Sentiment Classification of IMDb Reviews

Yang Liu

23204543

yang.liu2@ucdconnect.ie

Abstract

Sentiment analysis is a highly studied domain aimed at extracting meaningful insights from large datasets. This paper investigated four models—CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory), CNN-LSTM, and ensemble of CNN and CNN-LSTM—to analyze the sentiment of IMDb reviews. The dataset underwent preprocessing and was randomly divided into training, validation, and test sets. Hyper-parameters were optimized for each model, and their classification performance was evaluated based on accuracy. Experiments have revealed that the performances of the three individual models are close, with the CNN classifier slightly outperforming the other two, achieving an accuracy rate of 89.86%. The ensemble model outperforms both the individual models, achieving an accuracy rate of 90.5%.

1 Introduction

With the continuous growth of the film industry and the expanding influence of social media, analyzing the sentiments expressed in film reviews has become increasingly valuable. This analysis is essential not only for individuals seeking to make informed choices about films but also for industry professionals who rely on feedback to gauge audience reactions and refine their strategies.

CNN and LSTM have been proved effective in sentiment classification of texts. This paper investigates three distinct models for sentiment classification of IMDb reviews, drawing inspiration from established methodologies. Various configurations are explored for each model. To leverage the advantage of CNN and LSTM, an ensemble of CNN and CNN-LSTM has also been explored and proves effective.

The remainder of the paper is organized as follows:

Section 2 reviews relevant literature in the field of text classification, focusing particularly on sentiment classification of movie reviews.

Section 3 details the preprocessing steps applied to the dataset and describes the structures of the models used.

Section 4 presents the results obtained using the best model configurations.

Section 5 concludes the paper and discusses the challenges encountered along with potential directions for future work.

2 Related Work

LSTMHochreiter and Schmidhuber [1997] is a popular RNN (recurrent neural network) architecture for modeling sequential data. Sentiment analysis of text based on LSTM has been explored extensively and has demonstrated promising results. Qaisar [2020] Murthy et al. [2020]

CNN-LSTMs were proposed in several papers. It is a class of models that is both spatially and temporally deep and has the flexibility to be applied to a variety of vision tasks involving sequential inputs and outputs. This architecture has been used on a variety of problems, including speech recognition and NLP (natural language processing) where CNNs are used to extract features from audio and text input data for the LSTMs layers. Sainath et al. [2015] Donahue et al. [2015]Shi et al. [2015]. LSTM combined with multiple branches of CNN was explored in Yenter and Verma [2017] which proposed a model of accuracy rate over 89%.

Ensemble is a very popular approach in deep learning. An ensemble of LSTM and CNN was proposed in Minaee et al. [2019], achieving an accuracy of 90% on IMDb dataset. Inspired by that, an ensemble of CNN and CNN-LSTM was explored in this paper.

Comparison studies of CNN, LSTM and CNN-LSTM have been conducted in several papers. Ali et al. [2019]Haque et al. [2019].

3 Experiment Setup

3.1 Data Preprocessing

The IMDb dataset contains 50k reviews, of which 25k reviews are labeled as 'positive', and 25k are labeled as 'negative'. The dataset was randomly shuffled (seed = 2023) and split into training, validation, and test set (ratio = 0.64 : 0.16 : 0.2).

First of all, HTML tags were removed from the reviews because they do not afford any useful information. Labels of 'positive' and 'negative' were converted to 1 and 0 respectively. From the train set, a dictionary of 10k words was built using the TextVectorization layer of keras. Preprocessing steps, such as removing punctuations, converting all letters into lower case letters, were conducted in this step using the default configuration of keras. Validation set and test set were excluded from the dictionary to make the model better adapt to unfamiliar words which are not in the dictionary. The maximum sequence length was set to 500 words. Reviews that were longer than that were truncated, leaving only the words with the highest frequency. Reviews shorter than 500 words were post-padded with zero.

GloVe 50d pre-trained embeddings Pennington et al. [2014] proved to have no positive effect, so it was not used in this paper.

The classification models were built with Python and Tensorflow on Colab. All training curves were drawn with Tensorboard. Early stop strategy was adopted during the training process. If the validation loss did not decrease for consecutive three epochs, the training would be stopped, and the model with the highest validation accuracy rate would be saved.

