Investigating methods incorporating Latent Space Representations

Vedant Devank Patadia
Lakehead University - Computer Science
email: vpatadia@lakehead.ca

Abstract— **Latent space representations are the building blocks of any machine learning or deep learning structure through which one can make out the general working of any newly made model. But it is only due to the experience with building and training countless structures that an engineer can predict the consequences of various scenarios that the structure may get into which is generally termed as data intuition. Data intuition is nothing but simply learning a trend in how the latent space behaves as the model is trained, i.e., practically delving into the dependency between the different inputs and the outputs being generated through the training process. The algorithms relying on the in depth understanding of latent learning are coming into the trend as the computational capacity is increasing to the new heights and this research is geared towards investigating such methods for the period of a month and, further recreating one of the unique interpretation against the common trends while maintaining feasibility.**

I. INTRODUCTION

The world around us is in 3 Dimensions, and the 4 dimensional visualization is the farthest to be on-sight interpretable. Thus, after 3 dimensions working with data and factually traversing through the process is quite hard for the untrained eye, let alone understanding anything beyond 4 Dimensions.

The basis of latent space representation is that, we embed these higher dimensions within lower dimensional representation, or typically compressed form which is typically a lossy process. In a more specific sense, the higher dimensional data can be learnt by a lower dimension latent space, which serves as a manifold for further generation, as described in [1].

Further as inspired by [5], the model being termed as a "Black Box", is simply not acceptable at certain tasks as reaching saturation on such critical tasks is not a simple task and what more is that the "Black Box" representation of AI models may lead to them being never leaving the Research phase off itself. What more is that the research in this field is never ending, and the importance is often overlooked, as surveyed in [6].

The importance of being able to observe the performance of an AI Model is non-trivial, and sometimes able to solve identified problems in short period that otherwise may have taken longer. This is quite evident from the development of StyleGAN2 from [8], where the authors not only did improve the model capacity but also solved artifact problems that were inherent in StyleGAN[9]. Although its a good example involving interpreting, incorporating and visualizing the representations for which the latent space data is responsible for, but the research is quite a high profile one involving researchers backed by NVIDIA.

The question proposed here and the experimental approach is quite similar to [10], where the authors use Generative Adversarial Networks[**?**] in interpreting the latent space of the same. The application itself is what they call InterFaceGAN, in which they explain the Latent Space of GANs as a Riemannian manifold and their modification of popular GANs like StyleGAN[9] and PGGAN[12] to encode semantics into them, i.e., add conditional manipulation on the GAN to create different set of manifolds on the same image based off on an attribute. Their interpretation in latent space is mediated by change in the input Faces by attribute manipulation, where they find that the more borderline the generation is, the more lookalike output will be to the input to the model, and if they go farther then at a certain point the faces will come out to be entirely different.

II. RESEARCH QUESTION

Even in today's computational prowess, the power of understanding and incorporating the techniques involving latent space is quite underestimated, further its treated as an under-performing field for the general audience.

The reason is quite simple, most of the researched geared towards it are high profile, and the requirements are still out of the reach for the most population. Adding to it, the majority of summarizations and publications are more inclined towards the mathematical and procedural aspects, thus the science behind is generally overlooked.

In the proposed research question, the solution is geared in onlooking the basic and recommended requirements to even approach the domain. Further, provide an experimental representation of what can be achieved using them.

III. RESEARCH METHODOLOGY

The approach is quite straight forward and can be typically divided into three bases:

- Constraint formulation
- Gathering Data
- Experimentation

Each is inter-related but at after a point are capable to evolve independently

A. Constraint Formulation

The Constraint Formulation, is basically intended to limit our search size while both "Gathering Data" and "Experimentation", the sole consequence to that would be a more simplified and approachable methodology which will help focusing on specific solutions.

It is a fact that, each and every implementation in the field of Artificial Intelligence is actually modelled with a latent space representation. Further the latent space quite literally refers to the embeddings of data to cover the feature space and draw conclusions from it.

To clearly investigate the domain core, i.e., Latent space representations it is important to limit our search space to where the direct conclusions can be drawn from the results reliably, concisely and transparently. Here, Generative Adversarial networks, Variational Auto Encoders and/or Auto-Encoder Networks are practically few of the best solutions for the experimental setup, also Representation Learning, DeepMDP as described in [2] will help us quantify our research and the Latent Space Physics described in [3] will drive the lime light towards how exactly the current researches are being done for the same.

