Data Interpolation with Deep Generative Models

Juan Ramirez[1]

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

Research practice II
Research proposal
Mathematical Engineering
Department of Mathematical Sciences
School of Sciences
Universidad EAFIT

August 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. CvLAC

1 Statement of the problem

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, specially for high dimensional spaces. These methods, while capable of performing dimensionality reduction, enable a suit of other possibilities. In short, they map data into a low dimensional (latent) space Z, while explicitly modelling a function which can map samples back to original (ambient) space X. Computations, visualisation, sampling or otherwise can then be performed efficiently in the latent space. An added benefit of having a mapping from the latent to the ambient space is that data can be easily generated. Goodfellow *et al.* (2016) provide a comprehensive overview of the field.

Once a deep generative model has been built, a practical representation of the data is available. 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 real data points. It is argued that this approach should produce more meaningful data than randomly sampling as it is constrained by actual data points, and it also allows for insights to be obtained from the latent space. Regardless, it can induce a plethora of issues. In short, it assumes that Euclidean distances on the latent space are meaningful. 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 in the Latent space. The authors then propose to define the distance between points as the length of a Riemannian geodesic between them.

Solving this curve is difficult as it requires specifying a model of the geodesic, then optimising its parameters as to minimise its length along a curved space, all while satisfying the boundary constraints given by the initial and final interpolation points. Different approaches have been taken to deduce it: solving a system of ordinary differential equations Arvanitidis *et al.*, modelling it as a neural network (Chen *et al.*, 2018) or traversing a nearest-neighbours graph of know data points (Chen *et al.*, 2019). Hence, we are interested in exploring the geometry of deep generative models in order to improve interpolations between data samples. In particular, we desire to propose parametrizations of geodesic curves in the latent space that may be modelled via neural networks.

2 Objectives

2.1 General objective

To propose a parametrization of geodesic for interpolation and distance measuring on the manifolds induced by Deep Generative Models.

2.2 Specific objectives

- To propose a parametrization of geodesics in the latent space of deep generative models, based on neural networks.
- To implement an algorithm which constructs these geodesics.
- To assess the quality of our approach.

3 Justification

Deep generative models have become very popular in the domain of unsupervised learning. Various interesting applications arise from their capabilities to reduce the dimension of data and generate samples. They manage to produce data representations which are useful for many subsequent applications. These models are generally concerned with situations where the data's dimension is very large. Hence, the fact that they perform dimensionality reduction is useful for various reasons. First, many calculations and analysis which may be intractable in the ambient space, can be efficiently performed in the latent space. Additionally, visualisation is made possible when mapping to two or three dimensions. Also, these models can be used for feature engineering, which is an important step in machine learning pipelines. This is as the latent space is designed to intrinsically capture the main characteristics of the data. Furthermore, the adverse effects of the curse of dimensionality when sampling, measuring distances and training statistical models can be mitigated. Regardless, when it comes to dimensionality reduction, whether deep generative models are preferable to other state of the art algorithms like t-SNE, UMAP, or classical approaches like PCA and LDA is disputable. Nonetheless, the generative component is the main reason for the popularity of these methodologies.

Deep generative models allow for a thorough understanding of data and its behaviour. For instance, producing a sample on the latent space, and then mapping it to the ambient space, allows for efficient data generation. Also, the handling of distances on the ambient space is generally intractable and subject to the curse of dimensionality. Because of this, defining an adequate geometry in the latent space allows for accurate distance measuring, which is of paramount importance for procedures like clustering, further supervised learning algorithms and statistical analysis on the data.

Interpolation between data points is useful for various reasons. When generating data, it is usually desirable to produce samples similar to some reference data point. Also, it allows for intuitive sampling and 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 so that they become more robust. Because of this, proper sampling and interpolating methods in the latent space are very useful. It is important for these methods to be precise, and for them to accurately represent a similarity structure within the data. Regardless, linear interpolation in the latent space is not precise. In general, as the dimension of Z is lower than that of X and so the mapping between the ambient and latent spaces is not isometric; distances between samples in X are not coherent with distances between projections in Z. Also, interpolation may traverse areas of uncertainty in the Z space where little training data was used. Finally, interpolations between two objects of the same class may traverse an area where another class is located. Solving for a geodesic is difficult. The optimisation has proven to be slow under some proposed methodologies. Hence, we intend to use neural networks to model the geodesics, which are broad enough to approximate any shape a geodesic may take and which can be adequated to be efficient to train and evaluate.

4 Scope

We expect to work with Variational AutoEncoders, under the Riemannian geometry that is induced in their latent space. We also expect to propose a novel parametrization of geodesics in the latent space, which is based on neural networks. Afterwards, we intend to develop and implement an algorithm for computing said geodesics, benefiting from Automatic Differentiation routines. At last, we expect to assess the quality of the proposed algorithm and contrast it with linear interpolations.