Four models, namely CNN, LSTM, CNN-LSTM, and ensemble of CNN and CNN-LSTM, are proposed in this study. The architecture of the CNN classifier will be described comprehensively, while many details of the LSTM and CNN-LSTM classifiers will be omitted for conciseness and to conserve space, except layers and parameters which are different from CNN.

Numerous experiments were conducted to optimize the hyper-parameters based on the performance observed on the validation set. The quality of the models was assessed by evaluating their classification accuracy on the test set.

3.2 CNN

The architectures of this model is shown in figure 1.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 500, 64)	640000
dropout (Dropout)	(None, 500, 64)	0
conv1d (Conv1D)	(None, 500, 128)	24704
conv1d_1 (Conv1D)	(None, 500, 128)	49280
conv1d_2 (Conv1D)	(None, 500, 128)	49280
max_pooling1d (MaxPooling1D)	(None, 250, 128)	0
flatten (Flatten)	(None, 32000)	0
dropout_1 (Dropout)	(None, 32000)	0
dense (Dense)	(None, 1)	32001

Figure 1: Architecture of CNN

3.2.1 Embedding Layer

The keras emdedding layer turns positive integers (indexes) into dense vectors of fixed size. Size of 64 integers per word best fit the classifier.

3.2.2 Convolution Layer

Three consecutive 1D convolution layers use ReLU as activation function. Experiments found ReLU performs better than tanh. Different number of convolution layers ($1\tilde{4}$) was attempted, and three was proved to be the optimum convolution layer number. Different filter sizes and kernel sizes were also attempted, and the optimum filter size is 128 and the optimum kernel size is 3.

3.2.3 Dropout and Max Pooling Layer

Two dropout layers were added to fight over-fitting. Different dropout rates were attempted and 0.2 was proved to be the optimum dropout rate.

The max pooling layer condenses the output of the convolution layer by half. Experiments found the optimum kernel size is two.

3.2.4 Dense layer and Optimiser

The output of max pooling layer was flattened before sending to the dense layer. The activation function of the dense layer is sigmoid.

The optimizer was Adam with learning rate set to 0.001. Increasing the learning rate or replacing Adam to SGD or AdamW didn't improve the performance of the model.

3.3 LSTM

The architectures of this model is shown in figure 2.

3.3.1 Embedding and Output

The embedding layer and the final dense layer are configured identically to those in the CNN classifier.

3.3.2 LSTM Layer and Dense Layer

A single LSTM layer is followed by a dense layer with a size of 64. The activation function of the LSTM layer is ReLU.

Various dropout rates for inputs and recurrent states, as well as the use of two LSTM layers, have been explored; however, these adjustments did not yield significant improvements. Consequently, no dropout layers were included, and only a single LSTM layer has been retained.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 500, 64)	640000
lstm (LSTM)	(None, 64)	33024
dense (Dense)	(None, 64)	4160
dense_1 (Dense)	(None, 1)	65

Figure 2: Architecture of LSTM

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 500, 64)	640000
conv1d (Conv1D)	(None, 500, 128)	24704
conv1d_1 (Conv1D)	(None, 500, 128)	49280
max_pooling1d (MaxPooling1D)	(None, 250, 128)	0
bidirectional (Bidirectional)	(None, 200)	183200
dense (Dense)	(None, 1)	201

Figure 3: Architecture of CNN-LSTM

In a different experiment, the LSTM layer was replaced with a bidirectional LSTM layer, as depicted in table 1. The resulting accuracy was slightly lower compared to that of the LSTM configuration, and the training time was extended. As a result, the use of LSTM alone was deemed sufficient for this task.

3.4 CNN-LSTM

The architectures of this model is shown in figure 3.

The CNN-LSTM model is inspired by Brownlee [2017], which employed a pre-padding strategy for word sequences shorter than 500 words. Experiments demonstrated that this classifier performed optimally with pre-padding, while post-padding severely hinders model convergence.

To address the sensitivity to padding, the LSTM layer has been substituted with a bidirectional LSTM layer.

The configuration of the embedding layer, convolution layers, and max pooling layer remains consistent with that of the CNN model.

3.5 Ensemble of CNN and CNN-LSTM

The structure of the ensemble model is shown in figure 4. The predictions from the CNN and CNN-LSTM are averaged to produce a final prediction. Accuracy of this ensemble is shown in table 1.