Further, the selection domain will be constrained to Image, i.e., Computer Vision which is not only does have the majority of work in our topic domain core, but they are generating results from images which is easy to understand to even the most inexperienced in the field.

In return for this, we will expect some limitations to be put on the Experimental Setup.

B. Gathering Data

The phase of gathering data will start only after successful "Constraint Formulation" and the idea is to formulate two streams of data, one in quite the literal sense, i.e., actually selecting datasets, and sub sampling important context for the experimental setup. The other stream is in form of domain flow, where different researches related to our core domain will be contrasted to draw the the ideas and their results to be used for a concrete understanding of the concepts as well as the applications of Latent space representations.

The major keywords to be used in this stage are:

- Unsupervised Learning
- Representation Learning
- Manifold Learning
- Auto Encoders
- Adversarial Networks
- Generative Networks

Further, the focus group for the base is decided in the "Constraint Formulation", which will evolve to adhere the research question to its solution.

More, "Kaggle" and "PapersWithCode" will serve as the data collection source for experimentation setup. Although it is not a general practice and the results or metrics that will generate with the data from the sources may not be deemed for a technical research for which more generalizations will have to be made.

C. Experimentation

The experimentation will also start after enough constraints are placed during "Constraint Formulation", and after enough data is gathered during "Gathering Data" for predictably successful implementation of the experimental setup.

The general workflow of the experimentation and a finalized approach is adapted from [1] and combined with the methodology of [7]. Our use case is nothing exceptional, experimenting with the Deep Learning Model, explaining the latent space behind it and how to justify its applicability. Thus the steps involve:

1) Data Collection
2) Data Preprocessing
3) Model Selection
4) Adapting model to Use case
5) Preparing data for ingestion
6) Training model
7) Testing for performance metrics
8) Visualizing the model basis

For a final touch, the contrasted results from 8) and 9) will generate some interesting insights. Though one major factor for the potential limitation of this approach will be the quality of the data points.

Where it is factual that research proposed here is quite intensive at the first glance, the approach chosen is to be intended for shorter span and lower profile this will likely generate some vulnerabilities and lower quality results. Although they will be enough to justify our solution.

IV. Workflow and Requirements

The whole research will provide the following approach stages:

1) Literature Review
2) Refining parameters (for selection)
3) Enlisting Methodologies
4) Requirements Finalization (finalized, on what can be procured)
5) Experimental Setup
6) Performing Experiment
7) Report Generation
8) Final Conclusion

A. Schedule

The overall schedule is described in the Figure 1 and the approximate schedule representation is as follows:

- Starting on - 2021-05-16: Literature Review
  Should take about: 60 days, and delivers to "Refining parameters", "Enlisting Methodologies" & "Experimental Setup"
- Starting on - 2021-06-18: Refining parameters
  Should take about: 25 to 30 days, and delivers to "Enlisting Methodologies" & "Requirements Finalization"
- Starting on - 2021-06-11: Enlisting Methodologies
  Should take about: 30 to 35 days, and delivers to "Requirement Finalization" & "Final Conclusion"
- Starting on - 2021-06-14: Requirements Finalization
  Should take about: 30 days, and delivers to "Experimental Setup" & "Report Generation"
- Starting on - 2021-07-01: Experimental Setup
  Should take about: 15 days, and delivers to "Performing Experiment" & "Final Conclusion"

