Hybrid learning methods in Artificial Intelligence

1st Pablo A. Osorio Marulanda
EAFIT University
School of Applied Sciences and Engineering
Medellín, Colombia
sisazac@eafit.edu.co

I. Introduction

With the aim of providing methods to solve problems applied in the real world, to adequately present the highly complex phenomena that exist in nature, this work is introduced, which applies methods using fuzzy inference systems, with learning machines that extract features from the data and reduce their dimensionality (convolutional networks). Additionally, the use of Generative Adversarial Networks was implemented.

II. Metodology

The methodology implemented for this work consists of four facets. First, an autoencoder is implemented to reduce the dimensionality of the data. Subsequently, the reduced data are used to integrate and run an ANFIS. Then, a LeNET-5 type convolutional network architecture is implemented, using an MNIST database. Finally, a GAN is used to perform a style transfer.

III. Methods

We pretend to present essential concepts in this section in order to solve the learning problem. The next sections will investigate these ideas in more depth.

A. Autoencoder

In order to create efficient encodings from unlabeled input, an artificial neural network technique known as an autoencoder is utilized (unsupervised learning). Encoding is checked and improved by attempting to reconstruct the input from the encoding. The autoencoder learns a representation (encoding) for a data collection, frequently to reduce dimensionality, by instructing the network to ignore unnecessary data. [1]

To define an autoencoder, we need to see the next concepts: The decoder family $D_\theta : \mathcal{Z} \to \mathcal{X}$, parametrized by θ (decoded) message, and the encoder family $E_\phi : \mathcal{X} \to \mathcal{Z}$, parametrized by ϕ. Typically, multilayer perceptrons are used to describe both the encoder and the decoder. One-layer MLP encoder $Ephi$ is, for instance:

$$E_\phi(\mathbf{x}) = \sigma(Wx + b)$$

where σ is an element-wise activation function such as a sigmoid function or a rectified linear unit, W is a matrix called "weight", and b is a vector called "bias".

An autoencoder is nothing more than a tuple of two functions on its own. We need an assignment in order to assess its quality. A "reconstruction quality" function $d : \mathcal{X} \times \S \to [0, \infty]$ and a reference probability distribution μ_{ref} over $\mathcal{X}$ are used to create a task. This function quantifies how much $xprime$ differs from x. These enable us to define the autoencoder's loss function as

$$L(\theta, \phi) := \mathbb{E}_{x \sim \mu_{\text{ref}}} \left[d\left(x, D_\theta\left(E_\phi(x)\right)\right) \right]$$

"Training the autoencoder" is the term used to describe this search procedure. The reference distribution is often merely the empirical distribution provided by a dataset $\{x_1, \ldots, x_N\} \subset \mathcal{X}$, so that

$$\mu_{ref} = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_i}$$

and the quality function is just L2 loss: $d\left(x, x'\right) = \|x - x'\|_2^2$. Then the problem of searching for the optimal autoencoder is just a least-squares optimization:

$$\min_{\theta, \phi} L(\theta, \phi), \text{ where } L(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} \left\| x_i - D_\theta\left(E_\phi\left(x_i\right)\right) \right\|_2^2$$

Several modifications attempt to force the learned representations to adopt useful characteristics. Examples include regularized autoencoders (Sparse, Denoising, and Contractive), which are effective in learning representations for subsequent classification tasks, and variational autoencoders, which have applications as generative models. Several problems, such as word meaning acquisition, feature identification, anomaly detection, and face recognition, can be resolved with autoencoders. Generative models like autoencoders have the ability to generate new data at random. This research's goal is to test an autoencoder's ability to encode ECG data from both healthy and ill individuals. An autoencoder arquitecture is shoed in figure 1

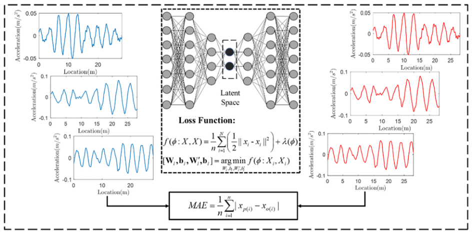


Fig. 1. Autoencoder Architecture [2]

B. Adaptative Neuro Fuzzy Inference System (ANFIS)

An adaptive neuro-fuzzy inference system, often known as an ANFIS, is an artificial neural network that is built on the Takagi-Sugeno-Kang (TSK) [3] fuzzy inference system (FIS). This technique was developed in the early 1990s. It provides the opportunity to integrate both of these technologies, combining the advantages of neural networks and fuzzy logic in a single framework. Its inference system has the capacity to learn and approximate nonlinear functions and resembles a fuzzy-defined collection of if-then rules. ANFIS is therefore viewed as a global estimator. The most efficient and successful way to use ANFIS is by using the best parameters produced by genetic algorithms. It is helpful in energy management systems..

