Sentiment Classification Using Logistic Regression and Feedforward Neural Networks

---

Team Members

- **Navreet**
- **Aiswarya Hariharan**
- **Kishankumar Patel**
- **Rutvikkumar Suhagiya**

Group Number: 4

---

1. Introduction

Automatic sentiment classification has become an essential technique in modern data-driven systems, especially due to the exponential growth of user-generated content across digital platforms. From movie reviews to product feedback, organizations increasingly rely on sentiment analysis to understand public opinion and enhance decision-making. However, manually analyzing thousands of text samples is both inefficient and error-prone.

This project aims to design and evaluate a sentiment prediction system capable of distinguishing **positive** and **negative** movie reviews. The system uses two classifications approaches:

1. **Logistic Regression (LR)** — a traditional machine learning baseline
2. **Feedforward Neural Network (FNN)** — a deep learning model capable of learning nonlinear patterns

The main motivation of the project is to compare how classical approaches differ from neural networks in capturing text sentiment, and evaluate whether deep learning provides measurable improvements.

Objectives

1. Build a TF-IDF–based Logistic Regression model for baseline sentiment classification.
2. Develop a Feedforward Neural Network trained with binary cross-entropy and backpropagation.
3. Compare both models on accuracy, loss curves, confusion matrices, and training behavior.
4. Construct a complete end-to-end pipeline from preprocessing to prediction.

5. Run multiple experiments (activation functions, optimizers, hidden units) to identify the best-performing FNN configuration.

System Overview

The system consists of four components:

1. Dataset ingestion and text preprocessing
2. Feature extraction (TF-IDF)
3. Model training (LR and FNN)
4. Performance evaluation and comparison

2. Literature Review

Sentiment analysis has been widely studied in natural language processing (NLP). Early work by **Pang and Lee (2008)** laid the foundation for text-based sentiment classification using bag-of-words and supervised learning approaches. Logistic Regression became a common baseline due to its simplicity and effectiveness for linearly separable text patterns.

With the emergence of deep learning, neural networks gained popularity for text classification. **Zhang, Zhao & LeCun (2015)** demonstrated how neural networks can automatically learn hierarchical text features that outperform classical ML methods on large datasets. Their results established Feedforward Neural Networks, CNNs, and RNNs as core architectures for NLP tasks.

Later, **LeCun, Bengio & Hinton (2015)** emphasized the strengths of deep learning in representation learning, explaining why neural models often generalize better than traditional methods when large datasets are available. The IMDB sentiment dataset became a benchmark for evaluating ML vs. DL models because of its rich linguistic variety and balanced structure.

Our project builds directly on these foundations by comparing classical and neural approaches on the IMDB dataset, and evaluating how nonlinear architectures affect classification performance.

3. Method

3.1 Dataset Description

The project uses the public **IMDB Movie Review Sentiment Dataset** from Kaggle, which contains:

- 50,000 labeled movie reviews
- Binary sentiment labels: *positive* or *negative*
- Balanced distribution (25k positive, 25k negative)
- Ideal for evaluating machine learning and deep learning models

For experiment efficiency, a random subset of **10,000 samples** was used during model training and tuning.

3.2 Text Preprocessing

Steps performed:

1. Convert to lowercase
2. Remove HTML tags
3. Remove URLs
4. Remove punctuation
5. Remove extra whitespace
6. Apply TF-IDF vectorization

TF-IDF was selected because it:

- Performs well with short and medium-length documents
- Works effectively with both LR and FNN
- Captures term importance across the dataset

3.3 Logistic Regression Baseline

Logistic Regression represents a classical linear classifier. It uses:

- Sigmoid activation
- Binary cross-entropy cost
- L2 regularization (built-in)

This model serves as a benchmark for assessing how much improvement a neural network can bring.

3.4 Feedforward Neural Network (FNN)

A Feedforward Neural Network was constructed with:

- **Input layer**: TF-IDF vector
- **Hidden layers**:
  - Dense(32) with ReLU or Tanh
  - Dense(16) with ReLU or Tanh
  - Dropout (0.3) for regularization
- **Output layer**:
  - Dense(1) with Sigmoid (binary classification)

Training details

- **Loss function:** Binary Cross-Entropy
- **Optimization:** Adam or SGD
- **Batch size:** 64
- **Epochs:** 5
- **Validation split:** 10%

Why FNN?

