Enhancing Sentiment Analysis with a CNN-Stacked LSTM Hybrid Model

Shuaijie Shao

Shijiazhuang No.2 High School, 050000, Shijiazhuang, China

Abstract. This paper focuses on developing a new hybrid model to solve sentiment analysis problems in Natural language processing. Sentiment analysis is a key branch of Natural language processing (NLP) and new models with better performance can boost the development of machine learning. The new model mentioned in this research is a hybrid model containing convolutional neural network (CNN), stacked multi-layer long short-term memory (LSTM) and max pooling layers. This model uses CNN for its advantage of capturing local features in the sequence after the embedding process, and LSTM for its advantage of capturing long-term dependencies in such sequential data after CNN layer. The global max pooling layer can better organize the entire sequence. This model has been tested to show that it has a better performance than previously mentioned models when solving the sentiment analysis task based on IMDB dataset provided by TensorFlow. Introducing this new model in sentiment analysis may open new avenues for research. The performance of the model can be further improved, offering valuable insights for future hybrid model development in machine learning tasks.

1 Introduction

Natural language processing (NLP) is an important branch of machine learning targeting on allowing machine to understand, process or generate human language [1]. Sentiment analysis has long been a hot topic in NLP, it is a method used to convert human language to machine and let them decide whether the sentiment in the paragraph is positive or negative [2]. Although current days the most advanced technology in machine learning is large language models, and they can complete most jobs NLP does, but due to the cost and model complexity of large language models, sentiment analysis still maintains an important role in analyzing user reviews, monitoring social media comments and even economic analysis [3]. Due to this unique importance, it still is and will always be an indispensable part in the development of machine learning.

However, although there are multiple models available for use, due to the restrictions of its model complexity, its accuracy is still far from large language models. Hence how to improve model performance in restricted model complexity has become a serious problem for researchers. There are already countless studies trying to boost the accuracy of sentiment

Corresponding author: shawn@shawnshao.org

analysis models, and here are some ways to do so. On the one hand, which is in fact a more basic work, is to fit in different vectorization methods or classification methods, such as Word2Vec [4], Term frequency-inverse document frequency (TF-IDF) [5], Bidirectional Encoder Representations from Transformers (BERT) or so for vectorizer [6], and logistic regression [7], convolutional neural network (CNN) [8], recurrent neural network (RNN) [9], long short-term memory (LSTM) or so as classifiers [10]. Moreover, researchers have put much effort on tuning these models to seek for the best hyperparameters. On the other hand, recent studies no longer limit their sights on a specific model, instead, they combined multiple models together as a single section, for example, Prof. Beakcheol Jang created a hybrid model applied a CNN model to the Word2Vec, then using bidirectional LSTM and process the result with an attention layer [11], which at last performed a higher accuracy than all tested single models [12].

In this research, the goal is to develop a new hybrid model to solve sentiment analysis problems. Meanwhile, since sentiment analysis has to keep a relatively low model complexity, the new model should not require many resources to train and use. Being encouraged by using a CNN model to extract the important sections of the transformed sentences, a multi-layer LSTM model was applied to the extracted input, which allowing the model although not processing sequences in two directions, but multiple times. The hybrid model has shown a great success, which by analyzing the validation accuracy to epochs curve, the model has overall better result than previous models, and after using early stopping method to get the best test accuracies of the model, they are noticeably better than others.

2 Dataset and methodologies

2.1 Dataset

In this study, the dataset of IMDB from TensorFlow was used, which is a dataset that is often used to train and test sentiment analysis models. Choosing this dataset can make it easier to compare the new model's performance with former models since most previous models already have a tested accuracy of this dataset. There are 25000 train data and 25000 test data in total, containing labels (containing pos, positive, and neg, negative) and user reviews (comments written in English, with sentiments which suits the labels).

2.2 Method

2.2.1 CNN

The key idea behind CNN is the convolutional layer, unlike normal CNNs that process images, which requires 2-dimensional process, the CNN used to extract text features only need to deal with one dimensional data. It works by sliding filters across the vectorized text and doing convolution operation to them to extract local features. A 1D convolution operation is defined as formula 1.

$$y(i) = \sum_{j=0}^{k-1} x(i+j) \cdot \omega(j) + b \tag{1}$$

Where the input vector is defined as x, the convolution kernel is ω, bias term is b and k stands for the size of the convolution kernel. Then we can see that $x(i+j)$ is the value of the input vector at position $i+j$, $w(j)$ is the weight of the convolution kernel at position j, and $y(i)$ is the output value at position i in the sequence.