Premise and consequence are the two parts of the network structure that may be separated. There are a total of five floors in the architecture. Using the input data, the first layer selects the membership functions. Another name for it is the layer of fuzzification. The underlying parameters a, b, and c are used to compute the membership degrees of each function. The second layer creates the firing strengths of the rules. Due to its function, the second layer is referred to as the "rule layer." The third layer normalizes the predicted trigger forces by dividing each value by the total The normalized values and the resulting parameter set p, q, and r are inputs to the fourth layer [4]. The initial layer of an ANFIS network describes how it varies from a conventional neural network. Data must first go through a preprocessing step where attributes are normalized into values between 0 and 1 in order for neural networks to work. Instead of utilizing a sigmoid function, an ANFIS network can carry out the preprocessing phase by transforming numerical inputs into fuzzy values. Figure 2 shows the architecture of ANFIS.

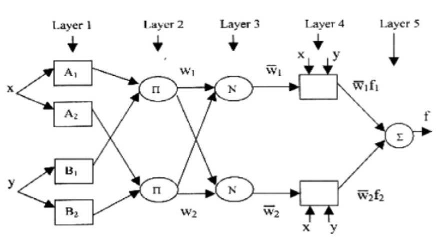


Fig. 2. ANFIS Architecture [5]

C. Convolutional Neural Networks

Artificial neural networks (ANNs) are convolutional neural networks (CNNs, or ConvNets), often employed in deep learning to analyze visual data. CNNs, also known as Shift Invariant or Space Invariant Artificial Neural Networks, are based on the shared-weight architecture of the convolution kernels or filters that slide along input features and create translation-equivariant outputs known as feature maps (SIANN). Contrary to common perception, most convolutional neural networks downsample the input, which prevents them from translating invariantly. They may be used for image and video recognition, recommender systems, image classification and segmentation, and medical image analysis.

Multilayer perceptrons are modified into CNNs. Fully linked networks, or multilayer perceptrons, are those in which every neuron in one layer is connected to every neuron in the following layer. Due to their "complete connectedness," these networks are vulnerable to data overfitting. Regularization or overfitting prevention methods frequently include punishing training parameters (such as weight decay) or cutting connectivity (skipped connections, dropout, etc.) By utilizing the hierarchical structure in the data and assembling patterns of increasing complexity using smaller and simpler patterns imprinted in their filters, CNNs adopt a novel strategy for regularization. CNNs are therefore at the lower end of the connectivity and complexity spectrum.

Convolutional networks were motivated by biological processes because to how closely the connection pattern between neurons resembles the structure of the animal visual cortex. Individual cortical neurons only react to stimuli in the restricted region of the visual field known as the receptive field. To complete the whole visual field, many neurons' receptive regions partially cross over one another. [6] .

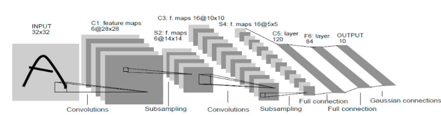


Fig. 3. CNN Architecture [7]

D. StyleGAN

The process of transferring an image's style from one domain to another is known as style transfer. Generative adversarial networks are a family of machine learning frameworks that Ian Goodfellow and his colleagues developed in June 2014 (GANs) [8]. One neural network gains at the expense of the other in a zero-sum game between two neural networks.

When given a training set, this approach learns to generate new data with the same statistics as the training set. For instance, a GAN that has been trained on photographs may develop brand-new photos with a variety of realistic qualities that, at least initially, give the impression that they were made by people. Despite being first proposed as a sort of generative model for unsupervised learning, GANs have shown useful for semi-supervised learning, fully supervised learning and reinforcement learning.

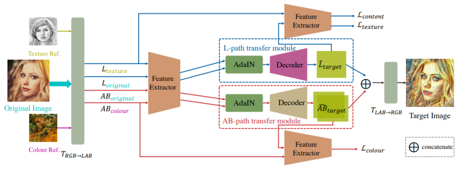


Fig. 4. Style transfer arquitecture [9]

IV. RESULTS

A. Data

Considering that the nature of the initial problem that was being used in the development of the work related to this subject is a classification problem, it was considered that, in order to exploit the capacity of the algorithms implemented here, a database that is given to a regression problem will be used. The diabetes dataset was used, which considers the following set of attributes:

- age in years
- sex
- BMI body mass index
- bp average blood pressure
- s1 TC, total serum cholesterol
- s2 LDL, low-density lipoproteins
- s3 HDL, high-density lipoproteins
- s4 tch, total cholesterol / HDL
- s5 ltg, possibly log of serum triglycerides level
- s6 glu, blood sugar level

Each of these 10 feature variables have been mean-centered and scaled by the standard deviation times the square root of n samples. And the target value is the median value of the house price. The target variable is a quantitative measure of disease progression one year after baseline