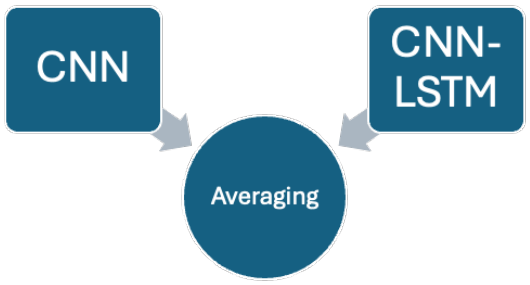

Figure 4: Ensemble of CNN and CNN-LSTM

4 Results

Learning curves calculated from test set and validation set are shown in figure 5, figure 6 and figure 7. The validation accuracy of the three models reached around 89.5% before overfitting.

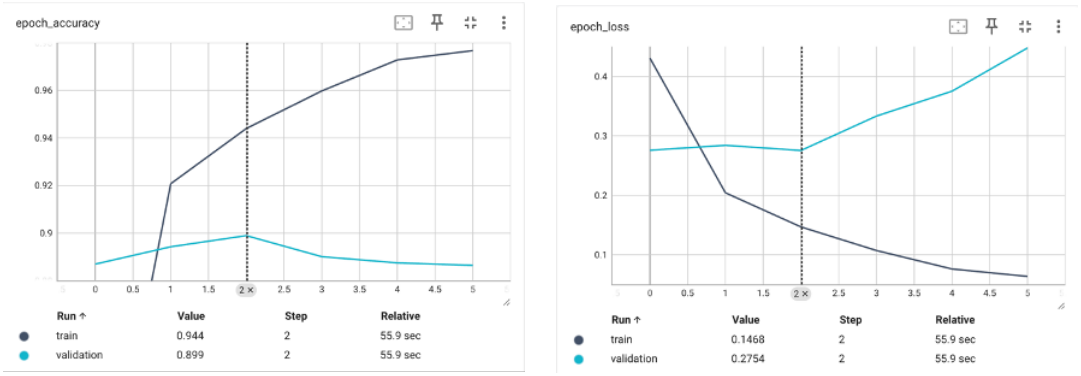

Figure 5: Left: Accuracy curve of CNN. Right: Loss cure of CNN

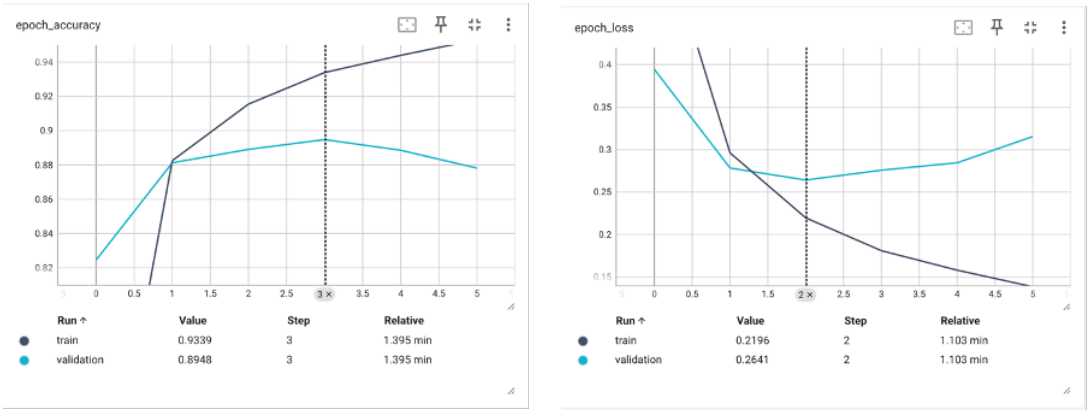

Figure 6: Left: Accuracy curve of LSTM. Right: Loss cure of LSTM

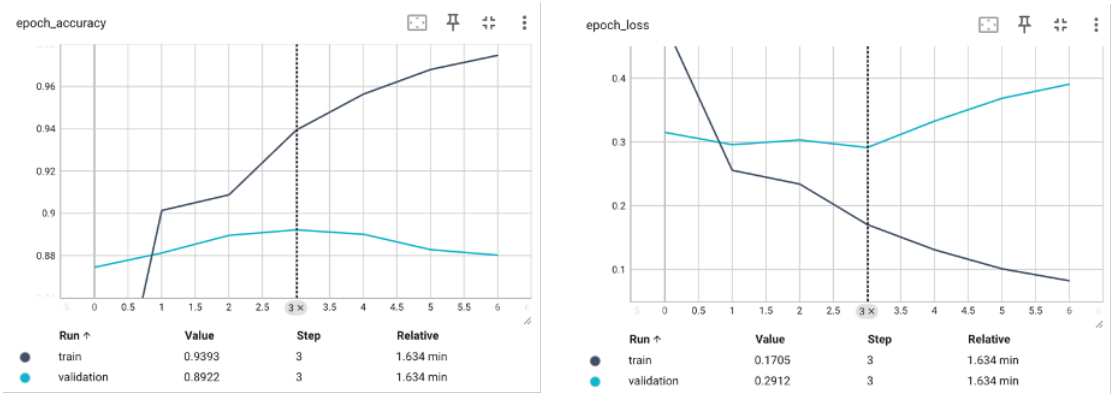

Figure 7: Left: Accuracy curve of CNN-LSTM. Right: Loss cure of CNN-LSTM

The performances of the three models are assessed by their classification accuracy on the test set which contains 10k reviews. The use of accuracy as an evaluation measure is appropriate for this dataset due to its balanced class distribution. The predictions for the test set consist of an array of floating-point numbers ranging from 0.0 to 1.0. Values greater than 0.5 are classified as indicating

Model	CNN	LSTM	Bi-LSTM	CNN-LSTM	Ensemble
Test Accuracy	89.86%	89.3%	89.54%	89.64%	90.5%

Table 1: Accuracy on test set

positive sentiment, while values equal to or less than 0.5 are classified as indicating negative sentiment. Accuracy are calculated and shown in table 1. After parameters tuning, the three individual models perform very similarly, each achieving an accuracy rate over 89%, with CNN slightly outperforming the other two. The ensemble model of CNN and CNN-LSTM outperforms both the individual models.

5 Conclusion and Future Work

This paper investigated four models—CNN, LSTM, CNN-LSTM, and ensemble of CNN and CNN-LSTM—for sentiment classification of IMDb reviews. Through extensive parameter tuning, the results demonstrated that all three models performed well on the IMDb test set, achieving accuracies exceeding 89% and comparable to existing approaches.

