UTM
UNIVERSITI TEKNOLOGI MALAYSIA

MECS1033 Advanced Artificial Intelligence

Assignment 1 MLP Report

"MLP Code in Python Language"

Programme	:	Master of Computer Science ODL
Semester	:	20252026-1
Section	:	53
Report Title	:	MLP Code in Python Language
Student Name	:	Lau Su Hui (Abby)
Matric No.	:	MEC245045
Lecturer Name	:	Prof. Madya. Dr. Haza Nuzly bin Abdull Hamed

1.0 Introduction

- **Objective:** The primary objective of this assignment is to develop a Multi-Layer Perceptron (MLP) from scratch in Python language to solve the XOR classification problem.
- **Problem Statement:** The XOR is a classic and fundamental problem in artificial intelligence and machine learning because it is a non-linear problem (data points not linearly separable). This means that we cannot draw a single straight line to separate the outputs of 0s and 1s.
- **Hypothesis:** A basic single-layer network will fail to solve this problem. To solve it and capture or learn the non-linear relationship required for XOR, a "Hidden Layer" is added to the network architecture. This allows the network to understand more complex patterns.

2.0 Methodology

To address the problem, the following "machine" architecture and logic were implemented:

- **Architecture:** The network consists of a 2-4-1 topology, which are 2 Input Nodes, 4 Hidden Nodes, and 1 Output Node.
- **Activation Function:** The Sigmoid function was used to convert signals into a 0-1 probability range.
- **Learning Algorithm:** Gradient Descent with Backpropagation was utilised to auto-correct weights during training.
- **Language:** The code was written in Python using manual list-based matrix multiplication from scratch, without the use of NumPy or external libraries.
- **MLP Architecture Design:**

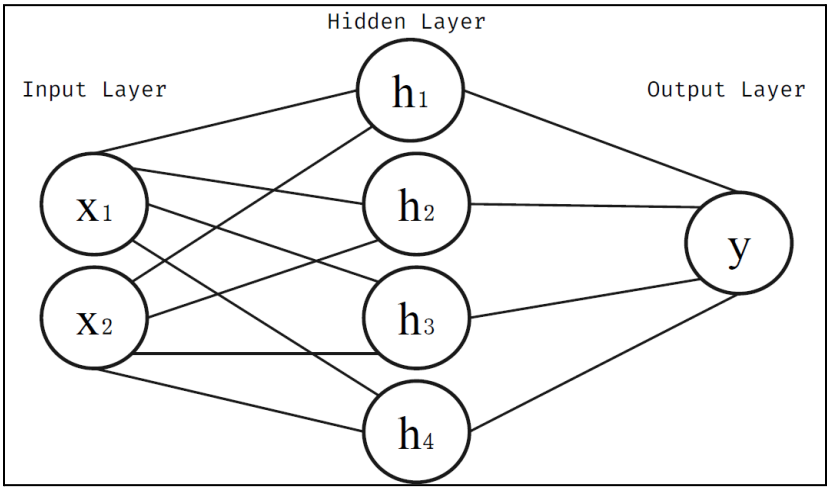

Figure 1.1 MLP Architecture for XOR

3.0 Results and Discussion

3.1 Initialisation (Pre-Training Check)

- **Architecture Setup:** Before training, the MLP (network) is initialised with a 2-4-1 topology (2 Inputs, 4 Hidden, 1 Output).
- **Random Weights:** Weights are randomised between -1 and 1 to break symmetry. This leads to unpredictable initial outputs.
- **Pre-Training Status:** As shown in the console and figure below, the initial predictions cluster around 0.51 - 0.58 to represent random guessing. These values are far from the expected 0 and 1 targets.
- **Assessment:** The confirms that the network currently possesses no logic and requires calibration via training.

