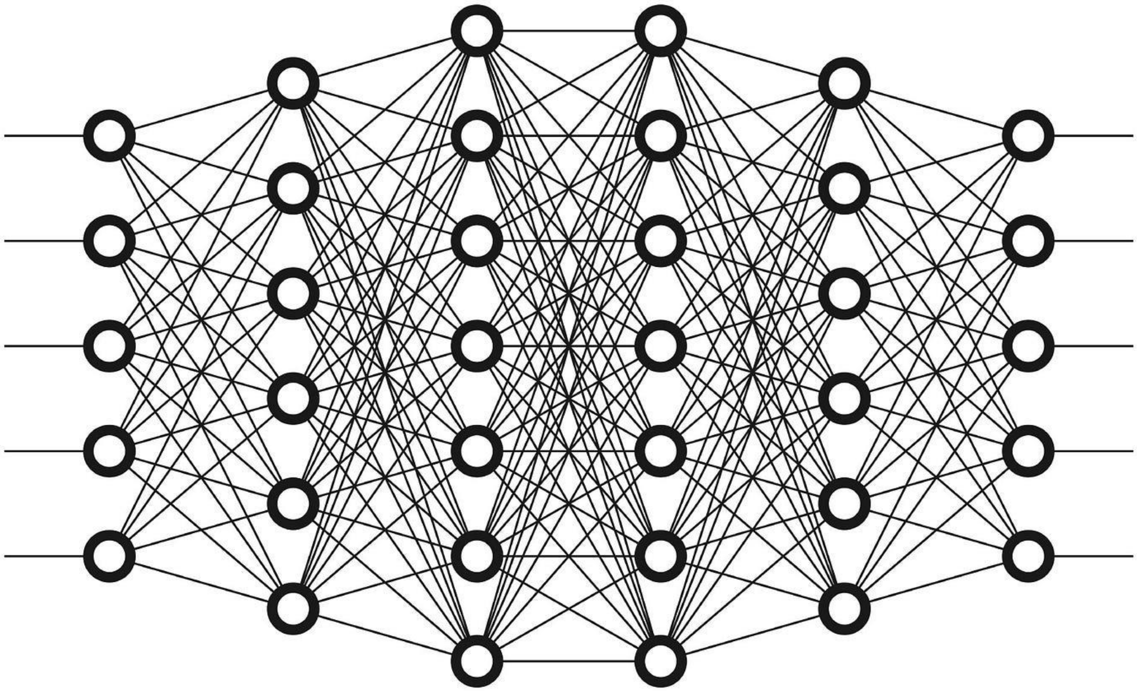

Project_2 Report

CIE 417

Neural Network Implementation &Analysis

UNDER SUPERVSION

DR. Khaled Mostafa

ENG. Marwa Monier

ENG. Gamal Zayed

Done by:

Mohamed Morshedy

20210037

1. Problem Definition and Motivation

Problem Statement:
The objective of this project is to implement and analyze neural network model on two distinct datasets: the Iris dataset for multiclass classification and the MNIST dataset for handwritten digit recognition. The primary motivation is to understand the performance of custom-built neural networks compared to established models (Scikit-learn and Keras) on both simple and complex datasets.

Importance:
Neural networks are widely used for various machine learning tasks, and understanding their behavior on different datasets is crucial for developing effective models. The project aims to assess the custom model's performance, identify its strengths and limitations, and provide insights for potential improvements.

2. Dataset

Iris Dataset:

- A classic dataset for multiclass classification.
- Contains measurements of sepal length, sepal width, petal length, and petal width for three species of iris flowers.
- Total Instances: 150 (50 per class)

MNIST Dataset:

- Commonly used for handwritten digit recognition.
- Consists of 28x28 pixel grayscale images of handwritten digits (0 to 9).
- Total Instances: 70,000 (60,000 for training, 10,000 for testing)

3. Approach and Methodology

3.1 Data Pre-processing

Iris Dataset:

- Standardized features using Scikit-learn's `StandardScaler`.
- One-hot encoded target variable for multiclass classification.

MNIST Dataset:

- Flattened and normalized pixel values (0-255) to a range of 0-1.
- One-hot encoded target labels.

3.2 Model Parameters

Custom Neural Network:

- Iris: 2 layers (8 neurons, sigmoid activation; output layer, sigmoid activation)
- MNIST: 2 layers (784 neurons, sigmoid activation; output layer, sigmoid activation)
- Training: 10,000 epochs for Iris, 1,000 epochs for MNIST
- Learning Rate: 0.01 for Iris, 0.001 for MNIST

Scikit-learn MLPClassifier:

- Iris: 1 hidden layer (100 neurons, 'relu' activation), 'adam' optimizer, 1,000 iterations

Keras Neural Network:

- MNIST: 2 layers (784 neurons, sigmoid activation; output layer, sigmoid activation)
- Training: 1,000 epochs, Learning Rate: 0.001

4. Implementation

4.1 Tools and Libraries

- NumPy: Used for numerical operations and array handling.
- Pandas: Employed for data manipulation and analysis.
- Scikit-learn: Utilized for data preprocessing, model training (MLPClassifier), and evaluation.
- Keras: Employed for building and training the custom neural network and Keras neural network.

4.2 Hyperparameter Tuning

- Custom Neural Network: Adjusted the number of layers, neurons, and activation functions to balance model complexity and performance.
- Scikit-learn MLPClassifier: Tuned the number of hidden layers, neurons, and training iterations.
- Keras Neural Network: Modified the architecture, batch size, and learning rate for optimal performance.

5. Conclusion

The custom neural network demonstrated excellent accuracy on the Iris dataset, matching the performance of established models. However, on the MNIST dataset, it showed limitations compared to the dedicated Keras model, highlighting the importance of specialized libraries for complex tasks.

Key Learnings:
- Importance of dataset-specific preprocessing.
- Model architecture plays a significant role in performance.
- Dedicated libraries offer advantages in terms of performance and ease of use.

Tips for Improvement:
- Experiment with different architectures and activation functions.
- Explore advanced neural network architectures (e.g., CNNs) for image classification tasks.
- Conduct more extensive hyperparameter tuning for optimal performance.

In conclusion, the project provided valuable insights into the behavior of neural networks on diverse datasets, emphasizing the need for tailored approaches based on dataset complexity.