model produces a well behaved geometry on the latent space. Furthermore, sampling is straightforward. Data is densely packed and gaps on the latent space are not commonplace. Similar to DAEs, the randomness associated with a reconstruction favours a locally smooth geometry. Finally, the modelling of variance produces a stochastic Riemannian metric which can adequately model the metric tensor away from the manifold (Hauberg, 2018). These reasons explain why geodesic interpolation has mainly been applied to VAEs.

3.1.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs) differ from the aforementioned methodologies as they do not model an encoder function. Furthermore, a downstream task is designed such that optimising the generator with respect to this task enforces the generated distribution to be close to the true data distribution. This procedure avoids the difficult task of comparing both distributions directly. More precisely, the generator function is asked to compete with a discriminator in an zero-sum adversarial game. The former produces data, while the latter tries to distinguish generated data from real data. Both networks are trained in tangle: the generator produces samples, which are provided to the discriminator alongside real data to be classified as real or fake. Then, the discriminator is updated in order to better discern between true and false data, while the generator is adjusted to fool the discriminator further. This design motivates the generator to accurately reproduce the distribution of data on the X space (Goodfellow *et al.*, 2014).

Let $f : Z \rightarrow X$ be a generator function from a d dimensional latent space to the ambient space. Also, let $d : X \rightarrow [0, 1]$ be a discriminator. Furthermore, allow z to be a batch of random elements of Z distributed as a $p_z \sim \mathcal{N}(0, \mathbf{I}_n)$. Then, given the true probability distribution over X, p_t, the expected absolute error of the discriminator is $\text{Er}(d, f) = E_{x\sim p_t}[1 - d(x)] + E_{z\sim p_z}[d(f(z))]$. When training the generator, it is desired to maximise this error while the discriminator attempts to minimise it. That is, some $f^* = \text{argmax}_f(\min_d \text{Er}(d, f))$. It can be shown that such an f^* is achieved when the data distribution it generates is equivalent to the true data distribution.

Similar to VAEs, GANs pack data compactly around the origin of the latent space. Furthermore, sampling is straightforward. In addition, when dealing with interpolations, the discriminator is useful to assess the quality of produced samples. Nevertheless, no encoder function is modelled and so interpolation between actual data samples is difficult. Also, as this deep generative model has no randomness associated with it, the smoothness of the tensor can not be guaranteed theoretically. Although GANs are very popular for data generation and interpolation, they have not been widely considered for generator-induced Riemannian interpolation.

3.2 Differential Geometry

Let $\mathcal{M}$ be a manifold embedded in X. Also, let $f : Z \longrightarrow \mathcal{M}$ be a generator function. On the vicinity of every point $z \in Z$, a metric space can be defined. Let $T_z\mathcal{M}$ be the tangent space to z in $\mathcal{M}$. This space is equipped with a dot product $\langle\cdot,\cdot\rangle_z : T_z\mathcal{M} \times T_z\mathcal{M} \longrightarrow \mathbb{R}$. Also, consider $\mathrm{J}_z = \partial f/\partial z$ the Jacobian of the generator function with respect to the latent space. This Jacobian comprises the curvature of the manifold $\mathcal{M}$; that is, it indicates how movements in the Z space are reflected in the X space along the manifold $\mathcal{M}$. Hence, the dot product in $T_z\mathcal{M}$ can be formulated as a dot product in Z which varies according to the metric tensor $M_z = \mathrm{J}_z^T\mathrm{J}_z$.

$$\langle a, b\rangle_z = \langle \mathrm{J}_z a, \mathrm{J}_z b\rangle = a^T \mathrm{J}_z^T \mathrm{J}_z b$$