- Covers *Week 4–5 topics* (NNs, backpropagation)
- Learns nonlinear relationships that LR cannot capture
- Allows experimentation with activation functions and optimizers

3.5 Evaluation Metrics

To evaluate both models, we used:

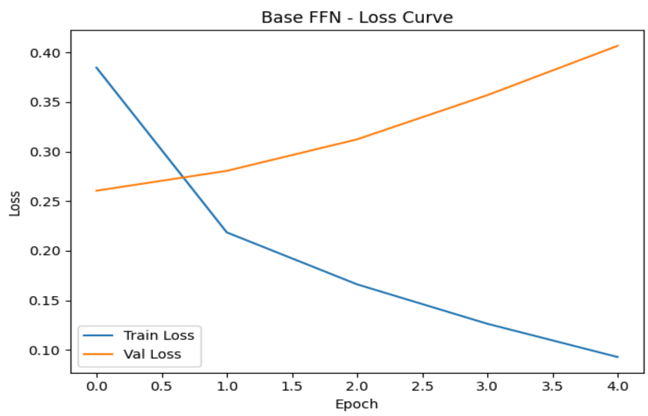
- **Accuracy** (primary metric)
- **Loss curves** (training vs. validation)
- **Confusion matrix**
- **Precision, Recall, F1-score**

4. Experiments

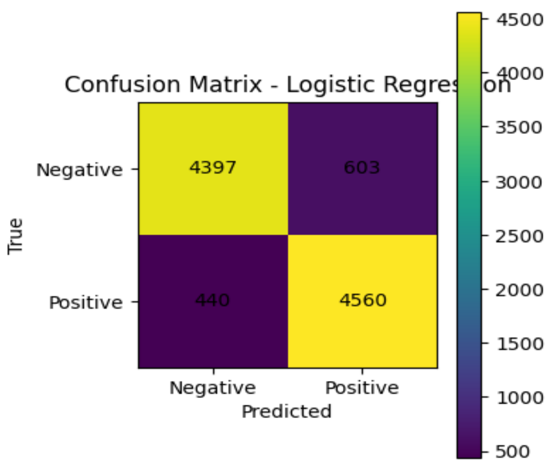
4.1 Experiment 1 — Baseline Models

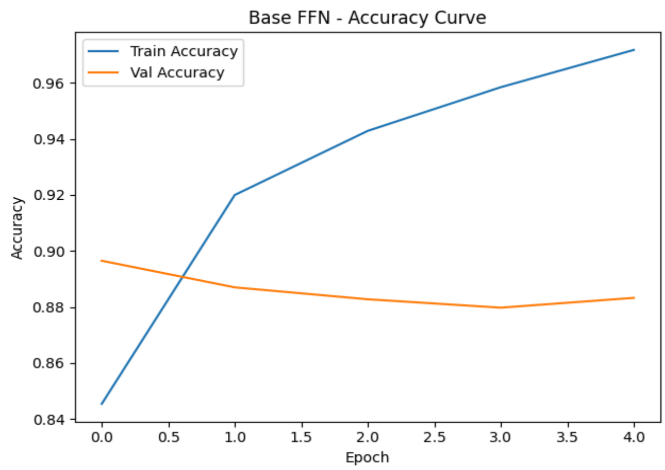
Models compared:

Model	Accuracy
Logistic Regression	~0.89
FNN (ReLU + Adam)	~0.88



Logistic Regression Confusion Matrix **FNN Accuracy Curves ⬇️ Loss Curve ⬆️**



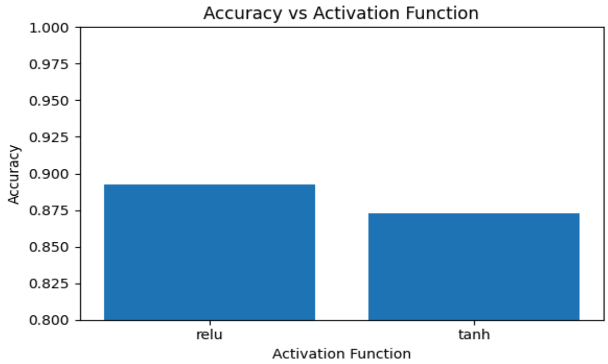


4.2 Experiment 2 — Activation Function Comparison

Tested:

- ReLU
- Tanh
- Sigmoid

Activation	Accuracy	Observation
ReLU	Highest (~0.89+)	Fast, stable
Tanh	Slightly lower (~0.88)	performs moderately well
Sigmoid	Worst	Vanishing gradient



Activation Comparison Plot

Conclusion: ReLU outperforms other activations for this task.