In order to assess the performance of our methodology, the freely available and widely utilised MNIST (LeCun *et al.*, 1998) and Fashion-MNIST (Xiao *et al.*, 2017) datasets will be considered. Artificial data may be used for illustrative purposes. We will use Python 3.8.5 with PyTorch 1.6.0 for Machine Learning related developments.

5 State of the art

AutoEncoders have been used as a basis for generative modelling. They consist on a pair of neural networks 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 $Z \subseteq \mathbb{R}^d$ be Euclidean spaces. For some submanifold $\mathcal{M}$ embedded in X, a pair of functions $e : X \longrightarrow Z$ and $\delta : Z \longrightarrow X$ that satisfy $\delta \circ e|_{\mathcal{M}} \approx id_{\mathcal{M}}$ are referred to as an AutoEncoder for $\mathcal{M}$. These are usually modelled as neural networks optimised to minimise the reconstruction error of some dataset. When constrained to reduce the dimensionality of data ($d < n$), 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.

A popular extension of AutoEncoders are Variational AutoEncoders (Kingma & Welling, 2013; Rezende *et al.*, 2014). VAEs approach the AutoEncoder in a probabilistic manner. They model the distribution of data in the ambient space under a Bayesian framework: $p(x) = \int p(x|z;\theta)p(z)dz$. The VAE generally assumes that the prior $p(z)$ is a $\mathcal{N}(0, \mathbf{I}_d)$ and so the likelihood $p(x|z;\theta)$ of generated data from some $z \in Z$ is distributed as a $\mathcal{N}(\mu(z), \sigma(z)\mathbf{I}_n)$. Then, $\mu(z) : Z \longrightarrow X$ 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 in X. Kingma & Welling (2019) present a comprehensive overview of Variational Autoencoders and mention some of its extensions.

These concepts have been applied to deep generative modelling. Tosi *et al.* (2014) postulate that the manifold embedded in the ambient space induces a Riemannian geometry in the latent space. Then, they propose a metric in Z which changes locally according to the reflected changes in X. More precisely, the metric tensor on Z is based on the Jacobian of the generator function with respect to the latent space. Subsequently, geodesics between points are formulated. Due to some practical constraints of the underlying generative model used by Tosi *et al.* (2014), similar concepts were applied to Variational AutoEncoders (Arvanitidis *et al.*, 2017; Shao *et al.*, 2018; Chen *et al.*, 2018). As VAEs are probabilistic models, the deduced metric tensor is stochastic rather than probabilistic. Regardless, the different authors consider the expected value of the Tensor for their developments.

Arvanitidis *et al.* (2017) deduce an ordinary differential equation for computing the geodesic. Their approach is not computationally efficient. Yang *et al.* (2018) expand on this idea and parametrize geodesics as quadratic splines. More importantly, they solve these equations while benefiting from Automatic Differentiation. This is relevant because of the reiterative use of Jacobians when evaluating the metric tensor at a given point in Z. Both approaches adjust the metric tensor to prevent geodesics from traversing areas where the generative model has high uncertainty; that is, where little data is used for training. Chen *et al.* (2018) approximate geodesics as neural networks. Additionally, the optimisation is performed on the length of the geodesic, with an added regularisation term. This term encourages the geodesic to follow the generated manifold, similar to the aforementioned adjustment (Yang *et al.*, 2018).

Laine (2018) perform a similar analysis, albeit utilising a different metric to the induced Riemannian one. They use a feature-based metric based on the activations' values of samples when passed through a VGG-19 network. In other words, similar images should display similar activations when passed to the neural network. Hence, the metric structure is derived from the data but not from the ambient space. Still, geodesics are computed in the latent space for practical reasons. Chen *et al.* (2019) utilise data to build a graph where every data point is connected to its k nearest neighbours. Then, geodesics consist on the shortest path through the graph connecting a pair of points. They argue their approach is fast and provides smooth interpolations in the original space. Arvanitidis *et al.* (2019) argue that the estimation of the Riemannian metric tensor can be quite unstable, specially on areas with few data points. They propose a numerical algorithm which does not explicitly compute the jacobians associated with the tensor and they argue it improves precision and speed. Chen *et al.* (2020) completely evade the problem of estimating geodesics, and rather propose a method that produces and approximately Euclidean latent space. They regularise the metric tensor during VAE training in order to obtain a flat manifold.

6 Proposed methodology

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 $\mathcal{M}$. When f is stochastic as produced by a VAE with networks $\mu : Z \longrightarrow X$ and $\sigma : Z \longrightarrow \mathbb{R}^+$, the expectation of the Jacobian is selected. This is $\mathbb{E}[\mathrm{J}_z^T \mathrm{J}_z] = (\mathrm{J}_z^\mu)^T(\mathrm{J}_z^\mu) - (\mathrm{J}_z^\sigma)^T(\mathrm{J}_z^\sigma)$ (Arvanitidis *et al.*, 2017). Hence, the dot product in $T_z\mathcal{M}$ can be reformulated as a dot product in Z which varies according to the metric tensor $\mathrm{J}_z^T \mathrm{J}_z$.