Let $g : [0, 1] \longrightarrow Z$ be a curve in the latent space. According to the tensor induced by f, its length is $\int_0^1 \sqrt{\langle g'(t), g'(t)\rangle_{g(t)}}dt = \int_0^1 \sqrt{g'(t)J^T Jg'(t)}dt$. Given a pair of interpolation points z_0 and z_1, we are interested in some g^* such that:

$$g^* = \underset{g}{\operatorname{argmin}} \int_0^1 \sqrt{g'(t)\mathrm{J}^T\mathrm{J}g'(t)}dt$$
$$\text{subject to } g(0) = z_0,\ g(1) = z_1$$

This is called a geodesic under the metric tensor M_z. These functions exist given the tensor is smooth. Regardless, they may not be unique. Furthermore, note that geodesics minimise the curve's length locally. This means that they may be different from the shortest curves between two points. In other words, when solving for a geodesic, an algorithm can get stuck in a sub-optimal curve which locally optimises curve length, but which is not a globally length-minimising curve.

3.2.1 Stochastic Riemannian Manifolds

Consider a variational autoencoder. Its generator function can be represented as $f(z) = \mu(z) + \sigma(z) \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I}_N)$ and $\odot$ is an element-wise product. As this generator is stochastic, its jacobian is also stochastic. Nonetheless, the metric tensor is usually modelled as its expected value under f. That is,

$$E\left[\left(\frac{\partial f}{\partial z}\right)^T\left(\frac{\partial f}{\partial z}\right)\right] = \frac{1}{N}\left(\frac{\partial \mu}{\partial z}\right)^T\left(\frac{\partial \mu}{\partial z}\right) + \frac{1}{N}\left(\frac{\partial \sigma}{\partial z}\right)^T\left(\frac{\partial \sigma}{\partial z}\right)$$

Which differs from deterministic metric tensors as it includes a term which models uncertainty. Hauberg (2018) argues that deterministic generators like those present in regular autoencoders and GANs fail to adequately model the geometry of tha data. This is due to the fact that the generator is not well defined outside of the data manifold. Hence, the energy required to traverse these areas may be low enough so that some geodesics leave the manifold. Therefore, he argues that only a stochastic generator with a variance term which is amplified away from data can be adequately represented with a Riemannian manifold.

Nonetheless, they note that variance estimates of VAEs are poorly defined away from data. Arvanitidis *et al.* (2017) had solved this problem by modelling the inverse variance separately via an RBF function which vanishes away from data. Under the notion that euclidean distances are not relevant in the latent space, this approximation to variance is poor. In fact, under a fixed bandwidth this completely disregards the purpose of the geometry induced by the generator. This may also produce very high variance areas in an arbitrary way, which may be reflected as a cut through the manifold. Hauberg (2018) agrees this solution is not satisfactory enough. Because of this, we argue none of these deep generative models quantify variance adequately. As such, we deem autoencoders and GANs to be reasonable alternative to VAEs for latent space interpolation. The current interventions which are made on the VAEs variance can be similarly applied on deterministic generators. Hence, they were all considered in this paper.

3.3 Jacobians

3.3.1 Magnitude of the Metric Tensor

The length of a curve g on the latent space with respect to the geometry induced by the generator function is:

$$\int_0^1 \sqrt{g'(t)\mathbf{J}^T\mathbf{J}g'(t)}dt = \int_0^1 ||\mathbf{J}g'(t)||_2 dt$$

Hence, the size of a curve can be measured locally by means of the jacobian. In particular, $\mathbf{J}^T\mathbf{J}$ may amplify or contract the length of the curve locally. Therefore, the determinant of the metric tensor, $\det(\mathbf{J}^T\mathbf{J})$ gives us a notion of how much energy a curve must spend to traverse certain areas of space. Regions with high values for this determinant should be avoided by geodesics in favour of others with low values.

