Theoretical Analysis and Experimental Breakdown: CNNs and BiLSTMs

Devesh Singh Chauhan

February 12, 2026

1 Overview

This document provides a detailed theoretical explanation of the models implemented in Lab Assignment 4 (AI 354). The assignment bridges the gap between classical machine learning and deep learning by implementing two core architectures:

1. **Convolutional Neural Networks (CNNs)** for Computer Vision.
2. **Bidirectional LSTMs (BiLSTMs)** for Natural Language Processing (NLP).

2 Part 1: Convolutional Neural Networks (CNN)

2.1 Why CNNs over MLPs?

In Lab 3, we used a Multi-Layer Perceptron (MLP) which required flattening a 2D image ($H \times W$) into a 1D vector ($1 \times N$).

- **The Flaw:** Flattening destroys spatial locality. The network loses the information that pixel $(0, 0)$ is adjacent to $(0, 1)$.
- **The Solution:** CNNs process the image as a grid, preserving spatial relationships.

2.2 Mathematical Components

The CNN implemented uses three key operations:

2.2.1 1. Convolution ($*$)

A kernel (filter) K of size 3×3 slides over the input image I. The output feature map S is calculated as:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n)$$

This operation allows the network to detect features like vertical edges, horizontal lines, or corners regardless of their position in the image (Translation Invariance).

2.2.2 2. ReLU Activation

To introduce non-linearity, we apply the Rectified Linear Unit:

$$f(x) = \max(0, x)$$

This mimics the firing rate of biological neurons and solves the vanishing gradient problem common in Sigmoid functions.

2.2.3 3. Max Pooling

Pooling reduces the spatial dimensions (e.g., $64 \times 64 \rightarrow 32 \times 32$) to reduce computation and prevent overfitting. It selects the dominant feature in a region:

$$P(i, j) = \max_{(k,l) \in Region} S(k, l)$$

2.3 Experimental Analysis: Depth Matters

We compared a **Shallow CNN (1 Layer)** vs. a **Standard CNN (2 Layers)**.

- **Shallow (F1: 0.33):** The single layer could only learn primitive features (edges). It lacked the capacity to combine these edges into shapes.
- **Standard (F1: 0.43):** The second layer takes the "edges" from Layer 1 and combines them to recognize "shapes" (e.g., a collar, a sleeve). This hierarchical learning is why Deep Learning outperforms shallow networks.

3 Part 2: Bidirectional LSTMs for NLP

3.1 The Challenge of Sequential Data

Unlike images, text is sequential. The meaning of a word depends on the words before *and* after it.

"The movie was not good, it was actually great."

A standard feed-forward network sees "not good" and assigns a negative sentiment. It fails to capture the long-range dependency of "actually great."

3.2 LSTM Architecture

Long Short-Term Memory (LSTM) networks solve the "vanishing gradient" problem of standard RNNs using three gates:

1. **Forget Gate (f_t):** Decides what information to discard from the cell state.
2. **Input Gate (i_t):** Decides which new values to update.
3. **Output Gate (o_t):** Decides what to output based on the cell state.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

3.3 Why Bidirectional?

Our model uses a **BiLSTM**, which consists of two LSTMs:

- Forward LSTM: Reads sentence $x_1 \to x_T$
- Backward LSTM: Reads sentence $x_T \to x_1$

The final output y combines both contexts:

$$y_t = [\vec{h_t}; \overleftarrow{h_t}]$$

This allows the model to understand that "good" is modified by the preceding "not" and the succeeding "great."

3.4 Regularization and Robustness

3.4.1 Dropout Analysis

We tested Dropout rates of 0.2, 0.4, and 0.6.

- **Dropout:** Randomly zeros out neurons during training with probability p. This prevents the model from relying on specific keywords (e.g., always predicting "Positive" if it sees "movie").
- **Finding:** Lower dropout (0.2) worked best (F1: 0.84). High dropout (0.6) removed too much information, causing underfitting (High Bias).

3.4.2 Noise Robustness (Responsible AI)

We injected noise (random word replacements) into the test data to simulate real-world errors (typos, slang).

- **Clean F1:** 0.8412
- **Noisy F1:** 0.7967

The small drop ($\approx 4.5\%$) indicates the model is **Robust**. It relies on the *semantic context* of the entire sentence rather than memorizing fragile keyword patterns.