Experimentation with CNN and GAN

*

Miguel Valencia Ochoa
University EAFIT
Medellin, Colombia
mvalenciao@eafit.edu.co

Olga Lucía Quintero Montoya
University EAFIT
Medellin, Colombia
oquinte1@eafit.edu.co

***Abstract*—This paper presents the results of three diverse experiments in neural network applications. First, we explore the impact of varying hidden layer sizes in a Multi-Layer Perceptron (MLP) autoencoder for 3D data compression. Second, we assess the adaptability of the classic LeNet Convolutional Neural Network (CNN) for handwritten digit recognition using both the MNIST dataset and a custom dataset. Lastly, we demonstrate the artistic potential of pre-trained Generative Adversarial Networks (GANs) through video style transfer.**

These experiments showcase the versatility of neural networks in data compression, recognition tasks, and creative visual transformations. Our findings provide valuable insights into neural network architectures and their practical applicability across a spectrum of real-world scenarios

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

This paper presents the results of three distinct experiments in the field of neural networks, each focusing on unique applications and architectures. The first experiment investigates the effectiveness of a Multi-Layer Perceptron (MLP) autoencoder in reducing the dimensionality of 3D data. The autoencoder is constructed with a single hidden layer, and its performance is evaluated with varying numbers of neurons in this layer. Our findings reveal the impact of different hidden layer sizes on data reconstruction quality, shedding light on the trade-offs between model complexity and data representation fidelity.

In the second experiment, we explore the classic LeNet Convolutional Neural Network (CNN) for handwritten digit recognition using the MNIST dataset. Additionally, we evaluate the adaptability of the trained LeNet model to recognize handwritten digits from our own dataset. Through testing and evaluation, we showcase some of the vulnerabilities of this well-established architecture in digit recognition tasks.

The third experiment ventures into the realm of artistic neural networks, where we employ a pre-trained Generative Adversarial Network (GAN) for video style transfer. Leveraging the power of GANs, we successfully transfer the visual style of a reference image onto a video stream, resulting in captivating visual transformations. This experiment exemplifies the creative potential of neural networks beyond traditional classification and data compression tasks.

Our comprehensive study demonstrates the diverse capabilities of neural networks in various domains, offering insights into their performance, adaptability, and artistic expression. These findings contribute to a deeper understanding of neural network architectures and their applicability across a wide range of real-world scenarios.

II. Methodology

A. explicación general

The experimentation uses a different method for each step, the autoencoder via MLP uses a code previously developed in class, the CNN LeNet 5 uses the tensorflow python package to make the architecture and the GAN experiment uses a fast style method from google, also the data for each was different.

B. Autoencoder

The autoencoder used a method for creating MLP made for the prior deliverable using different architectures, all architectures were based on one single hidden layer and the size (number of neurons) varies from 1 to 10 and because of the nature of the normalization of the data (between 0 and 1) we use linear activation functions to keep as much importance (movement) in the region the model is in most cases.
The data used for the experiment is the dataAI.mat dataset showcasing variables of three dimensions, for making the autoencoder we use this data as the input of the network but also as the desired output so the network trains to represent the data as good as possible. .

C. CNN

LeNet method was used to classify the number an image of a handwritten digit represents, for this method the tensorflow constructed method is used, although here is the LeNet 5 architecture:

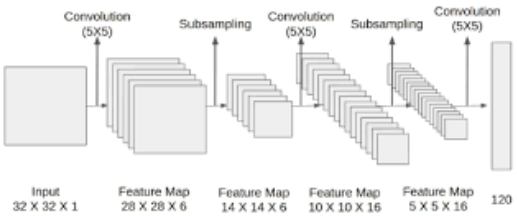

Fig. 1. Architecture of LeNet 5 network

And the model was trained using the classic MNIST handwritten digits dataset and tested with both the MNIST handwritten testing set and with a dataset made in class for experimenting how robust is the model and how can it be affected by the way of writting of people that created the MNIST dataset and a nowadays real life dataset.