3.3.2 Condition Number

The condition number of a function measures how the effects of noise on its input impact its output. In other words, they measure the propagation of errors through a system. When considering matrices, a high condition number implies that applying them on vectors yields an unstable output. Consider a matrix being applied as a linear transformation on an euclidean space, as is the case of the generator's jacobian. A high condition number implies that the length of a geodesic can change drastically with small changes on where it passes through. Hence, a solver may have trouble converging to a solution. In practice, this also implies a non-smooth metric tensor. Given the problem at hand is already difficult, these complications can make it very difficult to solve in practice. Let $A \in M_{m\times m}$ be a non singular matrix. Its condition number is defined as:

$$\kappa(A) = ||A|| \, ||A^{-1}||$$

When $||\cdot||$ is selected to be the usual metric on l_2,

$$\kappa(A) = \frac{|\lambda_{max}|}{|\lambda_{min}|}$$

Where λ_{max} and λ_{min} represent the largest and smallest singular values of matrix A. Indeed, a condition number near 1 implies a transformation which changes almost equally regardless of the direction of change of its input. On the other hand, a high condition number makes the direction of the change really significant. Hence, updates on geodesics can change its length in an unstable manner.

4 Experiments

We propose to visualise the determinant of the metric tensor as well as the condition number of the jacobian. In order to do this, we train a suit of different deep generative models with a 2-dimensional latent space. We use the totality of the MNIST dataset for these experiments. In particular, we train GANs, VAEs and DAEs, all under the same architecture. That is, two hidden layers, and either with 256 and 64 hidden dimensions or 128 and 32. Furthermore, we consider sigmoid and hyperbolic tangent activation functions. This is due to the fact that the smoothness of the generator is required in order to define an appropriate metric tensor. Hence, activation functions like rectifier linear units are disregarded. Additionally, all models are trained with an Adam optimiser, a learning rate of 0,01 and betas of 0,99 and 0,5.

4.1 Variational Autoencoders

Figure 1 presents the latent space of a VAE with 128 and 32 hidden dimensions and a hyperbolic tangent activation function. The background represents the determinant of the metric tensor and the condition number of the jacobian. Evidently, the magnitude of the metric tensor is very large, specially near the origin. This is due to the fact that a compact two-dimensional space must produce vastly different images. Regardless, the metric tensor can be very low far away from data. This is not desirable as it may motivate geodesics to leave the data manifold. Furthermore, the jacobian is well conditioned throughout the latent space.

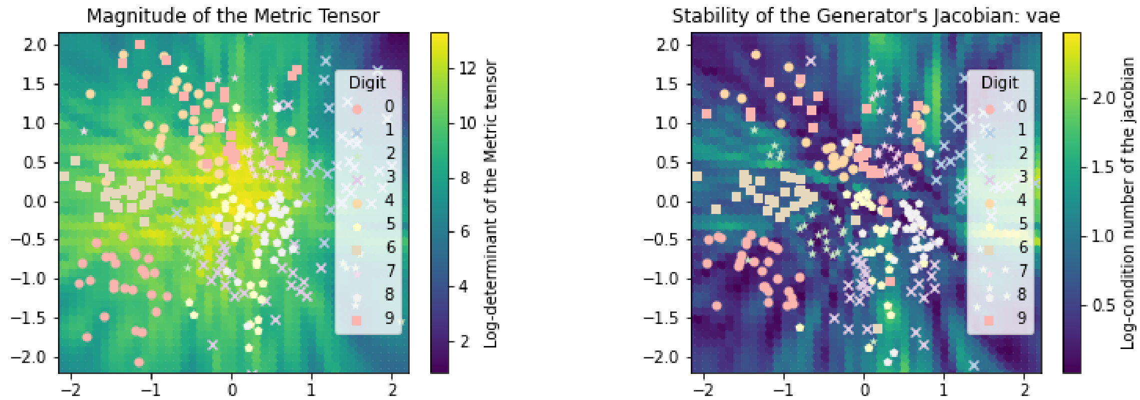

Figure 1: Magnitude of the metric tensor and condition number of the jacobian for a VAE with 128 and 32 hidden dimensions and a Tanh activation.

Figure 2 is analogous to the aforementioned one. The hidden dimensions are modified to 256 and 64, whilst the activation function is maintained. The magnitude of the tensor is slightly smoother throughout the space. This is likely due to the added capability of this architecture. It is evident that this modification does not influence the condition number significantly.

