BLOG

Sentiment Analysis on IMDB Movie Reviews Using CNN — A Deep Learning Approach

NAME: R. SHREYA JANAKI [AM.SC.P2ARI25018]

Class: S1 M tech AI

Institution: School of engineering , Amritapuri

Faculty mentor : Dr Swaminathan J

Introduction

With the growth of online streaming platforms, people continuously express their opinions about movies through reviews. While these reviews help future viewers decide what to watch, analysing thousands of them manually can be challenging. This is where **sentiment analysis** comes in.

In this project, I built a **Convolutional Neural Network (CNN)** model that automatically classifies IMDB movie reviews as *positive* or *negative*. The goal was to learn sentiment patterns from text and predict how viewers feel about a movie.

Objective

The main objective of the project is to design and train a deep learning model that can accurately classify the sentiment of movie reviews using the IMDB dataset.

Dataset Overview

The project uses the **IMDB Movie Review Dataset**, a benchmark dataset for sentiment analysis.

Key Dataset Details

- Total reviews: **50,000**
- Training samples: **25,000**
- Test samples: **25,000**
- Balanced classes (equal positive & negative)
- Reviews are converted into sequences of integers
- Vocabulary limited to **10,000 words**
- Each review is padded to a fixed length of **200 tokens**

Input (X): Padded word index sequences
Output (Y): 0 = Negative, 1 = Positive

Methodology

The workflow followed in this project is:

1. **Load dataset** → IMDB reviews
2. **Text preprocessing**
    - Tokenization
    - Convert text to sequences
    - Padding
3. **Build CNN model**
    - Embedding Layer
    - Conv1D + MaxPooling
    - Conv1D + GlobalMaxPooling
    - Dense Layers → Sigmoid
4. **Train model**
    - 80% Train
    - 20% Validation
5. **Evaluate on test data**
6. **Predict new reviews**

This workflow enables the model to learn both semantic and contextual features from the reviews.

Why CNN?

Although CNNs are widely known for handling images, they are also highly effective in text classification. CNN filters help capture important features (like key phrases) from reviews.

Benefits of CNN for text

- Faster than LSTM
- Captures local context
- Requires fewer parameters
- Performs well with less tuning

Tech Stack

- **Programming Language:** Python
- **Frameworks/Libraries:**
    - TensorFlow / Keras
    - Scikit-learn
    - Pandas
    - Matplotlib
- **Environment:** Google Colab

Model Architecture

- Embedding (vector representation)
- Conv1D
- MaxPooling
- Conv1D
- GlobalMaxPooling
- Dense (ReLU)
- Dense (Sigmoid)

This architecture detects text features and predicts the final sentiment.

Results

The model performed well, achieving:

Metric	Value (%)
Accuracy	84.7%
Precision	85.0%
Recall	85.0%
F1-Score	85.0%

The training and validation accuracy curves showed stable learning after three epochs. Slight overfitting was observed, but performance remained strong on test data.

Insights & Takeaways

- CNNs are powerful for text classification tasks.
- Even without LSTM or Transformer models, CNN achieved **~85% accuracy**.
- Tokenization and padding are essential steps for text handling.
- Using more advanced architectures (LSTM, BERT) could further improve accuracy.

Future Improvements

- Experiment with LSTM / Bi-LSTM / Transformer models
- Use pretrained word embeddings (GloVe, Word2Vec)
- Hyperparameter tuning
- Add dropout / regularization to reduce overfitting
- Deploy as a web app / API

Applications

- Product review sentiment recognition

- Social media opinion monitoring
- Customer feedback analysis
- Personalized recommendations

Conclusion

This project demonstrates how CNNs can be effectively applied to sentiment analysis. The model successfully classifies IMDB reviews with high accuracy and offers opportunities for further enhancement. This approach can be extended to analyze opinions in many real-world domains such as e-commerce, entertainment, and marketing.

References

- IMDB dataset — TensorFlow/Keras
- Deep learning references on CNN text modeling
- Project notebook