By calculate all the convolution result of the original vector, this layer gives an output of a new feature sequence that captures the local patterns in the input vector.

As for the activation function, a Rectified Linear Unit (ReLU) activation function was used to add non-linear features to the input sequence. The formula of ReLU activation function is as follows:

$$f(x) = max(0, x) \tag{2}$$

Which sets any negative input to zero and keeps other values not changed. This can make the model converge more easily and enhance feature representation meanwhile.

The model used max pooling as its pooling method, which is a method to choose the maximum value inside the pooling window. In this way it can reduce output dimensions and keep the extracted features. Formula 3 gives a mathematical representation of this behavior.

$$y(i) = max\big(x(2i), x(2i+1)\big) \tag{3}$$

2.2.2 LSTM

LSTM is a type of advanced RNNs, it uses three different gates to control the data flow and solve the problem of gradient disappear and gradient explode.

The first gate is the forget gate, it uses a sigmoid function as it's activation function to decide the importance of the data from the previous memory cell, and pointwise multiply the data to the number generated by the sigmoid function, 0 means to clear all the former data and 1 to keep all of them. Formula 4 is the sigmoid function and formula 5 is the formula of the forget gate.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

$$f_t = \sigma\big(W_f x_t + W_f h_{t-1} + b_f\big) \tag{5}$$

Here σ is the sigmoid activation function, W_f is the weight matrix, b_f is the bias term, h_{t-1} is the hidden state of time $t-1$, x_t is the current input and f_t is the output of the forget gate.

The second gate is the input gate, which is used to decide how much information should be added to the memory cell in this time. Formula 7 is its formula, and formula 8 is a formula to calculate the candidate cell state. Which i_t is the output of the input gate and $\tilde{C}_t$ is the candidate cell state, $tanh$ is the hyperbolic tangent function which is used as an activation function shown in formula 6.

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{6}$$

$$i_t = \sigma(W_i x_t + W_i h_{t-1} + b_C) \tag{7}$$

$$\tilde{C}_t = tanh(W_c x_t + W_c h_{t-1} + b_C) \tag{8}$$

There is also a function used to update the cell state after forget gate and input gate, which is shown in formula 9, here C_{t-1} stands for the previous cell state.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{9}$$

The last section is the output gate, which directly decide the output of the entire LSTM using the information inside the cell state. Formula 10 is the formula of the output gate and the final hidden state h_t is shown in formula 11.

$$o_t = \sigma_g(W_o x_t + W_o h_{t-1} + b_o) \tag{10}$$

$$h_t = o_t \odot tanh(C_t) \tag{11}$$

2.2.3 Hybrid model

The hybrid models introduced in this paper uses embedding layer, CNN layer, max pooling layer, two LSTM layers, attention layers, global max pooling layer and dense layer. The embedding layer maps

each word to a fixed-dimensional vector representation, which captures the semantic information of them. The CNN layer extracts the local features such as phrases and local features. The max-pooling layer down samples the output of the convolutional layer, in this way it can reduce the dimensionality and preserve important features. The two LSTM layers are used to capture long-term dependencies in the data. By using a stacked multi-layer LSTM, the model can better capture all the contextual information in reviews. Due to the length of the reviews, attention layers were added behind each LSTM layer, this mechanism weights different time steps' outputs and highlights the important features. To alleviate the vanishing gradient problem, the sequence being processed by attention layer will be connected with the LSTM output using residual connection then normalize the output sequence and continue the further steps. Finally, the global max-pooling layer extract the most significant features and the dense layer maps them together and process them with a hidden layer, here the result will be sent to the output layer and complete the entire process. Fig. 1 is a more lucid pipeline of the model's process mentioned above.