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

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

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

Minimising the length of the geodesic is equivalent to minimising the energy of traversing it Yang *et al.* (2018). This yields an objective function which is easier to optimise:

$$\mathcal{E} = \int_0^1 g_w'(t)^T \mathrm{J}_{g_w(t)}^T \mathrm{J}_{g_w(t)} g_w'(t) dt$$

In practice, this integral can not be explicitly computed. However, we approximate it via a discretization of $[0, 1]$ into n equidistant points (Chen *et al.*, 2018). The estimated energy for a geodesic, as well as the loss function to be used during back-propagation is as follows:

$$\mathcal{L}(g_w(t)) = \frac{1}{n}\sum_{i=1}^{n} g_w'(t_i)^T \mathrm{J}_{g_w(t_i)}^T \mathrm{J}_{g_w(t_i)} g_w'(t_i) \tag{1}$$

$$= \frac{1}{n}\sum_{i=1}^{n} ||\mathrm{J}_{g_w(t_i)} g_w'(t_i)||_2^2 \tag{2}$$

Note that the derivative of the loss function with respect to the neural network's parameters is required during its training. As our particular loss function requires Jacobians in itself, a second order derivative will be evaluated during backward passes. Because of this, piece-wise linear activation functions would produce zero-valued gradients and would render optimisation impossible. Hence, hyperbolic tangent, sigmoid and softplus activation functions will be considered.

An additional constraint is that g must follow the data manifold. In practice, the considered geometry is only well behaved near real data points. Lowly populated regions in the latent space will produce poor estimations of the generator's networks. There, the Jacobian and metric tensor will be imprecise. Then, in order for the geodesic to avoid high uncertainty regions, the σ networks is adjusted to exponentially increase the generators variance in areas away from data points. This will force geodesics to avoid high uncertainty areas. We will do this according to what is presented by Arvanitidis *et al.* (2017). Furthermore, the network must satisfy the boundary conditions $g(0) = z_0$ and $g(1) = z_1$. In order to do this, we propose to model the geodesic as:

$$g(t) = (1-t)z_0 + tz_1 + t(1-t)\psi(t, \theta) \tag{3}$$

Where $\psi(t, \theta)$ is a feed-forward neural network with parameters θ. Note that $g(t)$ trivially satisfies the boundary conditions and is continuously differentiable with respect to t and θ. Additionally, when t is closest to a midpoint in $[0, 1]$, the gradient of g with respect to θ is amplified in comparison to extreme values of t. This indicates training of ψ will focus on points further away from z_0 and z_1, which are the relevant ones during interpolation.

We will implement our proposed model for the geodesic. After that, it will be tested on various data sets. We intend to visualise samples generated with our interpolations. This will allow us to see whether our method is producing reasonable or noisy outputs. For labelled data like MNIST, we plan to see how likely it is for our geodesics to cross between classes. Also, using our distance metric, some algorithm like k-means will be applied to see whether clusters are coherent with said class labels. As our methodology will be unsupervised, these will be indicators of how well it captures the structure of the data. In all of the aforementioned cases, the results will be contrasted with their Euclidean counterparts: linear interpolation or $L2$ k-means. Finally, the obtained results will be analysed and compiled into a report.

7 Schedule

Table 1 presents the projected schedule for the development of this project.

Table 1: Schedule

Activity	*Weeks*																	
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
Literature review	■	■	■	■														
Model development				■	■	■	■	■										
Model implementation									■	■	■	■	■					
Results analysis													■	■	■	■		
Final report redaction																■	■	■

8 Intellectual property

According to the internal regulation on intellectual property within Universidad EAFIT, the results of this research practice are a product of *Juan Ramirez*, *Jose Gallego* and *Maria Eugenia Puerta*.

In case of further production, beside academic articles, that could be generated from this work, the intellectual property distribution related to it will be directed under the current regulation of this matter determined by Universidad EAFIT (2017).

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

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.*

Laine, Samuli. 2018. Feature-based metrics for exploring the latent space of generative models.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, & Haffner, Patrick. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.

Rezende, Danilo Jimenez, Mohamed, Shakir, & Wierstra, Daan. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

Shao, Hang, Kumar, Abhishek, & Thomas Fletcher, P. 2018. The riemannian geometry of deep generative models. *Pages 315–323 of: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

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

Universidad EAFIT. 2017. *Reglamento de propiedad intelectual*.

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*.

Xiao, Han, Rasul, Kashif, & Vollgraf, Roland. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

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