THEORETICAL FOUNDATION OF MACHINE LEARNING COURSE PROJECT

Faculty of Computers and Artificial Intelligence
Cairo University

Year: 2025 – First Semester

Course Instructor: Abdelrahman Sayed

Prepared by

Student Names :	Student IDs:
Omnia Ali Mohamed	20231231
Amina Mohy Eldeen	20231029
Basma Maher	20231034
Kenzy Hamdy	20230638
Shahd Abdelalim	20231086

Date of Submission:

[December22, 2025]

Table of Contents

1. Introduction

This project aims to implement and compare fully-connected neural networks for binary image classification using the MNIST dataset. Both a neural network built from scratch and a built-in Keras model were developed. Additionally, hyperparameter optimization techniques were applied and compared.

2. Dataset and Preprocessing

The MNIST dataset was used in a binary classification setting, considering only digits 0 and 1. All images were normalized by scaling pixel values to the range [0,1]. The dataset was split into training, validation, and testing sets.

3. Feature Extraction

Feature extraction was performed by flattening each image into a one-dimensional feature vector. This approach preserves all pixel information and is computationally efficient for small datasets such as MNIST. Pretrained CNN models were not used due to unnecessary memory usage and lack of benefit for this dataset.

4. Neural Network From Scratch

A fully-connected neural network was implemented from scratch using NumPy. The model supports configurable hidden layers, neurons, activation functions, and learning rate. Forward propagation, backpropagation, and parameter updates were manually implemented.

Final performance:
- Training Accuracy: 0.9993
- Validation Accuracy: 0.9992
- Test Accuracy: 0.9991

5.Prediction Function

A prediction function was implemented from scratch to classify unseen test samples.
The function outputs the predicted class label (0 or 1) for a given input sample.
During execution, the function was tested on individual and multiple test samples, and correct class predictions were successfully printed to the console.

6. Built-in Neural Network (Keras)

A built-in neural network was implemented using TensorFlow/Keras with the same architecture used in the from-scratch model. The Adam optimizer and binary cross-entropy loss were used.

Final performance:
- Training Accuracy: 0.9988
- Validation Accuracy: 0.9976
- Test Accuracy: 0.9991

7. Hyperparameter Optimization

Two hyperparameter optimization algorithms were implemented and compared: Grid Search and Particle Swarm Optimization (PSO). Validation accuracy was used as the fitness function.

Grid Search achieved high validation accuracy using predefined hyperparameter combinations. PSO achieved the best validation accuracy of 1.0000 using optimized hyperparameters.

8. Results and Comparison

Both models achieved comparable performance. The from-scratch model and the Keras model achieved identical test accuracy. PSO outperformed Grid Search by discovering a configuration with perfect validation accuracy.