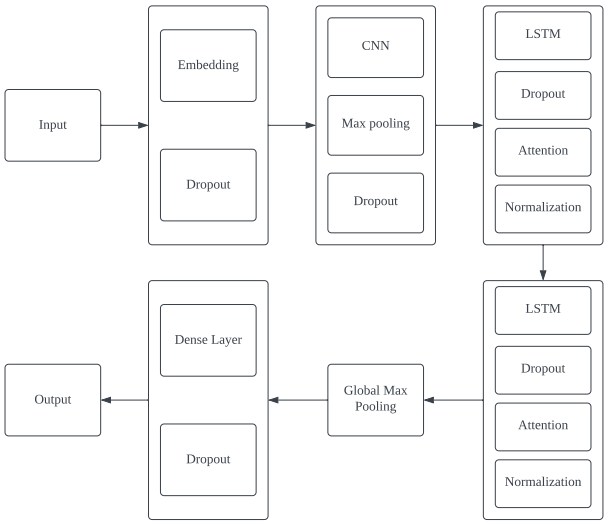

Fig. 1. Pipeline of the hybrid model (Photo/Picture credit : Original)

In this study, the embedding size was set to 500, and all dropout layers have the same dropout rate, which is 0.2, the batch size of the model is 128. In the dataset, only the 10000 words having the most occurrence was taken into consideration. The tested model has a training epoch of 20 and for the final result there is an early stop mechanism applied to the model. There is a brief overview of the parameters used in Table 1.

Table 1. Parameters used for the model.

Embedding size	Dropout rate	Batch size	Words considered	Epochs trained
500	0.2	128	10000	20

2.3 Evaluation matrix

To evaluate the model's performance, this research used accuracy as the evaluation method. The formula of accuracy is shown as formula 12 [13].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP+TN+FP+FN}} \tag{12}$$

Where TP stands for true positive, which is the number of instances which the model correctly predicts the positive class. TN stands for true negative, which is the number of

instances the model correctly predicts the negative class. FP stands for false positive, which is the number of instances the model incorrectly predicts the positive class. FN stands for false negative, which is the number of instances the model incorrectly predicts the negative class.

3 Result and analysis

After having a result accuracy of the hybrid model developed, some other models containing simple models such as CNN or LSTM, hybrid models such as one using Bidirectional LSTM and other models were also tested. The result of these models was listed in Table 2, these results are ones using the early stop mechanism. It can be seen that CNN models perform worse in such occasions, with only a validation accuracy of 0.889 while other models have accuracy higher than 0.91. Meanwhile, the hybrid model introduced in this paper performs the best among all other models with an accuracy of 0.9257, surpassing a previous hybrid model of more than 0.01 in solving such task.

Table 2. Validation accuracy of the tested models.

Models	LSTM	CNN	Previous Hybrid model	Research Hybrid model
Val accuracy	0.9103	0.8892	0.9115	0.9257

Later on, the early stop mechanism was removed and a graph of model accuracy versus trained epochs were drawn in Fig. 2.

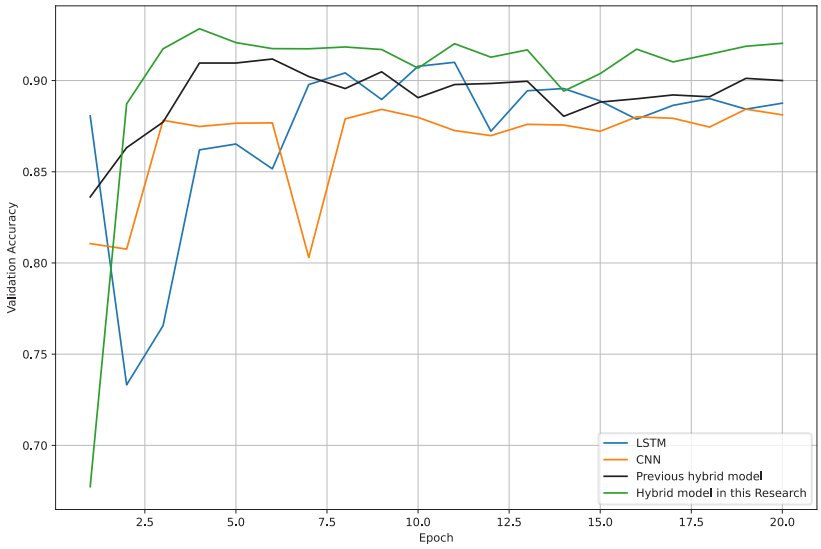

Fig. 2. Model accuracy versus epochs (Photo/Picture credit : Original)