D. GAN

For the styled GAN experimentation, a pretrained styling model is used, this model uses a styleGAN2 architecture from tensorflow hub google magenta package here:

https://tfhub.dev/google/lite-model/magenta/arbitrary-image-stylization-v1-256/int8/prediction/1

This model has the following architecture:

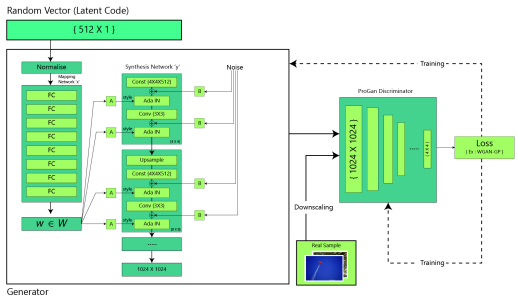

Fig. 2. Architecture of styleGAN2

The video to which the style applied is a fragment of ”Adventure of a lifetime” by coldplay and the image from which the style is taken is the painting of starry night by Vincent Van Gogh

III. Results

A. Autoencoder

Initially we splir the dataset in validation, testing and training sets:

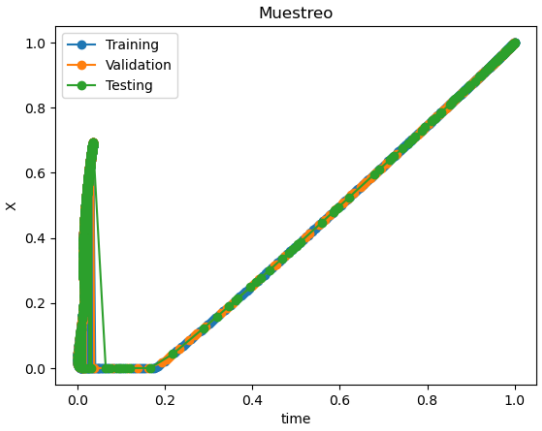

Fig. 3. sampling

And then we create the different possible models as autoencoders, first with compresion and then with expansion:

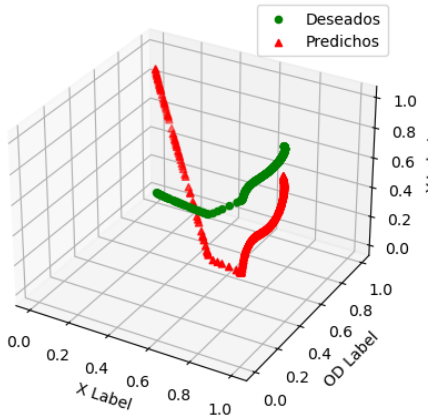

Fig. 4. 1 neuron in the hidden layer compression autoencoder

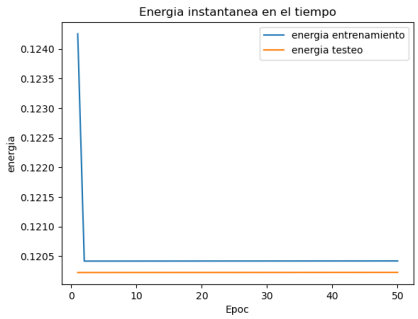

Fig. 5. Training and testing error for model with 1 neuron in hidden layer

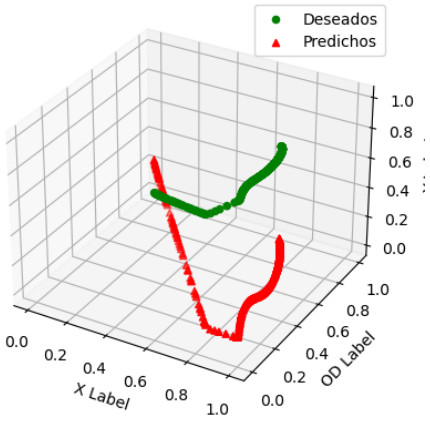

Fig. 6. 2 neuron in the hidden layer compression autoencoder

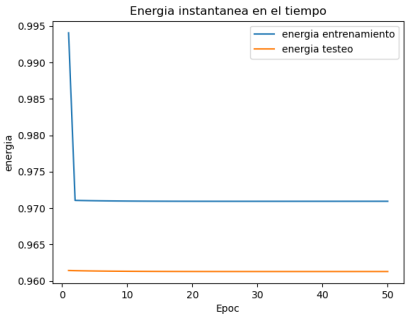

Fig. 7. Training and testing error for model with 2 neuron in hidden layer

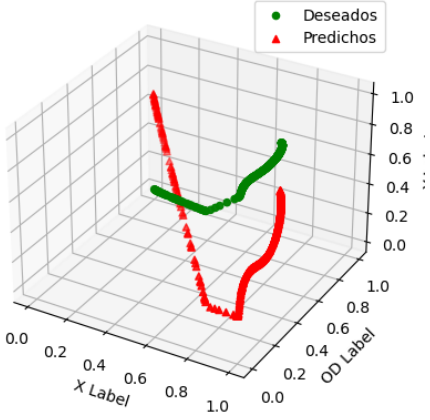

Fig. 8. 4 neuron in the hidden layer expansion autoencoder

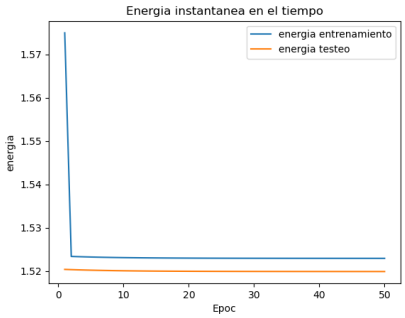

Fig. 9. Training and testing error for model with 4 neuron in hidden layer

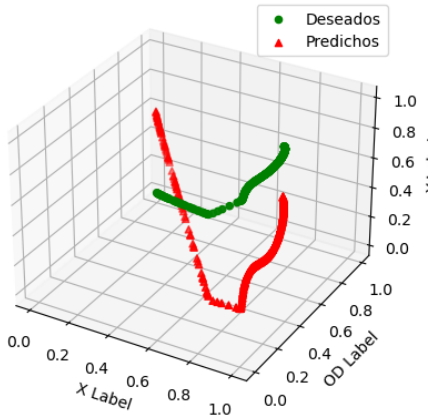

Fig. 10. 6 neuron in the hidden layer expansion autoencoder

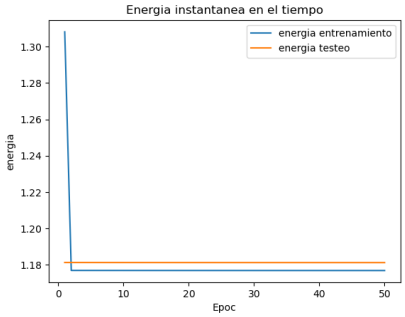

Fig. 11. Training and testing error for model with 6 neuron in hidden layer

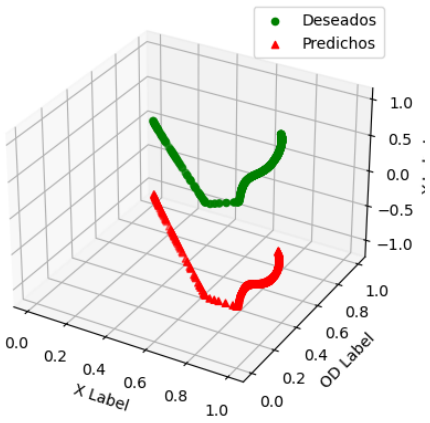

Fig. 12. 8 neuron in the hidden layer expansion autoencoder

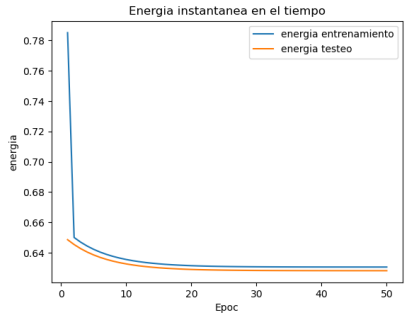

Fig. 13. Training and testing error for model with 8 neuron in hidden layer

B. CNN

Frist lets see a comparison between the data from the MNIST dataset and the data made in class:

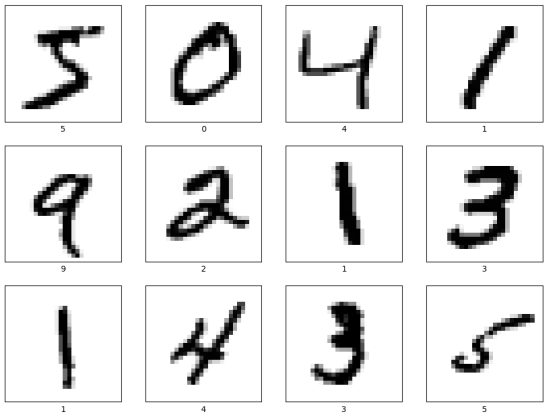

Fig. 14. numbers of MNIST dataset

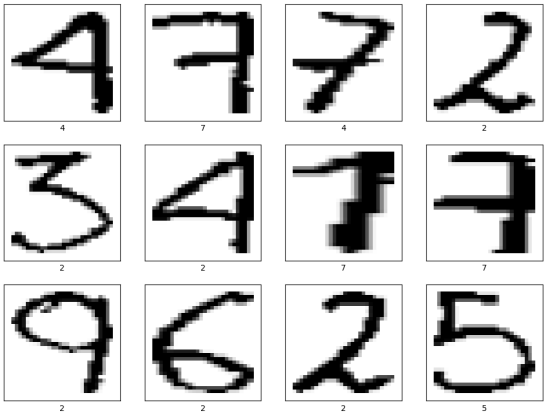

Fig. 15. numbers of our own dataset

The training of the model goes as follows in precision and accuracy.

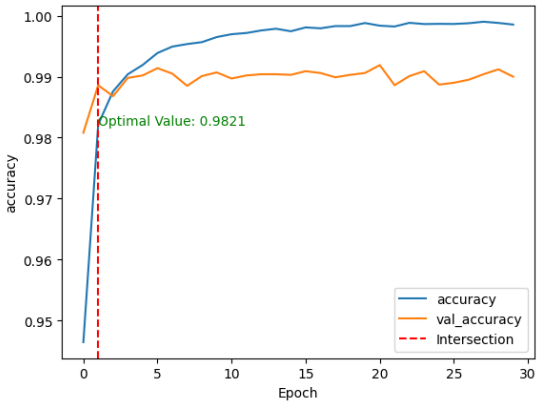

Fig. 16. Accuracy of trainning with LeNet5 with MNIST dataset

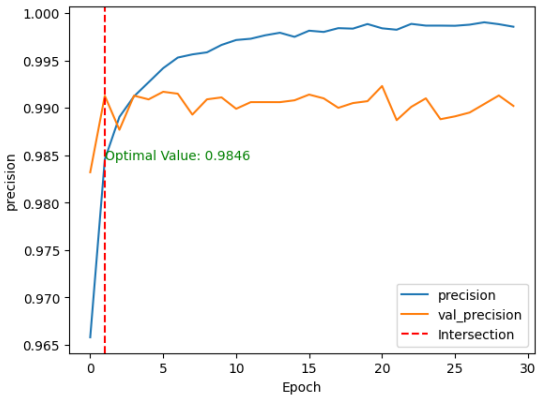

Fig. 17. Precision of trainning with LeNet5 with MNIST dataset

At last lets see the confusión matrix for the classification of the MNIST testing data and with our own data:

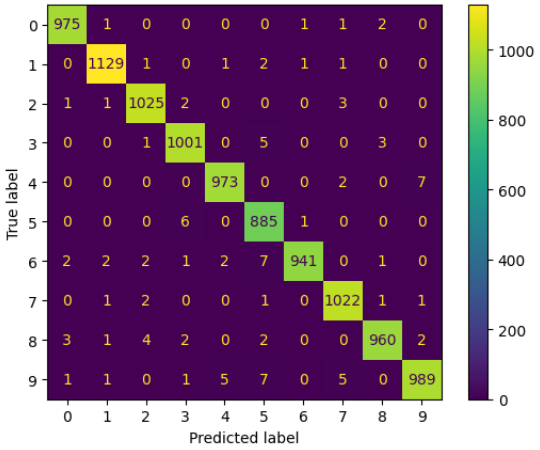

Fig. 18. Confusion matrix for testing of the MNIST dataset

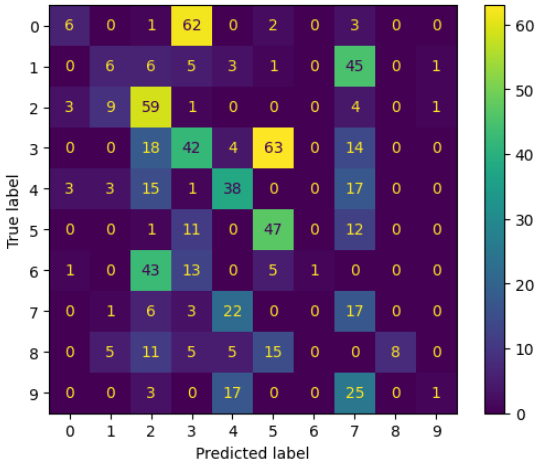

Fig. 19. Confusion matrix for testing with our own dataset

C. GAN

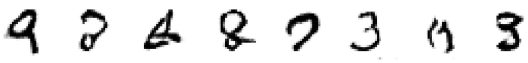

Fig. 20. GAN created pictures of handwritten digits

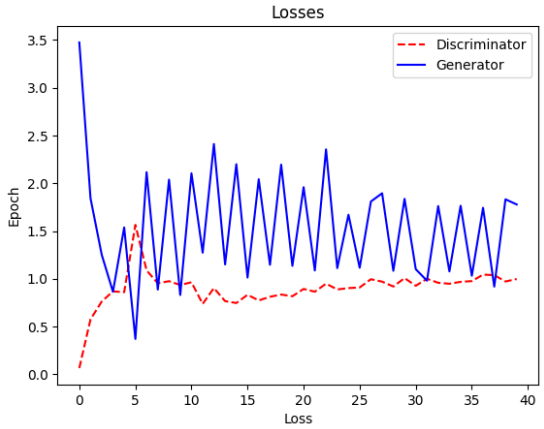

Fig. 21. GAN generator and discriminator loss

D. StyledGAN

The results for the StyledGAN are in the folder GAN in form of mp4 archives.

IV. Discussion

A. Autoencoder

The autoencoder with the better results in terms of representing the original data is the expansion one with 8 neurons in the hidden layer, this can be a consecuence of a ”slower training” that is shown in the error graph through the epocs, although this might change from time to time as the lack of a bias might be causing some change on the location of the values, anyways, most architectures end up having the same behaviour as the desired data. As most of them are very similar the ones recommended to use are either the architecture with one neuron in the hidden layer as it is the simplest and needs less time to run or the one with 8 neurons as it also represents the same general direction of the data. Here write about the keras autoencoder

B. LeNet5 CNN

Visually the handwritten digits from the MNIST dataset are very different from our own testing dataset, for example the number four is even written in a different way, also the ones from our own dataset are zoomed in and this might change the spatial correlation between the pixels in the picture probably causing the model to make mistakes.
Watching how the training goes for the LeNet5 we can see a high accuracy and high precision are reached from the very beggining in the first epochs meaning the model gets to learn very fast, with the confusion matrix we can see if the model is as good as it is fast and indeed, when validating the model with test data of the MNIST dataset we can see it gets most values correct having reached a very good configuration of weights but as expected because of the differences between datasets, there is a greater confusion between digits and in some cases you can even see what causes this confusions, for example you can say there is a similar spatial correlation between pixels

in the first 5 of the MNIST dataset and with the first 3 in our own dataset, in conclusion, for modern handwritten digit recognition a LeNet5 should be trainned with modern data as the way digits are written have changed significantly since the time the MNIST dataset was first created.

C. GAN

After training a GAN model with the MNIST dataset for 40 epocs we got some interesting results in III-C were some patterns can be seen have taken place, for example the number 9 in the first position is very clear and recognizable. Taking into account the loss shown in the III-C we can say maybe more epocs could improve the quality of the generator in general terms but after epoc 10 we have similar results each time so if computing time is a priority this could be the best number of epocs to train the GAN for.

D. styleGAN

The results for the styled video are pretty amazing and clearly sustain the Van Gogh patterns of the star drawings from starred night while still having recognizable figures from the original figure.