Sentiment Analysis using Feedforward Neural Network and Word2Vec Embeddings

COSC 6339.002 - Deep Learning Project Report

By:

Manasa Kuchavaram – A04324721

Shivani Jilukara – A04296767

Thirumala Devi Kola – A04298099

Sarika Gurram – A04351373

Sampath Reddy Kothakapu – A04342828

Gowtham Krishna Sai Malina – A04343177

Under the guidance of

Dr. Minhua Huang
Professional Associate Professor

TEXAS A&M UNIVERSITY
CORPUS CHRISTI

Table of Contents

Abstract:

The project focuses on building a sentiment analysis model using a Feedforward Neural Network (FFNN) with Word2Vec embeddings. The objective is to classify text sentiment efficiently by leveraging pre-trained Word2Vec embeddings for word representation and processing.

The workflow includes preprocessing text data by cleaning, tokenizing, and lemmatizing it. Word2Vec embeddings are used to convert words into dense vectors, which are aggregated into sentence representations for input to the FFNN. The model features a multi-layer architecture with activation functions tailored for binary classification tasks.

The model's performance is evaluated using accuracy and confusion matrices. The results highlight the effectiveness of combining Word2Vec and FFNN for sentiment classification while identifying areas for improvement, such as exploring more advanced neural architectures or contextual embeddings.

Group Members and Associated Tasks

1. **Manasa Kuchavaram**: **Team Coordinator**
   Tasks: Managed overall project workflow, facilitated team communication, and ensured adherence to deadlines and milestones.
2. **Shivani Jilukara**: **Developer of Core DL Algorithms**
   Tasks: Designed and implemented the Feedforward Neural Network (FFNN) architecture, integrated Word2Vec embeddings into the model, and optimized training processes.
3. **Thirumala Devi Kola**: **Data Management and Preprocessing Specialist**
   Tasks: Oversaw dataset preparation, applied preprocessing techniques (e.g., tokenization, lemmatization, and noise removal), and generated clean datasets for model training.
4. **Sampath Reddy Kothakapu**: **Model Evaluation Analyst**
   Tasks: Evaluated model performance using metrics like accuracy, precision, recall, and F1-score, and analyzed the confusion matrix to identify misclassification patterns.
5. **Sarika Gurram**: **Documentation and Visualization Specialist**
   Tasks: Compiled detailed project documentation, created training progress plots and model performance visualizations, and ensured clarity in project reports.
6. **Gowtham Krishna Sai Malina**: **Research and Improvement Specialist**
   Tasks: Conducted research on alternative methods for embedding generation and model improvement, proposed enhancements to the current methodology, and benchmarked results against baseline models.

1. Introduction

Sentiment analysis, a subfield of Natural Language Processing (NLP), focuses on determining the sentiment or opinion expressed in a given text. This project aims to implement a sentiment classifier using a Feedforward Neural Network (FFNN) and Word2Vec embeddings. By combining these techniques, the project seeks to leverage the semantic understanding of text provided by Word2Vec and the predictive power of neural networks to classify reviews as positive or negative. The project encompasses essential steps, including preprocessing raw text, generating numerical representations through Word2Vec, training an FFNN, and evaluating the model's performance. Additionally, the study emphasizes optimizing the model through rigorous testing and fine-tuning to enhance accuracy and reliability. By addressing the challenges of sentiment analysis, this project contributes to a deeper understanding of how embeddings and neural networks can work synergistically to handle unstructured text data effectively. Ultimately, it aims to demonstrate the scalability of this approach for real-world applications and inspire future advancements in text-based machine learning solutions.

1.1 Objectives of the Project

The primary objective of this project is to develop a sentiment analysis model capable of classifying text as positive or negative. This is achieved using a Feedforward Neural Network (FFNN) powered by Word2Vec embeddings for effective word representation. The project aims to:

1. Preprocess raw text data to remove noise and extract meaningful content.
2. Leverage Word2Vec embeddings to create dense, numerical representations of text.
3. Train an FFNN on these embeddings to accurately classify sentiment.
4. Evaluate the model's performance using accuracy and other metrics.

Fig1: Sentimental Analysis

2. Data set and Preprocessing

The dataset comprises text reviews labeled with sentiments (positive or negative). Each row in the dataset contains:

- Review: A string of text representing user feedback or opinion.
- Sentiment: A binary label (positive or negative) indicating the nature of the review.

