A COMPARISON OF DEEP LEARNING ARCHITECTURES FOR THE GENERATION OF 3D DATA

Yasmin da Silva Bonfim[a] , Gabriel Sete Ribeiro Lago dos Santos[a], Gustavo Oliveira Ramos Cruz[a] , Flávio Santos Conterato[a]

[a] SENAI Cimatec, Brazil

Abstract: Faced with several Deep Learning architectures for generating artificial images, there is a need to identify which are the best for each use case. To compare several networks with the generative architectures Autoencoder, Variational Autoencoder, and Generative Adversarial Networks in the 3D MNIST dataset, 12 models with different hyperparameters were created. After training, the models were compared with loss functions to assess the difference between the original and artificial data, so that greater complexity did not translate into better performance, indicating the Autoencoder models as the best cost-benefit.

Keywords: Generative Networks; 3D data; Comparison; Machine Learning.

COMPARAÇÃO DE ARQUITETURAS DEEP LEARNING PARA A GERAÇÃO DE DADOS 3D

Resumo: Diante de diversas arquiteturas de Deep Learning para geração de imagens artificiais, surge a necessidade de identificar quais destas melhores se adequam a cada caso de uso. Com o objetivo de comparar diversas redes com as arquiteturas generativas Autoencoder, Variational Autoencoder e Generative Adversarial Networks no dataset 3D MNIST, foram criados 12 modelos com diferentes hiperparâmetros. Após os treinamentos, os modelos foram comparados com funções de Loss para avaliar a diferença entre os dados originais e aqueles artificiais, de modo que maior complexidade não se traduziu em melhor desempenho, indicando os modelos de Autoencoder como o melhor custo-benefício.

Palavras-chave: Redes Generativas; Dados 3D; Comparação; Aprendizado de Máquina.

1. INTRODUCTION

Computer vision has been providing many projects developed in the areas of image generation, with deep learning technologies (DL) showing great advances for the generation of data in 2D, making use of architectures already relevant in the area [1]. On the other hand, the 3D segment is often left in the background, either due to its high complexity concerning 2D or the computational power needed to process this data [2].

The need for automated 3D data generation comes from the difficulty in creating three-dimensional representations manually, requiring too much time and research to build the items that will be portrayed [3].

The 3D MNIST dataset was used, which has 12,000 images in three dimensions [4]. The data was adapted from MNIST, which has numbers from 0 to 9 handwritten in a 2D representation [5].

The article aims to compare the Autoencoder, Variational Autoencoder (VAE), and Generative Adversarial Networks (GANs) architectures regarding several evaluation metrics to present the performance of each architecture for representing 3D data [6-8].

1.1. Autoencoder

The Autoencoder (AU) architecture is composed of two smaller networks that seek to compress the input into a latent representation, a version where only the essence of its structure remains. In the first network, encoder, the original data is reduced to a one-dimensional vector h, where its characteristics are categorized by increasing importance, between 0 and 1, to be discarded or preserved. In the next step, the decoder network receives the vectorized structure and performs the inverse process, returning the data to its original size and aspect, but with only the essence of its structure [9].

$$Loss = -Log\ P(x|x') \quad (1)$$

To check the quality of the representation created by the network, the loss function observed in equation 1 is used, where -log P compares the original input x with its latent representation x'. The loss in an autoencoder should be as small as possible, but it will hardly be zero. Considering that one of the main characteristics of the AU architecture is to learn the essentials and return data with reduced dimensionality, a loss of value 0 implies a faithful reproduction of the image, which in turn denotes low learning of its main components, essentially creating a network that just returns your input without a concrete benefit [10].

1.2. Variational Autoencoder

The Variational Autoencoder (VAE) is an architecture composed of the union of two networks, an encoder, which maps the inputs and compresses

them from the input to the latent space, and the decoder, which maps the data from the latent space to perform its decompression. The difference between VAE and Autoencoder architectures is the guarantee of good properties in the latent space to allow the generation of new data. The latent space is the representation of the compressed data: its reproduction with lower dimensionality.