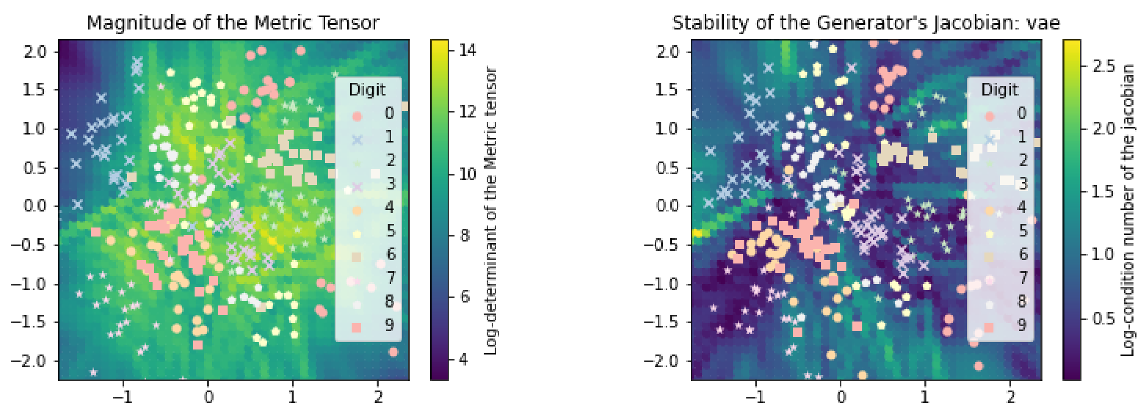

Figure 2: Magnitude of the metric tensor and condition number of the jacobian for a VAE with 256 and 64 hidden dimensions and a Tanh activation.

Figure 3 shows the graphs when considering 128 and 32 hidden dimensions and a sigmoid activation. This configuration yields a slightly smoother tensor and lower condition numbers throughout. Regardless, all configurations yield similar results and a well conditioned jacobian.

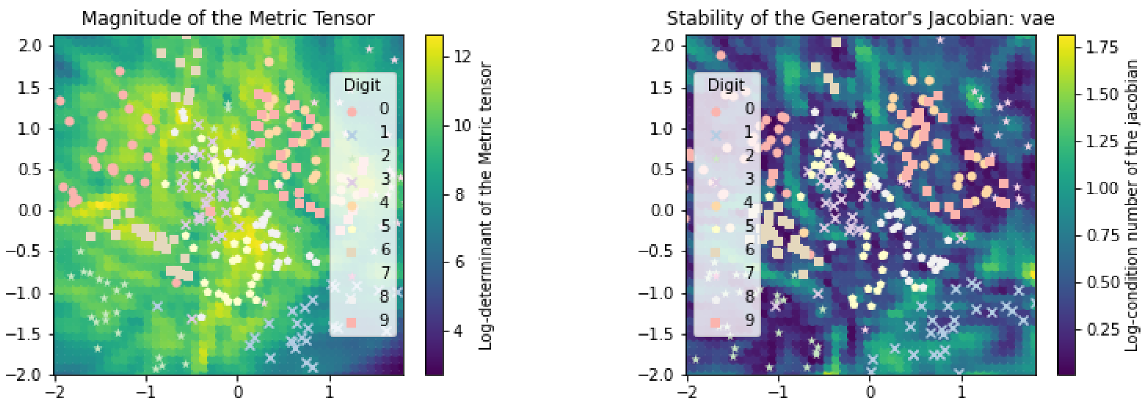

Figure 3: Magnitude of the metric tensor and condition number of the jacobian for a VAE with 128 and 32 hidden dimensions and a Sigmoid activation.

4.2 GANs and DAEs

Different configurations of DAEs behave relatively similar to one another. As evidenced in figure 4, a sigmoid activation function with the 128 and 32 hidden dimension representation produces a metric tensor which is less smooth as opposed to VAEs. In fact, the log-magnitude of the tensor reaches values as high as 16. Furthermore, the conditioning of the jacobian is worse as opposed to VAEs.