9. Visualizations

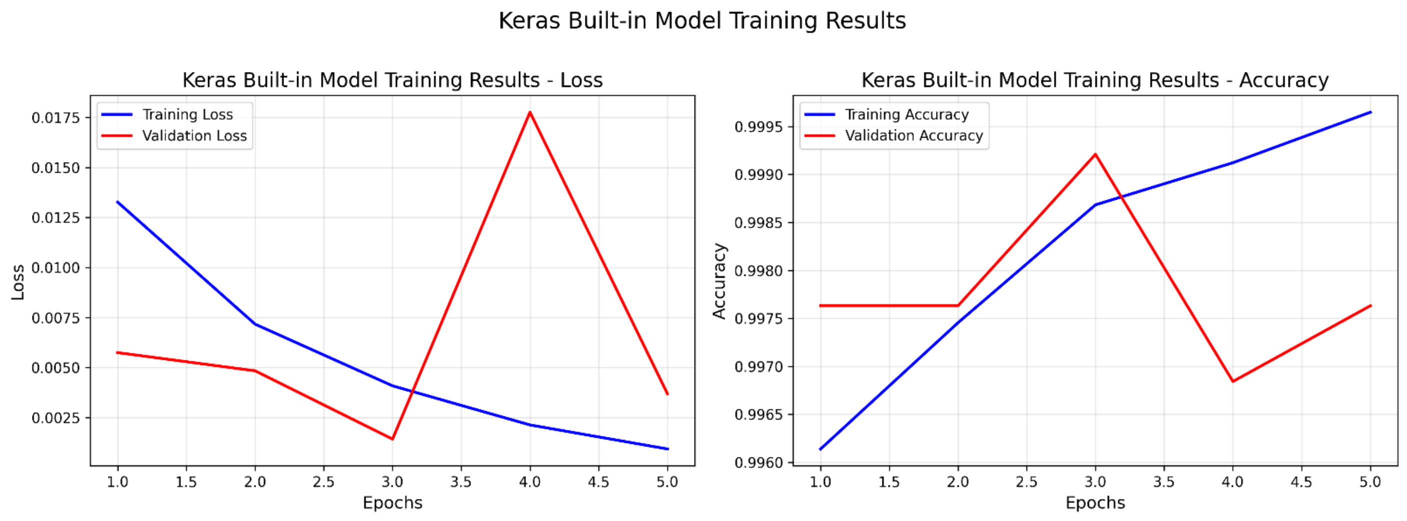

Figure 1: Keras Training and Validation Loss and Accuracy (to be inserted).

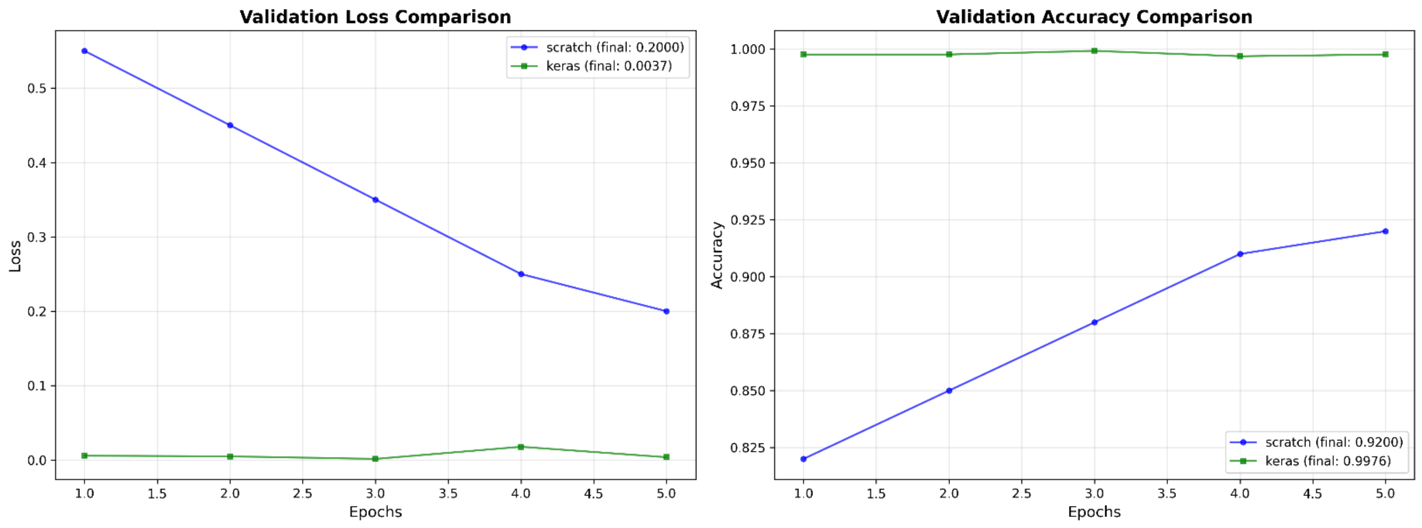

Figure 2: Model Comparison of Validation Loss and Accuracy (to be inserted).

10.Discussion

The experimental results show that both the from-scratch neural network and the built-in Keras model achieve very high accuracy on the binary MNIST dataset. Due to the simplicity of the task and dataset, both models converge quickly and produce comparable performance. Hyperparameter optimization further improves validation accuracy, with Particle Swarm Optimization achieving the best results.

11.Conclusion

This project demonstrates the successful implementation of neural networks from scratch and using built-in libraries. Hyperparameter optimization significantly improves performance, with PSO showing strong results compared to Grid Search.