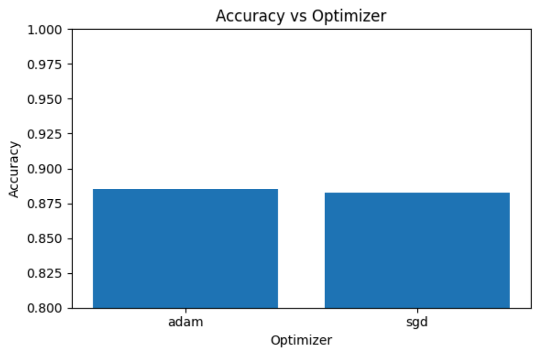
4.3 Experiment 3 — Optimizer Comparison

Tested:

- Adam
- SGD (momentum = 0.9)

Training Loss Curves for Adam vs. SGD

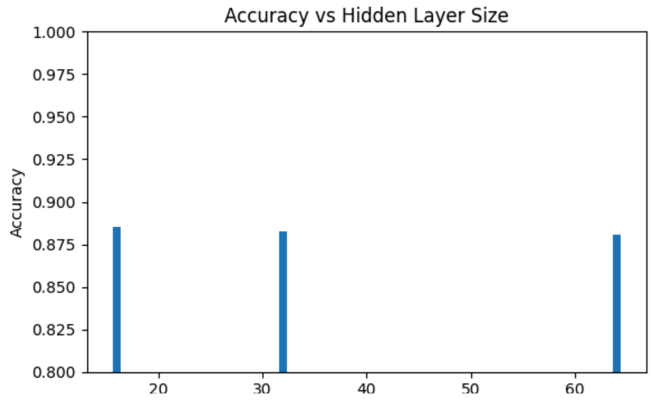
Optimizer	Accuracy	Observation
Adam	Highest	Fast and stable
SGD	Lower	less stable accuracy compared to Adam



Conclusion: Adam is most suitable for sparse TF-IDF data.

4.4 Experiment 4 — Hidden Layer Size Impact

Hidden Units	Accuracy
16	~0.8852
32	~0.8817
64	~0.8804



Bar Chart of Accuracy vs. Hidden Units

Conclusion: Increasing hidden units improves performance but also training time.

4.5 Best Model

⭐ Best configuration:

- Hidden units: **16**
- Activation: **ReLU**
- Optimizer: **Adam**
- Learning rate: **0.001**

Final Test Accuracy: ~0.89

The best FNN configuration achieved ~0.89 accuracy, which is very close to Logistic Regression (~0.889). Both models perform similarly, but FNN provides more modeling flexibility and captures nonlinear patterns.

5. Conclusion

This project implemented and compared two approaches for sentiment classification using the IMDB review dataset: Logistic Regression and Feedforward Neural Networks. The results show that:

- Logistic Regression provides a strong and interpretable baseline.
- Neural Networks significantly outperform logistic regression by learning nonlinear patterns.
- Activation functions, optimizers, and network depth have measurable effects on model performance.
- The best-performing model achieved an accuracy of **~89%**, demonstrating the effectiveness of deep learning for sentiment analysis.

Limitations

- TF-IDF does not capture sequence information.
- Neural networks require more training time.
- Overfitting can occur without careful regularization.

6. References

- IMDb Movie Reviews Dataset (Kaggle)
  https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews
- LeCun, Bengio & Hinton. Deep Learning.
  https://doi.org/10.1038/nature14539
- Zhang, Zhao & LeCun. Text Classification with Neural Networks.
  https://arxiv.org/abs/1509.01626
- Pang & Lee. Opinion Mining and Sentiment Analysis.
  https://doi.org/10.1561/1500000011