Laboratory Report 2
Fundamentals of Artificial Neural Networks

Course: COMP-341L – Artificial Neural Networks Lab

Lab Assignment: 2

Topic: Understanding ANN from Scratch

Submission Date: January 26, 2026

Student Name: Zarmeena Jawad

Roll Number: B23F0115AI125

Section: AI Red

1 Executive Summary

This laboratory assignment investigates the foundational principles governing artificial neural networks through practical experimentation. The work focuses on constructing individual neurons, examining their response behavior, identifying architectural limitations, and implementing probability-based output transformations.

All implementations were carried out using Python with NumPy for numerical computation and Matplotlib for visualization, following the specifications outlined in the laboratory manual.

2 Learning Objectives

Upon completion of this laboratory assignment, the following learning outcomes were achieved:

- Construction of artificial neurons with activation mechanisms
- Analysis of activation function behavior on identical inputs
- Understanding the influence of bias on neuron sensitivity
- Identification of limitations in single-layer models
- Implementation of multi-layer architectures
- Application of softmax for probability distribution generation

3 Methodology

All experiments were implemented using Python 3. Numerical operations were performed using the NumPy library, while graphical illustrations were generated using Matplotlib.

A structured experimental approach was followed: baseline models were implemented first, after which parameters were systematically varied to observe changes in neuron behavior.

4 Task 1: Constructing a Decision-Making Neuron

4.1 Purpose

The objective of this task was to design a single artificial neuron capable of evaluating emotional stress from acoustic speech features and to analyze how different activation functions interpret identical evidence.

4.2 Feature Selection

Three acoustic features were selected:

- Vocal Tempo (normalized 0–1)
- Frequency Instability (normalized 0–1)
- Silence Intervals (normalized 0–1)

Simulated input values:

$$x = [0.78,\ 0.65,\ 0.30]$$

4.3 Weight Assignment

$$w = [0.35,\ 0.55,\ 0.10]$$

These values reflect the relative importance of each feature.

4.4 Threshold Configuration

A bias value of -0.25 was used to ensure moderate conservatism in neuron activation.

4.5 Activation Functions

The integrated signal is computed as:

$$z = w \cdot x + b$$

The following activation functions were implemented:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{2}$$

$$\mathrm{ReLU}(z) = \max(0, z) \tag{3}$$

4.6 Results

- Integrated Signal: $z = 0.460$
- Logistic Output: 0.6131
- Tanh Output: 0.4298
- ReLU Output: 0.4600

4.7 Discussion

Each activation function transforms the same signal differently. Logistic produces probabilistic confidence, tanh produces signed activation, and ReLU produces raw activation strength.

5 Task 2: Sensitivity Analysis Through Threshold Variation

5.1 Purpose

This experiment examined how bias controls neuron sensitivity by varying threshold values while keeping inputs and weights fixed.

5.2 Experimental Setup

- Inputs: $[0.78, 0.65, 0.30]$
- Weights: $[0.35, 0.55, 0.10]$
- Bias range: $[-3.5, +3.5]$

5.3 Observations

- Logistic activation increases gradually
- ReLU activates abruptly at zero
- Tanh transitions smoothly through a neutral zone

5.4 Key Insight

Bias determines *when* activation occurs, while the activation function determines *how* activation evolves.

6 Task 3: Limitations of Single Neurons and Multi-Layer Solutions

6.1 Single Neuron Limitation

A single neuron produced overlapping outputs for relaxed-fast and anxious-slow speech patterns due to its linear decision boundary.

6.2 Layered Network Architecture

A two-neuron hidden layer was introduced to extract intermediate representations before final classification.

6.3 Forward Pass Implementation

```
def layered_network_forward(input_features):
    z1 = np.dot(input_features, W_h1) + b_h1
    z2 = np.dot(input_features, W_h2) + b_h2

    a1 = logistic(z1)
    a2 = logistic(z2)

    z_out = np.dot([a1, a2], W_output) + b_output
```

```
    return logistic(z_out)
```

6.4 Analysis

Layered architectures enable hierarchical feature learning and non-linear decision boundaries, allowing complex pattern separation.

7 Task 4: Softmax for Probability Distribution Generation

7.1 Softmax Equation

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

7.2 Results

- Relaxed: 0.5889
- Anxious: 0.2925
- Stressed: 0.1186

7.3 Interpretation

Softmax distributes probability across all classes, ensuring values lie within $[0, 1]$ and sum to one.

8 Results Summary

- Activation functions exhibit distinct behaviors
- Bias strongly influences neuron sensitivity
- Single neurons cannot model non-linear boundaries
- Multi-layer networks enable hierarchical learning
- Softmax produces interpretable probability distributions

9 Conclusion

This laboratory assignment provided practical insight into the internal mechanics of artificial neural networks. Through implementation from first principles, it became evident that neurons act as decision-making components governed by activation dynamics and sensitivity thresholds.

The experiments demonstrated the necessity of non-linearity, depth, and probabilistic output modeling in modern neural architectures. These concepts form the foundation for advanced deep learning systems.

10 References

1. Lab Manual: Lab 02 – Understanding ANN from Scratch
2. NumPy Documentation: `https://numpy.org/doc/stable/`
3. Matplotlib Documentation: `https://matplotlib.org/stable/contents.html`

Report Prepared By: Zarmeena Jawad
Roll Number: B23F0115AI125
Section: AI Red

Date: January 26, 2026