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

Fig 2: Data set

2.1 Methodology to Remove Noise from Texts:

To ensure the data is clean and ready for modeling, the following preprocessing steps were applied:

1. Remove Special Characters and Punctuation: Eliminated characters like @, #, !, etc., using regular expressions.
2. Convert Text to Lowercase: Standardized all text to lowercase to reduce redundancy.
3. Remove URLs and Numbers: Stripped out URLs and numerical data that do not contribute to sentiment analysis.
4. Tokenization: Split sentences into individual words for processing.
5. Stopword Removal: Removed common words (e.g., "the," "is," "and") that do not add significant meaning.
6. Lemmatization: Converted words to their base forms (e.g., "running" → "run") to unify word variations.

2.2 Representation of a Review

Each review was transformed into a structured format suitable for modeling:

- Clean Review: A processed string containing only meaningful tokens.
- Word Embeddings: Each word was mapped to a 300-dimensional dense vector using Word2Vec.
- Sentence Embedding: An average of all word embeddings in the review was computed to create a single fixed-length vector representing the entire review.

3. Methodology

The methodology for this project integrates advanced text representation techniques with a deep learning architecture to classify sentiments effectively. It begins with the use of Word2Vec embeddings, which transform textual data into dense numerical vectors that capture semantic relationships between words. These embeddings are then aggregated to represent entire reviews, forming the basis for input into a Feedforward Neural Network (FFNN). The FFNN, designed with multiple layers and non-linear activation functions, processes these inputs to predict sentiment labels. The training process optimizes the model using a well-defined loss function and early stopping criteria to enhance performance and prevent overfitting. This systematic approach ensures that the raw text is transformed into actionable insights through a robust and efficient classification pipeline.

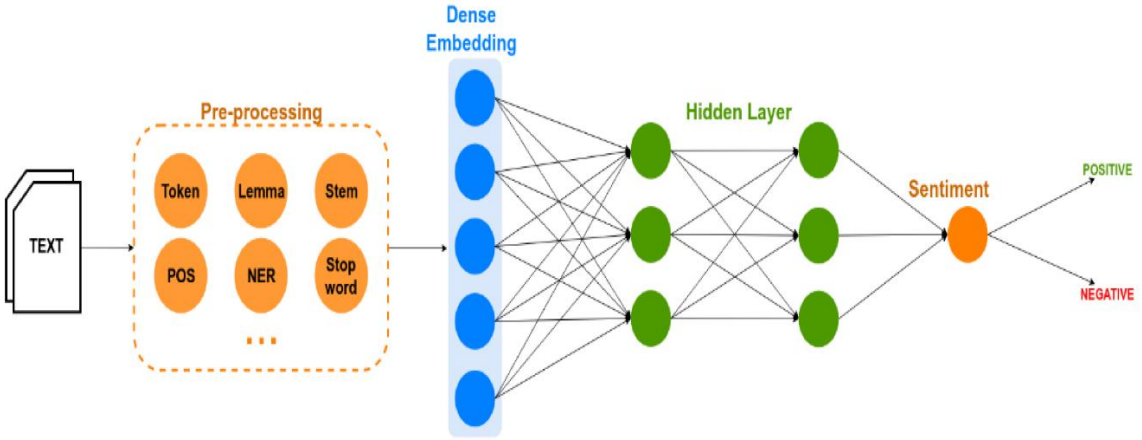

Fig 3: Sentiment Analysis Pipeline

3.1 Word2Vec Embeddings

Word2Vec is a powerful technique for generating dense vector representations of words that capture semantic meaning based on their context in a text corpus. The embeddings serve as the foundation for converting textual data into numerical format for input into the Feedforward Neural Network (FFNN).

- The Size of the Embedding of a Word

  Each word in the vocabulary is represented as a 300-dimensional dense vector. This dimensionality is widely used in pre-trained Word2Vec models (e.g., Google News Word2Vec) to strike a balance between capturing semantic relationships and computational efficiency.

- Embeddings of a Review

To represent a review, we aggregated the embeddings of individual words. Specifically, the embeddings of all words in a review were averaged to form a single fixed-length vector representing the overall sentiment and context of the review.