Broadly, the VAE requires the standard Gaussian distribution anterior to the latent space. Thus, the VAE tends to maximize equation 2 [11].

$$P(z) = N(z|0, I) \quad (2)$$

To solve it, the VAE needs to deal with defining the information that will be represented by the latent variable z and how to deal with the integral over z. The latent variable can be understood as the choice of a character to be generated by the model before assigning a value to any specific pixel, that is, the model will produce configurations for the generation of the character. The z settings tend to produce a character that resembles the initial die. Furthermore, the interpretation of dimensional samples can be extracted from a simple distribution, being it N (0, I), where I is an identity matrix [11]. That said, the model parameters are trained to minimize the reconstruction error between the reconstructed and the initial data, making use of the Loss function KL divergence, acting as a regularization term, to calculate this divergence.

1.3. Generative Adversarial Networks

Generative Adversarial Networks (GAN), are generative architectures based on Deep Learning, in which an adversarial training process takes place between two networks: A Generative model G that is based on the original distribution of data to generate a new sample, and a Discriminative model D that estimates the probability a data sample coming either from the original data distribution or from the sample generated by the Generative model G. This training occurs until the Discriminator becomes unable to discern between the original and generated data [8].

GANs are often used in the Computer Vision field to perform various tasks involving images. They can be used to generate higher resolution versions of images, create sketches, paintings, and others.

During the training stage of this architecture, with the data generated by the Generator model, the Discriminator model has the role of correctly classifying between real and generated data. In consideration of the above, the final function of value V (G, D) is based on Equation 3, which involves the minimization of the Discriminator's error and the maximum precision of the Generator when creating the images [8]:

$$\min_G \max_D V(D, G) = E_{x \sim P_{data}(x)} [\log D(x)] + E_{z \sim P_z(z)} [\log (1 - D(G(Z)))]. \quad (3)$$

1.4. Recurrent Neural Networks

The Recurrent Neural Network (RNN) is a type of artificial neural network, used for sequential data or time series. The RNN, unlike traditional neural networks, can remember previous information from the feedback, allowing the information to persist [12]. To decide, the network considers its current input and what it learned from the previous input.

It has a “memory”, which stores the information of the calculations performed, enriching the expressive power of the model by capturing causal and contextual information [13]. That said, RNN manages to reduce the complexity of parameters, in addition to adjusting the weights through back-propagation and descending gradient processes, facilitating the learning process.

As there were advances in the development of RNNs, other architectures were created from it, such as Long-Short Term Memory (LSTM) [13] and Gated Recurrent Unit (GRU) [14].

1.5. Convolutional Neural Network

The Convolutional Neural Network (CNN) is a neural network widely used in problems dealing with image data, such as pixels. Important applicability of CNN is the extraction or detection of image contents when the input propagates through deeper layers [15]. During the convolution process applied to images, weights are assigned to certain sets of pixels that can indicate lines, curves, and eventually, complex patterns, where higher weights denote greater importance of that set of pixels for the current task.

In addition, there are other types of convolutional neural network architectures, such as the Fully Convolutional Network (FCN), a type of convolutional neural network, which contains only convolutional layers, not having “Dense” layers.

1.6. Multilayer Perceptron

The Multilayer Perceptron, or MLP, is a simple artificial neural network with several interconnected neurons that present a non-linear mapping between an input vector and an output vector [16].

Efficiently, MLPs backpropagate the network’s error, based on that error, the weights of previous layers are recalculated starting from the last layer up to the first.

2. METHODOLOGY

The approach chosen for this work was the comparison between practical experiments of several generative networks with different activation functions, number of layers, and number of neurons per layer. This exploratory, empirical, quantitative, and qualitative research seeks to identify the advantages

of each architecture, ranging from the network training time to the quality of the data generated at the end of the process. The work was divided into 3 stages: (1) search, (2) generation, (3) evaluation and synthesis.