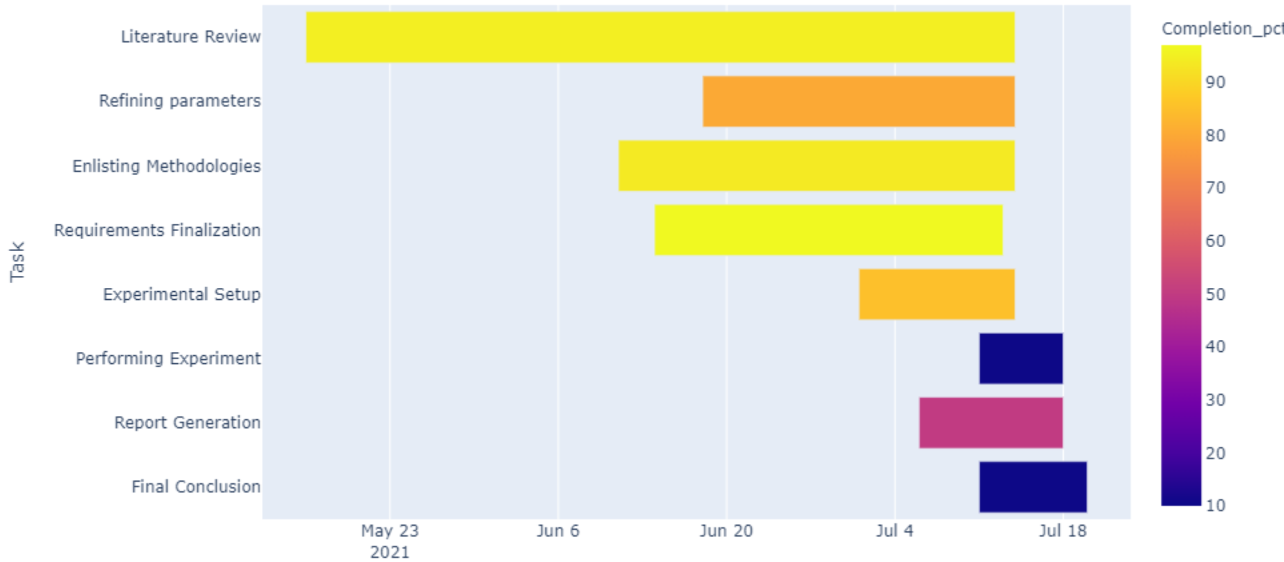

Fig. 1. The Schedule

- Starting on - 2021-07-11: Performing Experiment
  Should take about: 5 to 10 days, and delivers to "Report Generation"
- Starting on - 2021-07-06: Report Generation
  Should take about: 10 to 15 days, and delivers to "Final Conclusion"
- Starting on - 2021-07-11: Final Conclusion
  Should take about: 10 days, and delivers to the "Final Research"

B. Requirements

The formal requirements are based on the literature review performed beforehand and the minimum requirements are speculated on basis of the general research trends and hardware at hand. Since the release of Google Colaboratory[13] in 2017, the actual hardware requirements of a research has been brought down a lot. The minimum requirements still renders on the basis of how the CPU performs in actually processing the outputs of those long notebooks. But still here we will assume an isolated/semi-isolated environment, which we can tie up with the basic hardware given to a Free Google Colaboratory Session/Kernel/Runtime.

Further, the Recommended Requirements are inspired from the StyleGAN implementation[15] and we can add another category here for taking the "NVIDIA DGX-1" and "Intel Knights Landing"[14] in account, which may seem overkill at a point but are quite realistic estimate.

- Minimum Hardware Requirements (To run Colaboratory)
  - OS: Windows, Linux or MacOS, just need a web browser
  - RAM: 8 GB, To interact with large enough notebooks as well as maintain the system sanity
  - Graphics: any supported dedicated graphics card, to give a smooth web browsing performance.
  - CPU: a Dual Core Processor is a must
- Minimum Hardware Requirements
  - OS: Windows or Linux, Linux recommended for better performance and compatibility
  - RAM: 16 GB, To load the data and maintain the variables at the least
  - Graphics: High end GPU, preferably from NVIDIA with atleast 6GB VRAM and CUDA Enabled
  - CPU: a recent Intel i7 Quad Core or better, to cope with the short burst CPU loads
  - Storage: a 256 GB M.2 PCIe SSD paired with 512GB SATA SSD or more, so loading data won't take days
- Recommended Hardware Requirements
  - OS: Windows or Linux, Linux recommended for better performance and compatibility
  - RAM: 64 GB, To keep data in RAM and cope with recurrent reads and writes
  - Graphics: High end Compute GPU, preferably from NVIDIA with atleast 11GB VRAM and CUDA Enabled
  - CPU: a recent generation Octa-Core or better, to keep up with CPU sensitive loads and frequent data reads and writes
  - Storage: a 2 TB M.2 PCIe SSD paired with 8 TB SATA SSD or more, for faster data loading
- Recommended Hardware Requirements("NVIDIA

DGX-1" and "Intel Knights Landing")
  - OS: Windows or Linux, Linux recommended for better performance and compatibility
  - RAM: 128 GB, To keep data in RAM, cope with recurrent reads and writes and further decrease CPU interrupts
  - Graphics: Eight Tesla V100 GPUs
  - CPU: a High end CPU of recent generation with 16 cores or better, to keep up with preprocessing loads and data reads and writes
  - Storage: a 8 TB M.2 PCIe SSD paired with two 8 GB SATA SSD or more, for storing and accessing multiple datasets

