**Week 3 Report Summer Internship**

Name: Mohamed Elfaki

University: Eastern Mediterranean University (Famagusta, Cyprus)

Department: Software Engineering

Training Organization: Cloud Solutions for IT & Communications Co, AI department

**Outline:**

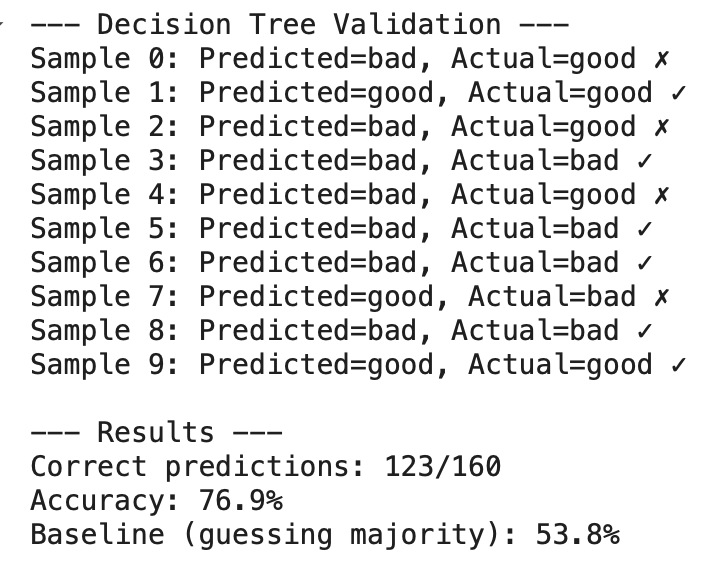
1. Introduction
2. Random Forest Implementation and results
3. Neural Network Fundamentals
4. Neural Network implementation
5. References
6. **Introduction**

For this week’s report, I’ll start by covering the random forest machine learning algorithm, its implementation from scratch and results. Then ill cover neural network fundamentals along with an implementation of neural networks. Then ill analyze the performance details of the neural networks compared to other methods.

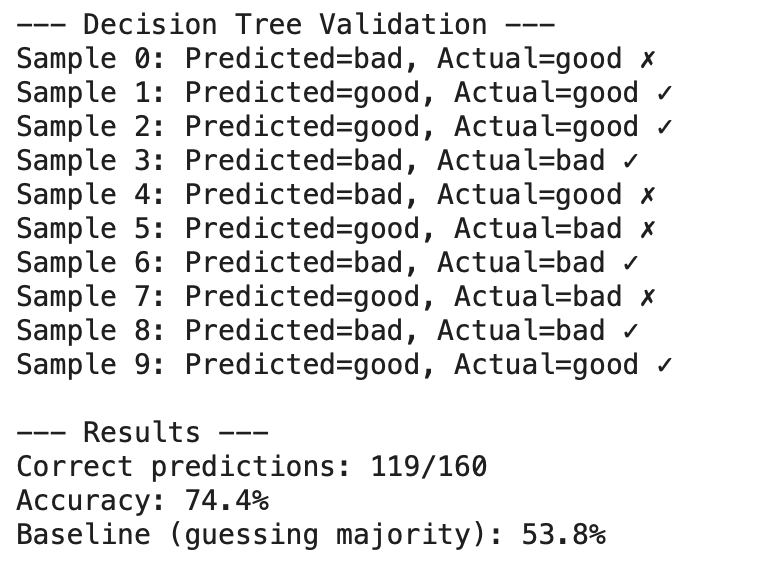
1. **Random forest implementation and results**

I’ve implemented random forest, which is built on my previous decision tree code, a random forest is basically a decision tree with many trees. All that’s changed from my previous implementation is that to arrive at a result we get the majority vote of many trees, these trees will receive a different portion of the training data, this improves accuracy.

For the first run I used maxdepth=8, and I used 3 trees, the results were good an improvement from the original 75% with decision tree to 76.9%.



The second run however was worse than the first run, though it consisted of 11 trees, the accuracy declined due to information loss.