In stage (1), a literature review was carried out where relevant works on the AU, VAE, and GAN architectures were identified, to verify the validity of the proposed comparison. During (2) a single base model was created for each architecture, subsequently, the bases were adapted into 4 models, divided into FCN, CNN with MLP, LSTM, and GRU, amounting to 12 models. In stage (3), the results of the models were grouped in tabular form, comparing the differences between the original image and that generated through the loss metrics Binary Cross-Entropy and Mean Squared Error (MSE), described in Equations 4 and 5, generating a Table per metric, with both divided between architectures and their respective networks [17].

$$H(X) = -[\theta \log_2 \theta + (1 - \theta) \log_2(1 - \theta)] \quad (4)$$

$$\frac{1}{N} \sum_{j=1}^{D} \left(\theta_j - \bar{\theta}_j\right)^2 \quad (5)$$

3. RESULTS AND DISCUSSION

As can be seen in Table 1, the Autoencoder FCN model presented the best results for Binary Cross-Entropy, with a total loss of 0.1304 concerning the original data, followed by the Autoencoder models GRU, LSTM, and CNN with MLP, respectively, with the latter having the same loss value as the VAE model with the same architecture.

For the MSE metric, Table 2 demonstrates a similar hierarchy, with FCN, GRU, and LSTM Autoencoder models having the smallest difference between the original and generated data, followed by the VAE CNN with MLP.

Comparing the two Tables, it becomes noticeable that GANs obtained the worst performance for both metrics in all proposed architectures. To match the performance of GANs to that of competing models, it would be necessary to increase the time and computational power expressively, leading to the conclusion that this model should be preferentially used when there is a high processing capacity.

Table 1. Loss Binary Cross-Entropy

	CNN+MLP	FCN	LSTM	GRU
AU	0.2249	0.1304	0.1626	0.1495
VAE	0.2249	0.2946	0.2283	0.2272
GAN	5.5372	6.3513	12.9795	5.9794

Source: The Autor

Table 2. Metric MSE

	CNN+MLP	FCN	LSTM	GRU
AU	0.0760	0.0110	0.0535	0.0156
VAE	0.0749	0.0888	0.0765	0.0733
GAN	0.9264	0.9139	0.8890	0.8449

Source: The Autor

4. CONCLUSION

This study aimed to evaluate the AU, VAE, and GAN architectures in their ability to reproduce three-dimensional data using the MNIST 3D dataset as a basis.

Using the Mean Squared Error and Binary Cross-Entropy metrics, it was possible to observe that the AU-based models obtained representations closer to the original data, furthermore, these models required a lower tuning of hyperparameters and training time, obtaining high cost-effectiveness in comparison to other architectures.

In parallel, the VAE architectures obtained results close to the original data, with the LSTM and CNN models being comparable to the quality of the AUs. As for the GAN constructions, in addition to having a longer training, the resulting images and the metrics evaluated had poor quality.

Acknowledgments

We would like to thank the institution SENAI CIMATEC and the coordination of the Artificial Intelligence course for providing us with skills and competencies in preparation for this research. We thank professor Flávio Santos Conterato for his commitment and impassivity in guiding students for the construction of this article.

5. REFERENCES

[1] KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, p. 1097-1105, 2012. On:

<https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45bPaper.pdf>. Available at: 31 de july 2021.

[2] AHMED, Eman et al. A survey on deep learning advances on different 3D data representations. **arXiv preprint arXiv:1808.01462**, 2018. On: <https://arxiv.org/pdf/1808.01462.pdf>. Available at: 31 de july 2021.

[3] TEO, Bee Guan; DHILLON, Sarinder Kaur. An automated 3D modeling pipeline for constructing 3D models of MONOGENEAN HARDPART using machine learning techniques. **Bmc Bioinformatics**, [S.L.], v. 20, n. 19, p. 1-21, 24 dez. 2019. Springer Science and Business Media LLC. http://dx.doi.org/10.1186/s12859-019-3210-x. On: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-0193210-x#citeas>. Available at: 31 july 2021.