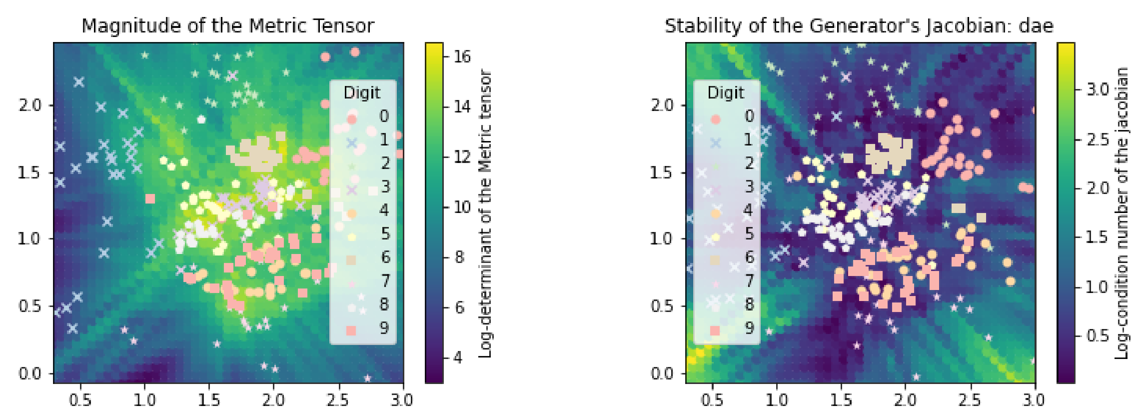

Figure 4: Magnitude of the metric tensor and condition number of the jacobian for a VAE with 128 and 32 hidden dimensions and a Sigmoid activation.

GANs do perform poorly. For simplicity, only one with a Tanh activation and 128 and 32 hidden units is presented. Its conditioning is presented on figure **??**. It can be seen that the magnitude of the tensor spans a larger range as opposed to VAEs and DAEs. Furthermore, the condition number of the jacobian reaches much higher values.

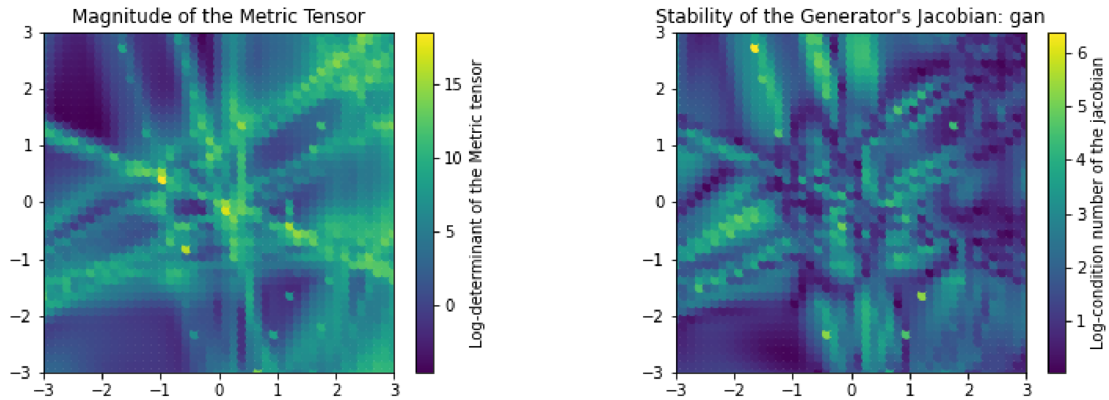

Figure 5: Magnitude of the metric tensor and condition number of the jacobian for a GAN with 128 and 32 hidden dimensions and a Tanh activation.