1. **Neural Network Fundamentals**

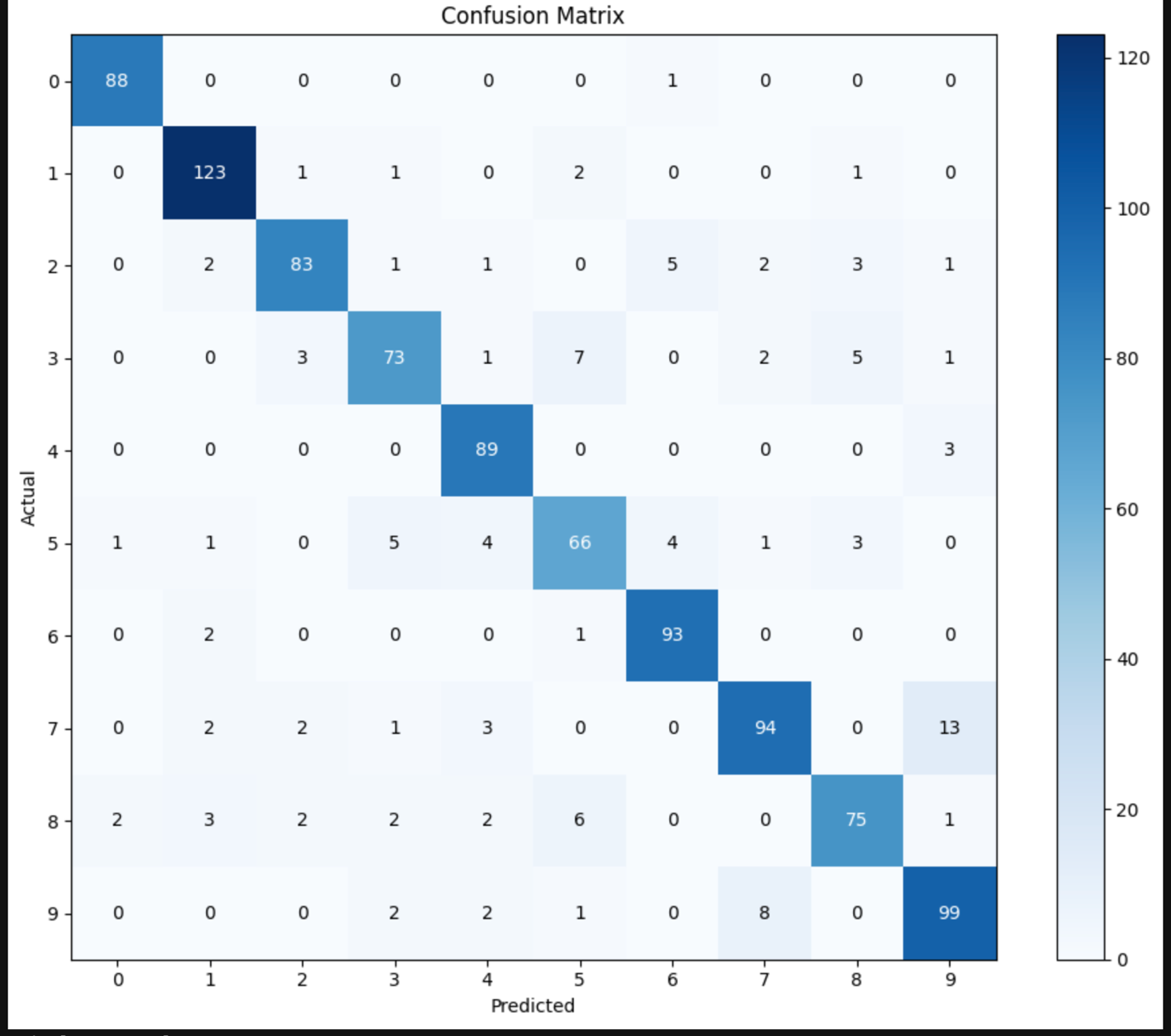
In a neural network there exist neurons (or nodes) each of which takes inputs and based on its weights, bias and activation function returns an output. These neurons are organized in layers starting with an input layer, several hidden layers and an output layer.

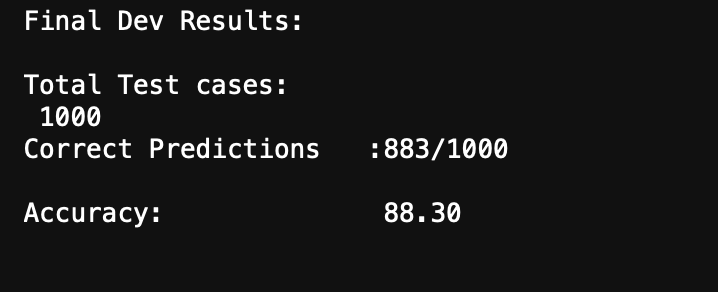
The network learns through the process of forward propagation, and backward propagation to calculate the error, followed by updating the parameters

1. **Neural Network Implementation**

I implemented a neural network that reads MNIST handwritten digits (0-9) from scratch using python. I used no libraries other than NumPy and pandas to maximize deep understanding of the topic. I used a 784 -> 10 -> 10 neuron architecture with 1 hidden layer and 3 total layers. Then I implemented forward and backward propagation which used many mathematical formulas along with a ReLU and softmax activation functions.

I managed to achieve 88.3% accuracy without any use of tensor flow and a relatively simple neural network architecture.





A number symbols in black squares

AI-generated content may be incorrect.

1. **References**

* <https://medium.com/coinmonks/the-mathematics-of-neural-network-60a112dd3e05>
* <https://www.ibm.com/think/topics/random-foresthttps://www.kaggle.com/datasets/hojjatk/mnist-dataset>
* https://aws.amazon.com/what-is/neural-network/