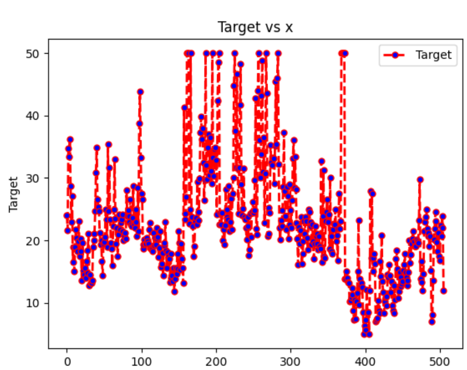
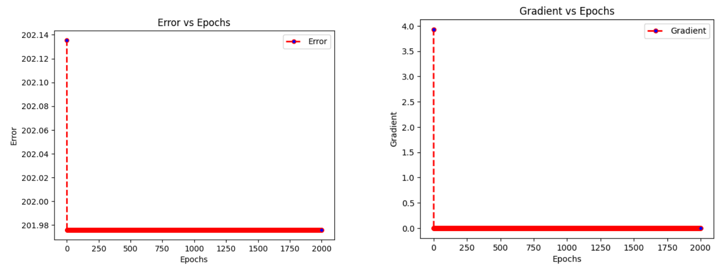


Fig. 5. Target variable (Prices)

B. Autoencoder

The autoencoder was used with the diabetes database. The results obtained are shown in the image 6. As is evident, there was not convergence for the error, but it was for the gradients. In light of this, it is evident that it was not possible to learn and encode the data from the used data set.



(a) Error

(b) Gradient

Fig. 6. Autoencoder results

C. ANFIS

With the results returned by the autoencoder, we start training ANFIS, which finally obtains the following configuration:

```
                     Name: "fis"
                AndMethod: "prod"
                 OrMethod: "max"
        ImplicationMethod: "prod"
        AggregationMethod: "sum"
     DefuzzificationMethod: "wtaver"
                   Inputs: [1×7 fisvar]
                  Outputs: [1×1 fisvar]
                    Rules: [1×128 fisrule]
   DisableStructuralChecks: 0
```

(a) Anfis configuration 1

```
ANFIS info:
    Number of nodes: 555
    Number of linear parameters: 2304
    Number of nonlinear parameters: 48
    Total number of parameters: 2352
    Number of training data pairs: 253
    Number of checking data pairs: 253
    Number of fuzzy rules: 256
```

(b) Anfis configuration 2

Fig. 7. ANFIS configuration

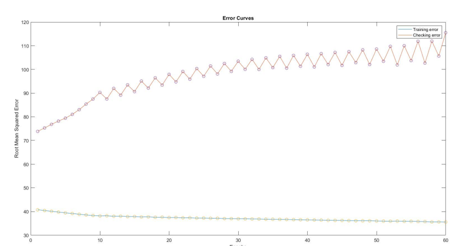


Fig. 8. Training error of ANFIS arquitecture

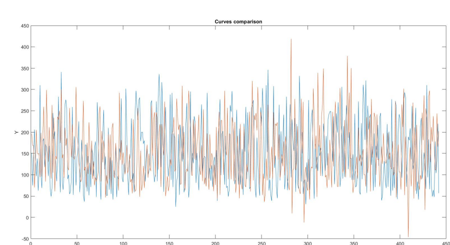


Fig. 9. Curves comparison

D. CNN

We consider a LeNet 5 structure in order to train a model able to classify an image into one of the digits, based on the MNIST digits dataset. For training set we took over two thousand samples, and a bigger size of validation set. Figure 10 and 11 shows a sample of the training set we're using for this work. Even though the digits are not drawn very clearly, our algorithm is nevertheless able to accurately classify virtually all of them.