From such results in Fig. 2 it can be seen that the hybrid model introduced in this research have surpass other previous models that are tested. All the models have shown a high performance after about 8 epochs, having validation accuracy ranged from 0.88 to 0.93, and did not improve for more than 0.01 when training further epochs. The black line can be seen as a baseline of hybrid model since it is a model mentioned in research published in 2022,

and the green line is the new model mentioned in this research which has successfully surpasses the previous hybrid model for more than 1 percent comparing the validation accuracy. Blue and orange line demonstrates two single models, which is a main component of hybrid models, and have shown quite a good performance, especially the LSTM model, which has the highest validation accuracy of around 0.91. Meanwhile, the new model did not consume too much computational resources and don't need large epochs trained to reach its peak performance. This has shown that CNN models can work well with stacked LSTM in doing sentiment analysis. This combination successfully used CNN as it can capture the local features through convolution operations and LSTM for it can capture long-term dependencies in sequential data. Such it can be shown that the new hybrid model developed is in fact a boost in the accuracy of sentiment analysis.

Although the model mentioned in this research is more effective than existed models, it still should be tried on other situations and tuned the parameters to reach its best performance. Meanwhile, only a single validation accuracy was applied to evaluate the model's performance, which is not enough to prove the model's high performance in all cases. Such other validation matrix should be applied to the model later on. Developing new sentiment analysis methods is pushing forward our understanding of basic machine learning models, which can benefit NLP and the entire machine learning progress. So, this work is well worth doing and need to be careful and take every step serious.

4 Conclusion

This focus of this paper is to develop a new hybrid model that can solve sentiment analysis problems with higher accuracy and without consuming too many computational resources. The hybrid model introduces mainly uses CNN model for local features detection, stacked layers LSTM model for long term features and max pooling layer to organize the entire sequence. The accuracy of the mentioned model and some other models mentioned in previous research contain both single and hybrid models. The results have shown that the new hybrid model has a better validation accuracy than its components and previous hybrid model using similar base models. This has shown that sentiment analysis in this case requires both short-term extracting and long-term skills when facing natural language processing tasks. Which no matter for CNN, LSTM or max pooling can only satisfy one of the skills required, hence by putting such models together to complement each other's advantages to form a hybrid model to process the text can be an appealing choice. This hybrid model can not only improve the section of sentiment analysis but also be an inspiring branch that can be further developed. The main idea of such a hybrid model is using different model to serve for problems they are good at, hence other models good at solving short-term or long-term sequences may also be applied to such hybrid models to test for accuracy and computational complexity to search for better combinations. Meanwhile, such a idea can also be applied on other part that requires machine learning models, researchers should not be limited on testing any single models, but to try to combined multiple base models together to create a new branch on deciding suitable models.

References

1. Jurafsky, D., & Martin, J. H. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (2nd ed.). Pearson Prentice Hall, (2009).
2. Pang, B., & Lee, L. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, **2**(1-2), 1-135, (2008).

3. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … & Amodei, D. Language models are few-shot learners. Advances in Neural Information Processing Systems, **33**, 1877-1901, (2020).
4. Mikolov, T., Chen, K., Corrado, G., & Dean, J. Efficient estimation of word representations in vector space. Proceedings of the International Conference on Learning Representations (ICLR), (2013).
5. Salton, G., & Buckley, C. Term-weighting approaches in automatic text retrieval. Information Processing & Management, **24**(5), 513-523, (1988).
6. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, **1**, 4171-4186, (2019).
7. Cox, D. R. The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), **20**(2), 215-232, (1958).
8. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324, (1998). https://doi.org/10.1109/5.726791
9. Elman, J. L. Finding structure in time. Cognitive Science, 14(2), 179-211, (1990). https://doi.org/10.1207/s15516709cog1402_1
10. Hochreiter, S., & Schmidhuber, J. Long short-term memory. Neural Computation, **9**(8), 1735-1780, (1997).
11. Schuster, M., & Paliwal, K. K. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, **45**(11), 2673-2681, (1997).
12. Kang, S., & Kim, J. W. Bi-LSTM model to increase accuracy in text classification: Combining Word2vec, CNN, and attention mechanism. Applied Sciences, **10**(17), 5841, (2020).
13. Lewis, D. D. Evaluating and optimizing autonomous text classification systems. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 246-254, (1994).