Future research should explore alternative ensemble approaches leveraging respective strengths of CNN and LSTM. Additionally, exploring popular architectures such as Transformers and Attention in future studies may lead to improved performance compared to the models presented in this paper.

References

Nehal Mohamed Ali, Marwa Mostafa Abd El Hamid, and Aliaa Youssif. Sentiment analysis for movies reviews dataset using deep learning models. *International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol*, 9, 2019.

Jason Brownlee. Sequence classification with LSTM recurrent neural networks in Python with Keras. `https://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/`, 2017. Accessed: [Insert Access Date].

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

Md Rakibul Haque, Salma Akter Lima, and Sadia Zaman Mishu. Performance analysis of different neural networks for sentiment analysis on imdb movie reviews. In *2019 3rd International conference on electrical, computer & telecommunication engineering (ICECTE)*, pages 161–164. IEEE, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Shervin Minaee, Elham Azimi, and AmirAli Abdolrashidi. Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models. *arXiv preprint arXiv:1904.04206*, 2019.

GSN Murthy, Shanmukha Rao Allu, Bhargavi Andhavarapu, Mounika Bagadi, and Mounika Belusonti. Text based sentiment analysis using lstm. *Int. J. Eng. Res. Tech. Res*, 9(05), 2020.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `https://www.aclweb.org/anthology/D14-1162/`.

Saeed Mian Qaisar. Sentiment analysis of imdb movie reviews using long short-term memory. In *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, pages 1–4. IEEE, 2020.

Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4580–4584. Ieee, 2015.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Alec Yenter and Abhishek Verma. Deep cnn-lstm with combined kernels from multiple branches for imdb review sentiment analysis. In *2017 IEEE 8th annual ubiquitous computing, electronics and mobile communication conference (UEMCON)*, pages 540–546. IEEE, 2017.