The "Minimum Hardware Requirements" is of course the necessity and the "Recommended Hardware Requirements" will make the process faster and more reliable but the best performance is given by the "Recommended Hardware Requirements("NVIDIA DGX-1" and "Intel Knights Landing")" build, which will literally "turn weeks into days"[14]

V. SUMMARY

The proposal is to, investigate Generative Adversarial Networks, Variational Auto Encoder Networks, Auto Encoder Networks and Representation/Manifold Learning as a step towards summarizing the efforts of the Latent Space Representation towards them, both by experimenting with a research methodology and surveying different approached where such representations play a more than trivial role. Thus providing an effort to make the general problems more approachable.

REFERENCES

[1] C. Fefferman, S. Mitter, and H. Narayanan, "Testing the manifold hypothesis," J. Amer. Math. Soc., vol. 29, no. 4, pp. 983–1049, Feb. 2016, doi: 10.1090/jams/852. [Online]. Available: http://dx.doi.org/10.1090/jams/852

[2] Gelada, C., Kumar, S., Buckman, J., Nachum, O. and Bellemare, M.G.. (2019). DeepMDP: Learning Continuous Latent Space Models for Representation Learning. Proceedings of the 36th International Conference on Machine Learning, in Proceedings of Machine Learning Research 97:2170-2179 Available from http://proceedings.mlr.press/v97/gelada19a.html

[3] S. Wiewel, M. Becher, and N. Thuerey, "Latent Space Physics: Towards Learning the Temporal Evolution of Fluid Flow," Computer Graphics Forum, vol. 38, no. 2, pp. 71–82, May 2019, doi: 10.1111/cgf.13620. [Online]. Available: http://dx.doi.org/10.1111/cgf.13620

[4] L. Deng, "Deep Learning: Methods and Applications," FNT in Signal Processing, vol. 7, no. 3–4, pp. 197–387, 2014, doi: 10.1561/2000000039. [Online]. Available: http://dx.doi.org/10.1561/2000000039

[5] W. Samek and K.-R. Müller, "Towards Explainable Artificial Intelligence," in Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Springer International Publishing, 2019, pp. 5–22 [Online]. Available: http://dx.doi.org/10.1007/978-3-030-28954-6_1

[6] N. Burkart and M. F. Huber, "A Survey on the Explainability of Supervised Machine Learning," jair, vol. 70, pp. 245–317, Jan. 2021, doi: 10.1613/jair.1.12228. [Online]. Available: http://dx.doi.org/10.1613/jair.1.12228

[7] Y. Yu, Z. Ji, J. Guo, and Z. Zhang, "Zero-Shot Learning via Latent Space Encoding," IEEE Trans. Cybern., vol. 49, no. 10, pp. 3755–3766, Oct. 2019, doi: 10.1109/tcyb.2018.2850750. [Online]. Available: http://dx.doi.org/10.1109/tcyb.2018.2850750

[8] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, doi: 10.1109/cvpr42600.2020.00813 [Online]. Available: http://dx.doi.org/10.1109/CVPR42600.2020.00813

[9] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, doi: 10.1109/cvpr.2019.00453 [Online]. Available: http://dx.doi.org/10.1109/CVPR.2019.00453

[10] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the Latent Space of GANs for Semantic Face Editing," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, doi: 10.1109/cvpr42600.2020.00926 [Online]. Available: http://dx.doi.org/10.1109/CVPR42600.2020.00926

[11] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (pp. 2672–2680). MIT Press.

[12] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.

[13] E. Bisong, "Google Colaboratory," in Building Machine Learning and Deep Learning Models on Google Cloud Platform, Apress, 2019, pp. 59–64 [Online]. Available: http://dx.doi.org/10.1007/978-1-4842-4470-8_7

[14] N. A. Gawande, J. A. Daily, C. Siegel, N. R. Tallent, and A. Vishnu, "Scaling Deep Learning workloads: NVIDIA DGX-1/Pascal and Intel Knights Landing," Future Generation Computer Systems, vol. 108, pp. 1162–1172, Jul. 2020, doi: 10.1016/j.future.2018.04.073. [Online]. Available: http://dx.doi.org/10.1016/j.future.2018.04.073

[15] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, doi: 10.1109/cvpr42600.2020.00813 [Online]. Available: http://dx.doi.org/10.1109/CVPR42600.2020.00813