[4] CASTRO, David de la Iglesia. 3D MNIST. **Kaggle**, 2019. On: <https://www.kaggle.com/daavoo/3d-mnist>. Available at: 30 de june 2021.

[5] LECUN, Yann. The MNIST database of handwritten digits. http://yann. lecun. com/exdb/mnist/, 1998. On: <http://yann.lecun.com/exdb/mnist/index.html>. Available at: 31 july 2021.

[6] BANK, Dor; KOENIGSTEIN, Noam; GIRYES, Raja. Autoencoders. arXiv preprint arXiv:2003.05991, 2020. On: <https://arxiv.org/pdf/2003.05991.pdf>. Available at: 31 july 2021.

[7] KINGMA, Diederik P.; WELLING, Max. An introduction to variational autoencoders. **arXiv preprint arXiv:1906.02691**, 2019. On: <https://arxiv.org/pdf/1906.02691.pdf>. Acesso em: 13 aug. 2021.

[8] GOODFELLOW, Ian et al. Generative adversarial nets. **Advances in neural information processing systems**, v. 27, 2014. On: <https://arxiv.org/pdf/1406.2661.pdf>. Available at: 1 aug. 2021.

[9] COSTA-JUSSÀ, Marta R.; NUEZ, Álvaro; SEGURA, Carlos. Experimental research on encoder-decoder architectures with attention for chatbots. **Computación y Sistemas**, v. 22, n. 4, p. 1233-1239, 2018. On: <https://www.researchgate.net/publication/332549468_Experimental_Research_on_Encoder-Decoder_Architectures_with_Attention_for_Chatbots>. Available at : 22 aug. 2021.

[10] MENG, Qinxue et al. Relational autoencoder for feature extraction. In: **2017 International Joint Conference on Neural Networks (IJCNN)**. IEEE, 2017. p. 364-371. On: <https://ieeexplore.ieee.org/abstract/document/7965877>. Available at : 20 aug. 2021.

[11] DOERSCH, Carl. Tutorial on variational autoencoders. **arXiv preprint arXiv:1606.05908**, 2016. On: <https://arxiv.org/pdf/1606.05908v3.pdf>. Available at: 1 aug. 2021.

[12] LIPTON, Zachary Chase *et al*. A Critical Review of Recurrent Neural Networksfor Sequence Learning. **Arxvi**, San Diego, p. 1-38, 5 june. 2015. On: <https://arxiv.org/pdf/1506.00019.pdf>. Available at: 1 aug. 2021.

[13] SHERSTINSKY, Alex. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. **Physica D**: Nonlinear Phenomena, [S.L.], v. 404, p. 132306, mar. 2020. Elsevier BV. http://dx.doi.org/10.1016/j.physd.2019.132306. On: <https://www.sciencedirect.com/science/article/abs/pii/S0167278919305974>. Available at: 1 aug. 2021.

[14] CHO, Kyunghyun et al. Learning phrase representations using RNN encoderdecoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014. On: <https://arxiv.org/pdf/1406.1078.pdf>. Available at: 1 aug. 2021.

[15] ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. **2017 International Conference On Engineering And Technology (Icet)**, [S.L.], p. 1-6, aug. 2017. IEEE. http://dx.doi.org/10.1109/icengtechnol.2017.8308186. On: <https://ieeexplore.ieee.org/abstract/document/8308186>. Available at: 14 aug. 2021.

[16] GARDNER, M.W; DORLING, S.R. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. **Atmospheric Environment**, [S.L.], v. 32, n. 14-15, p. 2627-2636, aug. 1998. Elsevier BV. http://dx.doi.org/10.1016/s1352-2310(97)00447-0. On: <https://www.sciencedirect.com/science/article/abs/pii/S1352231097004470>. Available at: 14 aug. 2021.

[17] MURPHY, Kevin P. **Machine learning: a probabilistic perspective**. MIT press, 2012. On:<http://noiselab.ucsd.edu/ECE228/Murphy_Machine_Learning.pdf>. Available at: 17 sep. 2021.