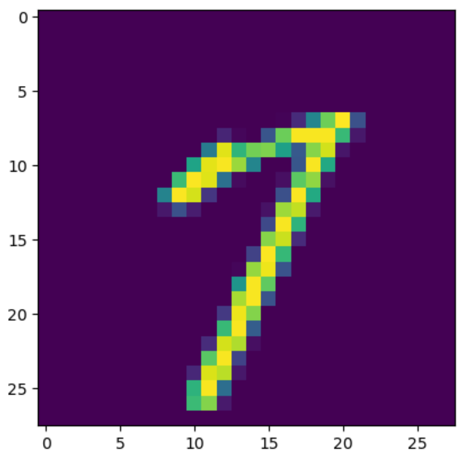


Fig. 10. Evaluating training set

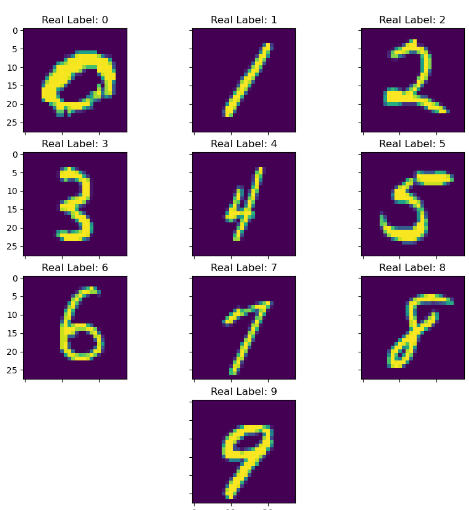


Fig. 11. Evaluating training set 2

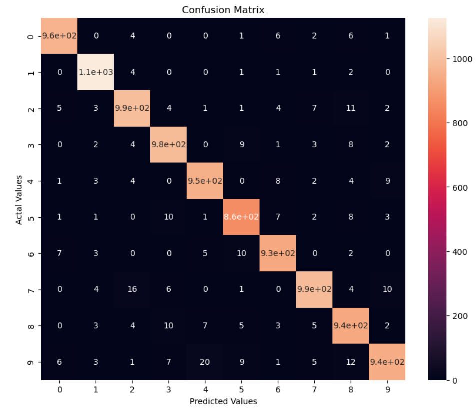


Fig. 12. Confusion network

Table I shows different metrics used to evaluate the performance on the net for the MNIST dataset. As is evident, the results are good, therefore the architecture is highly recommendable for this particular problem. Nonetheless, more data could help improve the already good results.

Accuracy: 0.9669 Precision: 0.9666296844517746 Recall: 0.9666026810722121 F1-Score: 0.9665681106370337

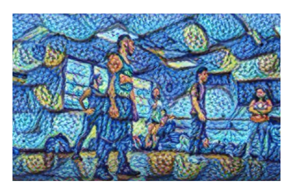
Metric	Score
Accuracy	0.9669
Precision	0.9666
Recall	0.9666
F1-Score	0.9665

TABLE I
LENET PERFORMANCE WITH MNIST DATASET

E. StyleGAN

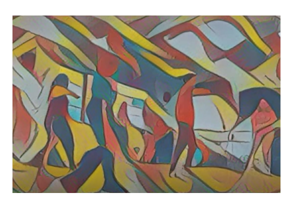
Some of the style transfer results can be seen in the following image, where a video of me dancing to a song was recorded in 2019.

(a) Normal frame



(b) Styled frame

(c) Normal frame



(d) Styled frame

(e) Normal frame

(f) Styled frame

(g) Normal frame

(h) Styled frame

(i) Normal frame

(j) Styled frame after

V. Conclusions

The LeNet5 architecture showed great performance even with a small training size for the convolutional neural network. This demonstrates the power of the model, which, as evidenced by the many measures employed, also demonstrated good performance on the validation and assessment set. With the aid of StyleTransfer, a new video may be created utilizing mobile phone footage as a foundation. One iconic frame style, inspired by Van Gogh's artwork, may be seen throughout the newly made film. In addition, the ANFIS structure did not show satisfactory results, although it can be hypothesized that this was due to the fact that the self-empowerment structure and its results were also unsatisfactory to their respective extent.

References

[1] M. Sewak, S. K. Sahay, and H. Rathore, "An overview of deep learning architecture of deep neural networks and autoencoders," *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 1, pp. 182–188, 2020.

[2] M. Z. Sarwar and D. Cantero, "Deep autoencoder architecture for bridge damage assessment using responses from several vehicles," *Engineering Structures*, vol. 246, p. 113064, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141029621012025

[3] P. Bergsten, R. Palm, and D. Driankov, "Observers for takagi-sugeno fuzzy systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 1, pp. 114–121, 2002.

[4] M. A. Denai, F. Palis, and A. Zeghbib, "Anfis based modelling and control of non-linear systems: a tutorial," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, vol. 4. IEEE, 2004, pp. 3433–3438.

[5] A. M. Ahmed and S. M. A. Shah, "Application of adaptive neuro-fuzzy inference system (anfis) to estimate the biochemical oxygen demand (bod) of surma river," *Journal of King Saud University - Engineering Sciences*, vol. 29, no. 3, pp. 237–243, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1018363915000021

[6] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*. Ieee, 2017, pp. 1–6.

[7] S. Indolia, A. K. Goswami, S. Mishra, and P. Asopa, "Conceptual understanding of convolutional neural network- a deep learning approach," *Procedia Computer Science*, vol. 132, pp. 679–688, 2018, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918308019

[8] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[9] Z. Hu, J. Jia, B. Liu, Y. Bu, and J. Fu, "Aesthetic-aware image style transfer," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 3320–3329. [Online]. Available: https://doi.org/10.1145/3394171.3413853