- The Input of FFNN for a Review

The aggregated review embeddings were used as inputs to the FFNN. Each input vector had a fixed size of 300 dimensions, regardless of the original length of the review, ensuring consistency across all inputs.

3.2 Feedforward Neural Network

The FFNN serves as the classifier that predicts the sentiment of a review based on its numerical representation.

- The Architecture of the Model

The model consists of the following layers:

1. Input Layer: A layer that accepts the 300-dimensional review embeddings.
2. Hidden Layers: Two fully connected layers with 128 and 64 neurons, respectively. These layers capture non-linear relationships in the data.
3. Output Layer: A single neuron with a Sigmoid activation function to output a probability score for binary classification (positive or negative sentiment).

- Activation Functions of Neurons

    - ReLU (Rectified Linear Unit): Used in the hidden layers to introduce non-linearity and prevent the vanishing gradient problem.
    - Sigmoid: Applied in the output layer to produce probabilities between 0 and 1, suitable for binary classification.

3.3 Training Process

Training the FFNN involves optimizing its weights to minimize prediction errors. This is achieved through backpropagation and iterative updates.

- The Loss Function

Binary Crossentropy was used as the loss function, which quantifies the difference between predicted probabilities and actual sentiment labels. It is well-suited for binary classification tasks.

- The Stop Criteria

Training stops when the validation loss no longer improves over successive epochs. This prevents overfitting and ensures that the model generalizes well to unseen data.

- The Early Stopping Method

Early stopping was implemented to monitor validation loss. If no improvement was observed for 5 consecutive epochs, training was halted to save computational resources and prevent overfitting.

4. Results

4.1 Evaluation metrics

The model's performance was evaluated using standard metrics for binary classification, including precision, recall, F1-score, and overall accuracy. The results from the confusion matrix and classification report are summarized below:

- Precision: The model achieved 0.83 for negative reviews and 0.86 for positive reviews, indicating a high level of correctness in predictions.
- Recall: Recall was 0.86 for negative reviews and 0.83 for positive reviews, demonstrating the model's ability to identify true positives effectively.
- F1-Score: Both classes achieved an F1-score of 0.85, reflecting a balanced trade-off between precision and recall.
- Accuracy: The overall accuracy of the model on the test dataset was 85%.

	precision	recall	f1-score	support
negative	0.8393	0.8692	0.8540	3708
positive	0.8675	0.8373	0.8521	3792
accuracy			0.8531	7500
macro avg	0.8534	0.8532	0.8531	7500
weighted avg	0.8536	0.8531	0.8531	7500

