UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Artificial Intelligence

Assignment 5

Artificial Neural Network: feedforward.

Name	ID
Sergio Orlando Alanís De La Rosa	2043932
Diego Castro Galindo	2109304
Sebastian Hernandez Renteria	2109140
Braulio Azael García Treviño	2043046
Miguel Angel Perez Luevano	2052579

Professor Name: Daniel Isaías Lopez Paez

Career: IB **Group:** 003

Date: 27/10/25

Introduction: Artificial Neural Networks (ANNs) are computational models inspired by the biological neural systems found in the human brain. They are designed to recognize patterns, process data, and learn from experience through interconnected layers of nodes called neurons. Each neuron receives input data, processes it through a mathematical function, and produces an output that contributes to the decision-making process of the network. ANNs have become fundamental tools in the field of artificial intelligence because of their remarkable ability to learn complex relationships and perform a wide variety of tasks such as classification, regression, and prediction.

A Feedforward Neural Network (FFNN) is one of the most basic yet powerful types of artificial neural networks. Its architecture is organized in layers—an input layer, one or more hidden layers, and an output layer—where data flows in one direction: from input to output, without any feedback connections. Each neuron in one layer is connected to every neuron in the next layer through weighted connections. During training, these weights are adjusted in order to minimize the error between the predicted and the actual outputs. The backpropagation algorithm, combined with an optimization method, is typically used to update these weights efficiently.

For this project, the MNIST dataset was selected. This dataset is a benchmark in machine learning and artificial intelligence that contains 70,000 grayscale images of handwritten digits (0–9), each with a resolution of 28×28 pixels. It is widely used to test image classification models and evaluate the performance of different neural network architectures. The main goal of this activity was to train a feedforward neural network using this dataset and evaluate its accuracy in predicting unseen handwritten digits. This process allows us to understand the fundamental principles of neural networks, from data preprocessing to training, testing, and prediction evaluation.

Methodology: To complete this assignment, a Feedforward Neural Network (FFNN) was designed, trained, and evaluated using the Python programming language in Google Colab. The main libraries used were Keras and TensorFlow for neural network construction and training, and scikit-learn for data evaluation and preprocessing. The workflow followed the structure recommended in class: loading libraries, loading data, preprocessing, model selection and training, and model testing.

The first step was to import the required libraries, including TensorFlow, Keras, NumPy, and matplotlib. TensorFlow and Keras provided the tools to create and train the neural network, NumPy was used for numerical operations, and matplotlib helped visualize the dataset and results.

Next, the MNIST dataset was loaded using the `load_data()` function from the Keras library. This function automatically divided the data into training and testing sets, named `(train_data, train_labels)` and `(test_data, test_labels).` The dataset consists of grayscale images of handwritten digits ranging from 0 to 9. Each image has a dimension of 28x28 pixels, which was reshaped and normalized by dividing all pixel values by 255.0 so that the data would be in a range between 0 and 1. This normalization helps the neural network learn more efficiently.

In the model design phase, a sequential model from Keras was implemented. The structure of the Feedforward Neural Network included:

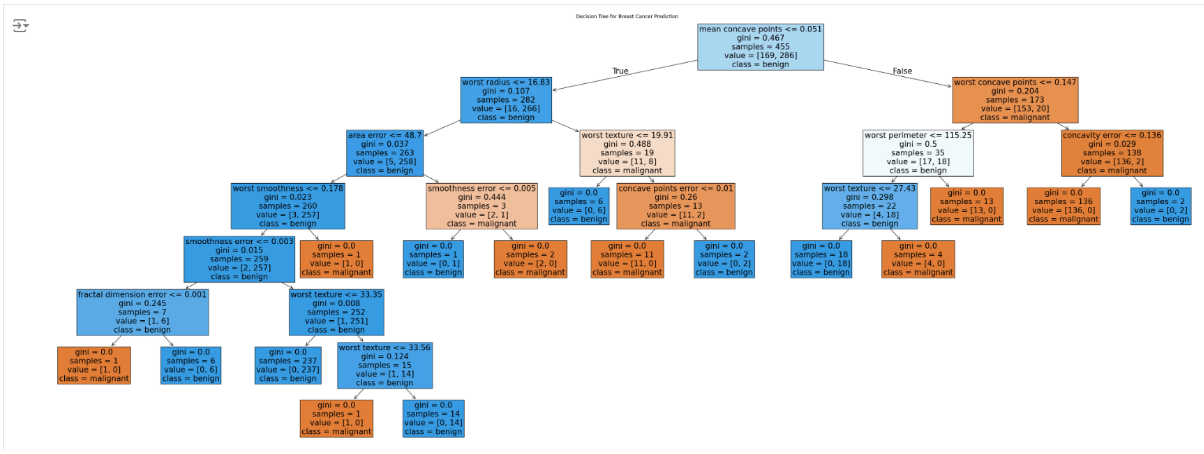
- An input layer that flattens the 28x28 pixel images into a one-dimensional vector of 784 elements.
- One or more hidden layers with a chosen number of neurons (for example, 128 or 256), each using the ReLU (Rectified Linear Unit) activation function to introduce non-linearity and improve learning capacity.
- An output layer with 10 neurons corresponding to the 10 possible digit classes (0–9), using the Softmax activation function to output probability distributions for classification.

The model was compiled using the categorical cross-entropy loss function, which is suitable for multi-class classification tasks, and the Adam optimizer, known for its efficiency and fast convergence. The main evaluation metric selected was accuracy, as it clearly measures how well the model predicts the correct digit classes.

The training process was carried out using the `fit()` function, which trained the model on the training dataset for a specified number of epochs (for example, 10 to 20) and with an appropriate batch size. During this stage, the model iteratively adjusted its weights to minimize the loss function and improve prediction accuracy. The training progress was monitored through accuracy and loss curves.

After training, the model was evaluated on the test dataset using the `evaluate()` function to determine its performance on unseen data. The resulting accuracy provided a quantitative measure of how well the model generalized beyond the training data. Finally, five random predictions were made using the `predict()` function on images from the test dataset. The predicted digit was displayed alongside the actual label, allowing a visual and numerical verification of the model's performance.

Results:



Google Colab:

https://colab.research.google.com/drive/1SB_S04g73oF0EK_T9vm6lyj3I41EJwJM?usp=sharing

Github:

https://github.com/SebastianHR19/Sebastian-s-2109140-Repository-

Conclusion:

Through this experiment, the implementation of a Feedforward Neural Network (FFNN) using the MNIST dataset demonstrated the fundamental principles of how artificial neural networks learn to recognize and classify visual patterns. By preprocessing the data, designing an appropriate network architecture, and applying the backpropagation algorithm with the Adam optimizer, the model achieved high accuracy in identifying handwritten digits. This practical exercise reinforced the importance of key concepts such as data normalization, activation functions, and optimization methods in improving model performance. Additionally, the visualization of predictions provided clear evidence of the network's capacity to generalize and make accurate classifications on unseen data.

References:

- Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.
- Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.
- Keras. (2025). *MNIST dataset*. Retrieved from https://keras.io/api/datasets/mnist/
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. *Journal of Machine Learning Research*, 12, 2825–2830.