Amongst Variational Autoencoders, we did not find a configuration with an ill-conditioned jacobian. Hence, this is not considered to be a problem when solving for a geodesic. Nonetheless, the metric tensor is, in general, poorly defined away from data. Some adjustment or correction is still required in order for geodesics to follow the data manifold. Furthermore, DAEs performed slightly worse and possess worse theoretical properties. GANs do present an ill-conditioned jacobian. We do not find that the architecture of the model significantly affects the conditioning of the jacobian on VAEs. Also, although the sigmoid activation function presented a better conditioning, Tanh also performed well.

5 Conclusions

Regardless of whether the computation of geodesics is of interest, the latent space of deep generative models must not be viewed as euclidean. Hence, the measuring of distances on straight lines should be done according to the metric tensor. In addition, the jacobian of the generator of VAEs is generally well conditioned, provided enough training data is available. Furthermore, variational autoencoders are the best suited for geodesic interpolation. This is true disregarding the hidden dimensions and activation functions, provided the latter are smooth. Nonetheless, deterministic generators also produce well behaved geometries near training data.

Various challenges remain. First, there is no straightforward way for bench-marking geodesics and the different proposed methodologies have not been compared systematically. Also, the metric tensor of various models is generally poorly defined away from data. An ideal solution is to modify some deep generative model so that it produces a jacobian with high magnitude away from training data. Alternatively, better procedures to intervene the jacobian after training the model could be introduced. Nonetheless, computing the geodesic whilst avoiding the jacobian is overly complex and comes with little theoretical justification.

References

Arvanitidis, Georgios, Hansen, Lars Kai, & Hauberg, Søren. 2017. Latent space oddity: on the curvature of deep generative models. *arXiv preprint arXiv:1710.11379*.

Arvanitidis, Georgios, Hauberg, Søren, Hennig, Philipp, & Schober, Michael. 2019. Fast and robust shortest paths on manifolds learned from data. *arXiv preprint arXiv:1901.07229*.

Chen, Nutan, Klushyn, Alexej, Kurle, Richard, Jiang, Xueyan, Bayer, Justin, & Smagt, Patrick. 2018. Metrics for deep generative models. *Pages 1540–1550 of: International Conference on Artificial Intelligence and Statistics*.

Chen, Nutan, Ferroni, Francesco, Klushyn, Alexej, Paraschos, Alexandros, Bayer, Justin, & van der Smagt, Patrick. 2019. Fast approximate geodesics for deep generative models. *Pages 554–566 of: International Conference on Artificial Neural Networks*. Springer.

Chen, Nutan, Klushyn, Alexej, Ferroni, Francesco, Bayer, Justin, & van der Smagt, Patrick. 2020. Learning Flat Latent Manifolds with VAEs. *arXiv preprint arXiv:2002.04881*.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, & Bengio, Yoshua. 2014. Generative adversarial nets. *Pages 2672–2680 of: Advances in neural information processing systems*.

Goodfellow, Ian, Bengio, Yoshua, & Courville, Aaron. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Hauberg, Søren. 2018. Only bayes should learn a manifold (on the estimation of differential geometric structure from data). *arXiv preprint arXiv:1806.04994*.

Kingma, Diederik P, & Welling, Max. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kingma, Diederik P, & Welling, Max. 2019. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*.

Tosi, Alessandra, Hauberg, Søren, Vellido, Alfredo, & Lawrence, Neil D. 2014. Metrics for probabilistic geometries. *arXiv preprint arXiv:1411.7432*.

Verma, Vikas, Lamb, Alex, Beckham, Christopher, Najafi, Amir, Mitliagkas, Ioannis, Lopez-Paz, David, & Bengio, Yoshua. 2019. Manifold mixup: Better representations by interpolating hidden states. *Pages 6438–6447 of: International Conference on Machine Learning*.

Yang, Tao, Arvanitidis, Georgios, Fu, Dongmei, Li, Xiaogang, & Hauberg, Søren. 2018. Geodesic clustering in deep generative models. *arXiv preprint arXiv:1809.04747*.