Fig 4: Classification Report

4.2 Confusion matrix

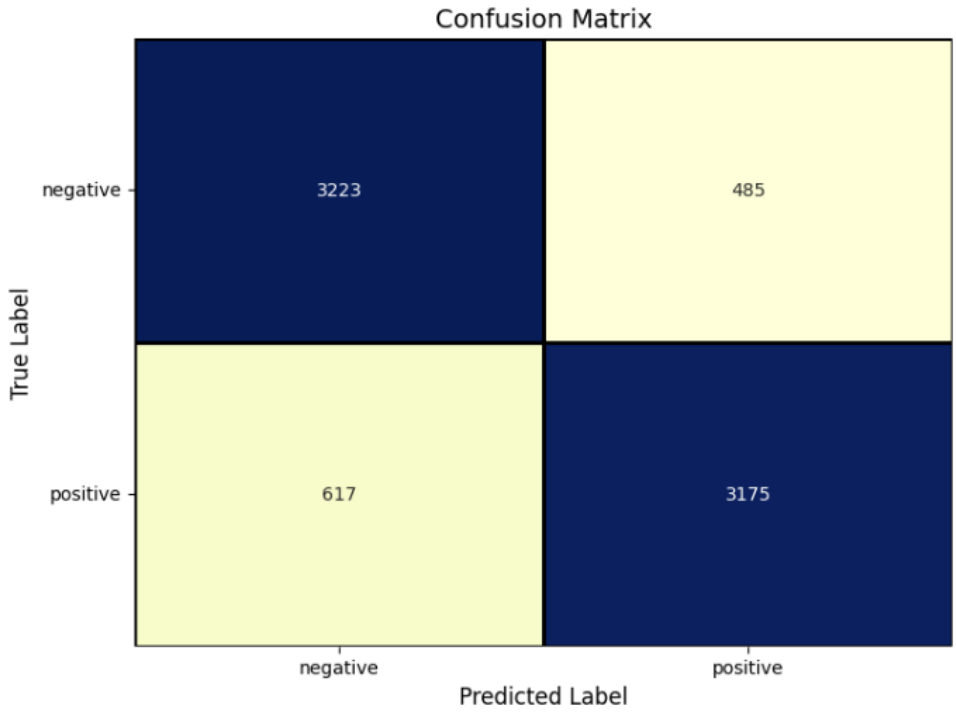

Fig 5: Confusion Matrix

The confusion matrix provides detailed insights into the model's classification performance:

- **True Negatives (TN)**: 3,223 instances were correctly classified as negative.
- **False Positives (FP)**: 617 instances were misclassified as positive when they were negative.
- **False Negatives (FN)**: 485 instances were misclassified as negative when they were positive.
- **True Positives (TP)**: 3,175 instances were correctly classified as positive.

This indicates that the model correctly classified the majority of both positive and negative reviews, with some minor misclassifications.

4.3 Results

The training and validation progress of the model was visualized through accuracy and loss curves:

1. Accuracy:

- The training accuracy steadily increased over epochs, approaching 85%.
- Validation accuracy remained consistently close to training accuracy, indicating good generalization and minimal overfitting.

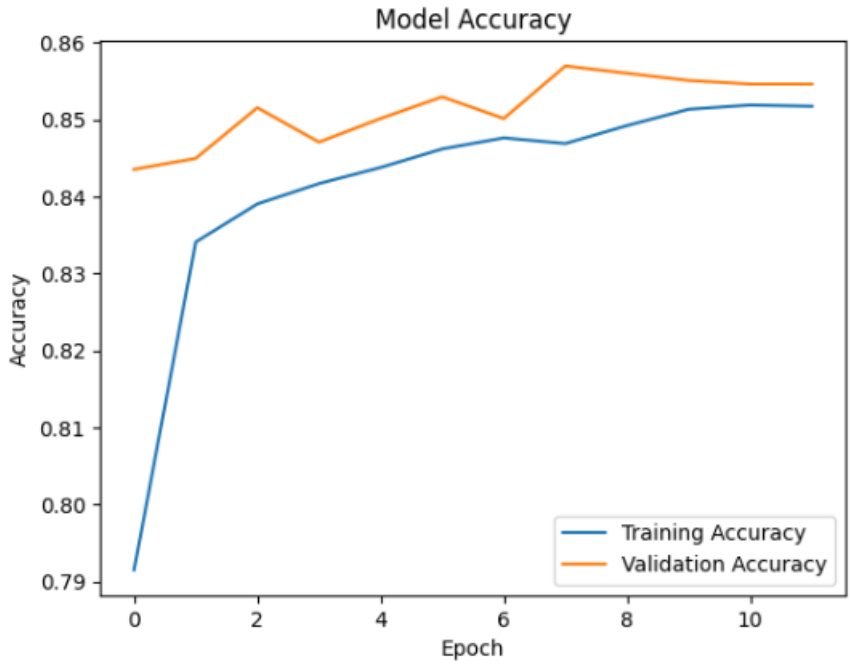

Fig 6: Model Accuracy

2. Loss:

- The training loss decreased steadily throughout the training process, reflecting improved predictions as the model learned.
- Validation loss closely followed training loss, demonstrating that the model avoided significant overfitting.