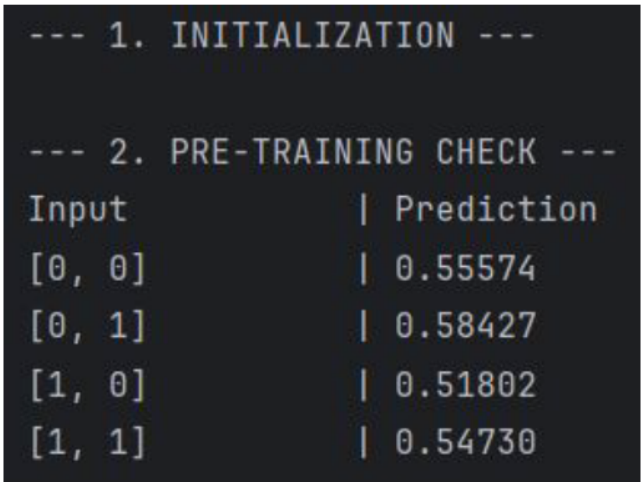

Figure 2.1 Result Screenshot 1 for Initialisation and Pre-Training Check

3.2 Training Phase

- **Learning Algorithm:** The network utilises Backpropagation to minimise the Mean Squared Error (MSE).
- **Training Duration:** The model underwent 10,000 epochs (iterations).
- **Observation:** The loss metric shows a consistent decline throughout the training phase.
    - **Start (Epoch 1):** Loss was 0.946456 (High Error), the model is guessing blindly
    - **End (Epoch 10000):** Loss dropped to 0.001180 (Near Zero Error), this steady decrease proves that the Backpropagation algorithm is successfully correcting the weights with every cycle.

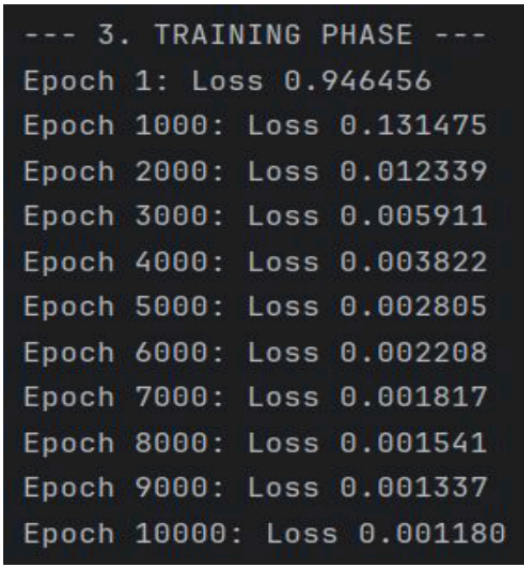

Figure 2.2 Result Screenshot 2 for Training Phase

3.3 Final Results

- **Logic Verified:** After training, the final predictions successfully mimic the Exclusive OR (XOR) truth table.
- **Clear Separation:** The model clearly distinguishes between the two classes.
    - Class 0 (Target 0): Inputs [0,0] and [1,1] resulted in Predicted Output ≈ 0.008 and 0.024
    - Class 1 (Target 1): Inputs [0,1] and [1,0] resulted in Predicted Output ≈ 0.98
- **Outcome:** The Multi-Layer Perceptron (MLP) has successfully learnt the non-linear decision boundary required to solve XOR.

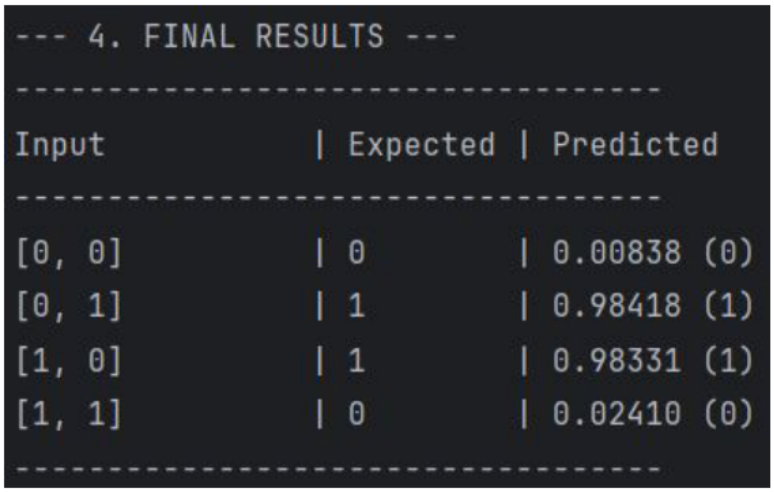

Figure 2.3 Result Screenshot 3 for Final Results

4.0 Conclusion

The implementation confirms that a 3-layer MLP (Input, Hidden, Output) can effectively solve non-linear problems like XOR. The inclusion of the "Hidden Layer" successfully transformed the input space to make the classes separable.

5.0 Appendix

1) Source Code: UTM MECS1033 AAI Assignment 1 MLP Code (No NumPy).
2) Slides: UTM MECS1033 AAI Assignment 1 MLP Slides (No NumPy)