Laboratory Report 4: Backpropagation Manual Analysis

Course: COMP-341L – Artificial Neural Networks Lab
Lab Assignment: 4
Topic: Backpropagation and Its Implementation from Scratch
Submission Date: February 10, 2026

Student Name: Zarmeena Jawad
Roll Number: B23F0115AI125
Section: AI Red

Academic Integrity

I declare that this report is my own work. I have not copied from other students or submitted another person's report. As per course policy, plagiarism or copy-pasting leads to zero marks for everyone involved.

1 Executive Summary

This laboratory involves tracing a single iteration of the backpropagation algorithm by hand on a small feedforward network. The work includes computing the forward pass to obtain a prediction, measuring the output error, computing responsibility terms (deltas) for the output and hidden units, and applying the weight-update rule. The report documents each step with the same network parameters as the lab manual and reflects on how error is distributed backward and how the learning rate affects training.

2 Learning Objectives

Upon completion of this lab, students should be able to:

- Compute weighted sums and sigmoid activations
- Compute and interpret output error
- Derive output and hidden deltas
- Apply the weight-update rule
- Explain backward propagation and learning rate effects

3 Methodology

Calculations follow Lab 04. All values were computed manually and verified using `backprop_trace.py`. Screenshots of the script output are included for each task.

4 Network Specification (Lab 04.pdf)

Inputs: $x_1 = 0.35$, $x_2 = 0.7$
Target: 0.5 Learning rate: $\eta = 1$ Activation: Sigmoid

5 Task 1: Forward Pass Analysis

$$a_1 = 0.21, \quad a_2 = 0.315$$

$$y_3 = 0.552, \quad y_4 = 0.578, \quad y_5 = 0.665$$

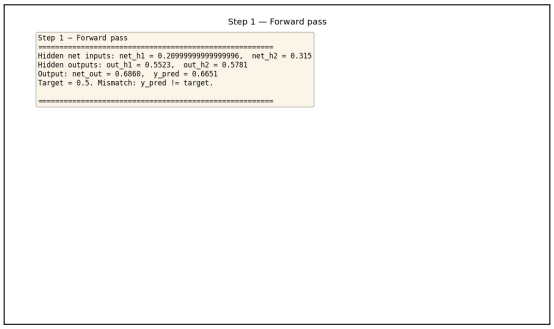

Figure 1: Task 1 – Forward Pass

6 Task 2: Error Calculation

$$\text{Error} = 0.5 - 0.665 = -0.165$$

```
err = target - y_pred
```

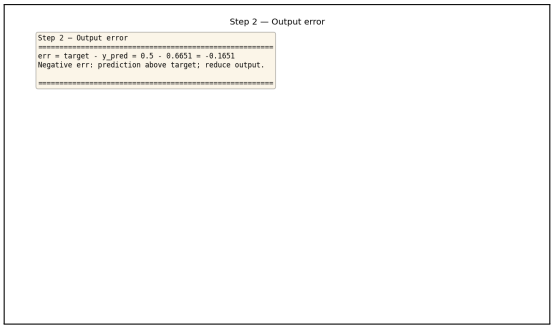

Figure 2: Task 2 – Error Calculation

7 Task 3: Output Neuron Responsibility

$$\delta_5 \approx -0.037$$

```
d_out = y_pred * (1 - y_pred) * (target - y_pred)
```

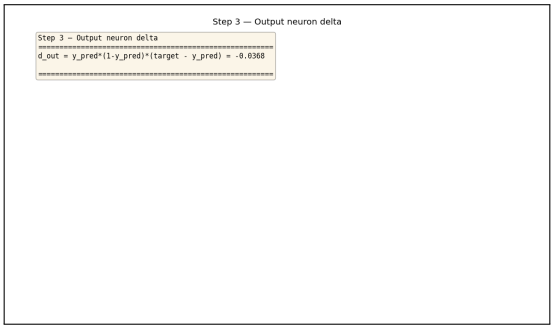

Figure 3: Task 3 – Output Delta

8 Task 4: Hidden Neuron Responsibility

$$\delta_3 \approx -0.0027, \quad \delta_4 \approx -0.0081$$

```
d_h1 = out_h1 * (1 - out_h1) * (w13 * d_out)
d_h2 = out_h2 * (1 - out_h2) * (w23 * d_out)
```

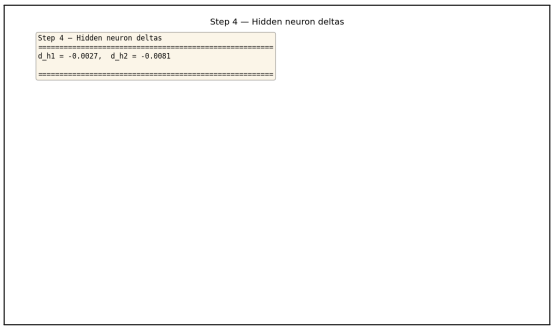

Figure 4: Task 4 – Hidden Deltas

9 Task 5: Weight Updates

```
d_w13 = eta * d_out * out_h1
d_w23 = eta * d_out * out_h2
```

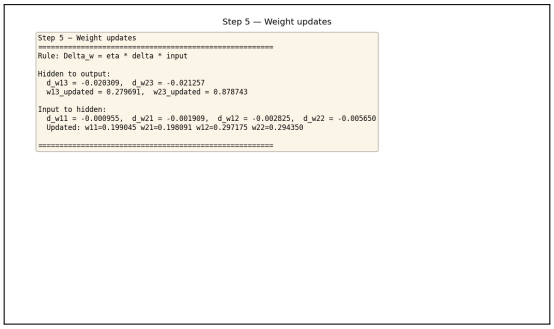

Figure 5: Task 5 – Weight Updates

10 Task 6: Interpretation and Reflection

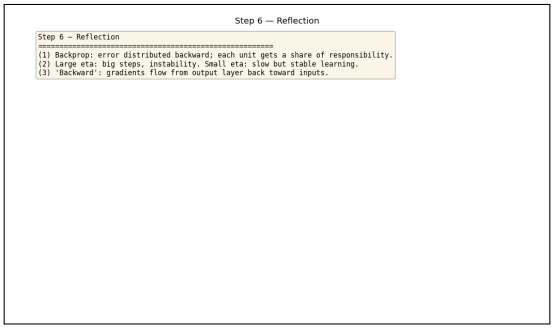

Figure 6: Task 6 – Reflection

11 Conclusion

This lab demonstrated backpropagation as the chain rule applied backward. Manually computing each step clarified responsibility assignment and the effect of learning rate.

12 References

Lab 04 Manual (Lab 04.pdf), pages 16–18.