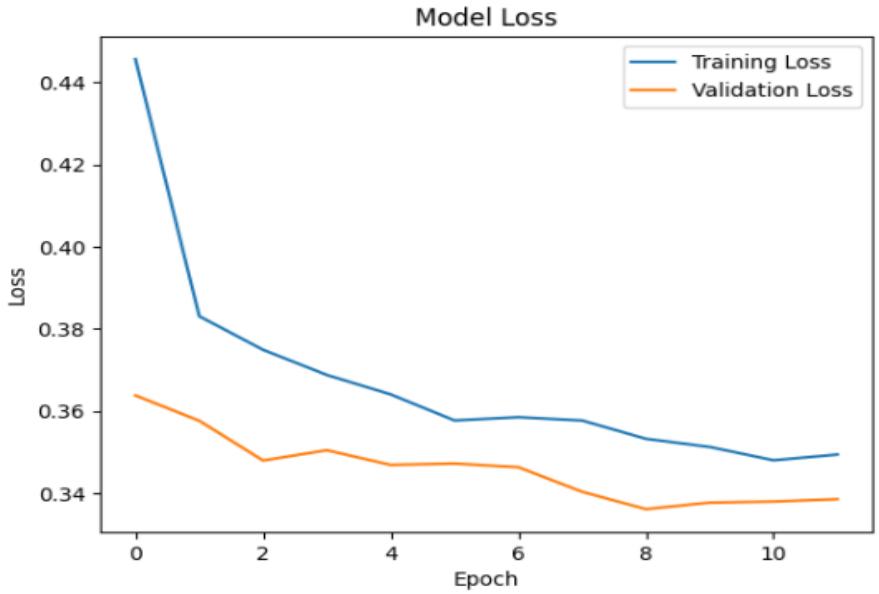

Fig 7: Model Loss

5. Discussions

5.1 Effectiveness of Word2Vec with FFNN

The integration of Word2Vec embeddings with a Feedforward Neural Network (FFNN) proved to be an effective approach for sentiment analysis. Word2Vec successfully captured the semantic relationships between words, transforming unstructured text data into meaningful numerical vectors. These dense representations enabled the FFNN to process the data efficiently and make accurate predictions. The high precision, recall, and F1-scores achieved for both sentiment classes demonstrate the robustness of this combination. Additionally, the use of pre-trained embeddings accelerated the development process and reduced the dependency on large labeled datasets for training.

5.2Pros and Cons

Pros:

- **Simplicity and Efficiency**: The FFNN architecture is computationally efficient compared to more complex models like transformers or recurrent networks.
- **Semantic Understanding**: Word2Vec embeddings encapsulate word meanings and contexts effectively, enabling the model to perform well even on relatively short reviews.
- **Generalization**: The training and validation curves showed minimal overfitting, reflecting good generalization to unseen data.
- **Interpretability**: The approach is relatively simple to interpret and explain compared to deep transformer-based models.

Cons:

- **Context Limitations**: Word2Vec embeddings do not capture word sense disambiguation or contextual nuances within the reviews, as they treat each word independently.
- **Fixed-Length Representations**: Averaging word embeddings to create sentence vectors may oversimplify complex sentence structures, potentially leading to information loss.
- **Dependency on Pre-trained Embeddings**: The effectiveness of the model is partly tied to the quality of pre-trained Word2Vec embeddings, limiting adaptability to domain-specific datasets without retraining embeddings.

5.3 Further Improvements

To enhance the performance and address the limitations of the current approach, the following improvements could be considered:

- **Contextual Embeddings**: Using modern embeddings like BERT or GPT, which capture contextual nuances, could significantly improve the model's understanding of sentiment.

- **Advanced Architectures**: Experimenting with recurrent networks (e.g., LSTMs, GRUs) or attention mechanisms could help in capturing sequential and hierarchical patterns within text.
- **Feature Engineering**: Incorporating additional features such as sentiment lexicons, part-of-speech tags, or TF-IDF scores might further refine predictions.
- **Data Augmentation**: Expanding the dataset with augmented reviews could improve model generalization, especially for minority sentiment classes.
- **Ensemble Learning**: Combining FFNN with other machine learning models could provide a robust ensemble approach for sentiment analysis.

6. Conclusions

This project successfully implemented a sentiment analysis model using Word2Vec embeddings and a Feedforward Neural Network (FFNN). The approach achieved an accuracy of 85%, with balanced precision and recall, effectively classifying both positive and negative sentiments. Word2Vec embeddings provided meaningful representations of text, and the FFNN leveraged these embeddings to deliver reliable predictions with minimal overfitting.

The results demonstrate the strength of combining traditional embedding techniques with neural networks for sentiment analysis. While the model performed well, there is room for improvement, such as incorporating contextual embeddings like BERT or exploring advanced architectures like transformers to capture more nuanced context. Overall, this project highlights the potential of embedding-based deep learning approaches for text classification tasks and provides a strong foundation for further advancements in sentiment analysis.