

A Comparative Exploration of BERT, RNN, and GRU for Sentiment Classification

Can Zhang

College of Letters & Science, University of Wisconsin-Madison, Madison, Wisconsin, 53706,
United State

czhang727@wisc.edu

Abstract. Sentiment classification tasks are a large branch of current natural language processing problems. The purpose of this task lies in determining the sentiment that the text is trying to convey given a piece of text or sentence. This technique can be widely referenced in areas such as, monitoring market opinion, protecting social security, and developing ai customer service. Currently, the most advanced solutions to the sentiment classification problem are based on Bidirectional Encoder Representations from Transformers (BERT), in order to try to research a better solution, this study will start from the bottom, first find the optimal Recurrent Neural Network (RNN) model to pave the way for future improvements. The research method is to start from one layer and keep adjusting the parameters or RNN types until the best model is found. Through an experimental study, it was found that the two-layer bidirectional Long Short-Term Memory (LSTM) performed best on the sentiment classification problem with an accuracy of 93.21%. It proves that LSTM is better than Gated Recurrent Unit (GRU) and simple RNN in understanding long text, and in the future, more datasets can be collected from other platforms, not twitter, and then the model can be trained more complexly, which will help the model to be generalized to other domains or occasions.

Keywords: Sentiment classification, deep learning, recurrent neural network, gated recurrent unit.

1. Introduction

Sentiment classification is currently an important research direction in the natural language processing problem. The core of this problem lies in determining the emotion the text tries to express through the provided text. For example, surprise, happiness, fear, and like. This research can be used for a wide range of purposes, such as market monitoring, where the technology can be used to analyze the reviews of products on social media or various websites to find the market's attitude towards their products and then adjust their sales strategy or make improvements. Regarding social security, the government can use the technology to monitor public opinion in social media if there is any crisis or riot, allowing timely interventions. This technology can monitor the customer's tone in the Artificial Intelligence (AI) customer service application. If it finds signs of anger or impatience, AI should choose artificial customer service intervention or provide better solutions for customer service to pacify the customer. Deep learning is advantageous in solving emotion classification problems. It can automatically extract emotionally relevant features from the text to learn without any set dictionary. Using Recurrent Neural Network (RNN) models, understanding the sentiment of a sentence or paragraph and analyzing more complex sentiments in context is possible.

Almost all current state-of-the-art solutions are based on Bidirectional Encoder Representations from Transformers (BERT), a pre-trained transformer model, which has been fine-tuned in various parameters, resulting in many BERT variants [1]. For example, robustly optimized BERT approach (RoBERTa) simply and directly improves the model's performance by increasing the training time and the amount of data [2]. There is also DistilBERT, which reduces some of the model parameters to make the model faster and thus can be deployed on more resource-limited devices [3]. Transformer is an RNN model that incorporates a self-attention mechanism, so to optimize the model from the root cause, the current study would like to find out the optimal structure of the RNN model that can be used for solving the emotion classification problem. This will pave the way for better BERT models in the future.

2. Method

2.1. Dataset and Preprocessing

In this study, different structures of recurrent neural networks will be constructed to compare and contrast to explore which structure of the model performs better in solving the text sentiment classification problem under the same dataset. The methodology of this experiment mainly includes data preprocessing, feature vectorization, model selection and training, hyperparameter optimization, and performance evaluation. The following are the specific experimental processes and steps. First, the dataset for this experiment comes from the open-source dataset on Kaggle created by user Nidula Elgiriye withana [4]. This dataset contains words expressing six different emotions, all from Twitter, which is authentic data with reference values. After the initial analysis of the dataset, it was found that the dataset had an uneven number of each type of data, so the dataset was downsampled by taking the minimum number of all the categories and then extracting the same number from the other categories.

The sampling process is to divide the original dataset into multiple parts for each category, in which a certain number of samples are randomly taken in each part, and finally, a total number of samples are taken with the minimum number of categories [5]. To reduce the impact of useless text symbols in the text on the model learning, the text needs to be cleaned. Data cleaning involves removing all Uniform Resource Locator (URLs), punctuation, numbers, blanks from the text, and stop words (e.g., is, in, at), as these generally do not contain any sentiment. Retaining them would be a waste of resources. Convert all words to their lowest form to avoid recognizing different words for the singular, plural, or different tenses of a word. After the text is cleaned, all words are separated by spaces so that each text becomes a phrase consisting of many words. This study created a vocabulary list to record each occurrence of a different word in the sample into a vocabulary label and numbered sequentially. In this way, each word in each previous sentence can be represented by a number and a sentence can be represented by a set of numbers. This way, the data can be expressed in a matrix to facilitate future model learning calculations. Sequences with insufficient length are padded with pad sequences to ensure all sequences are the same length. The label corresponding to each sequence is converted from one number to one hot encoding format, which means that the six emotions correspond to six numbers, where the number corresponding to the emotion position is one, and the others are 0. The purpose of this is to prevent the original numerical magnitude relationship between 0 and 6 from being misinterpreted by the model as an ordered relationship, and the experiments now use one-hot coding so that each category is independent and does not affect each other. Finally, the total dataset was split into two parts, test and train, where the test accounted for 20%. The dataset has been processed, and the next step is to find the most suitable model based on the previously transformed dataset.

2.2. Model Architecture

The basic types of models used in this paper are simple RNN, Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), and there are also variations of them, such as Bidirectional LSTM (Bi-LSTM) and Bidirectional GRU (Bi-GRU). Simple RNN is the most basic recurrent neural network, but due to the problem of gradient vanishing or gradient exploding, an improved version of the RNN, which is also known as the LSTM, has been created by introducing input gates, forget gates and output gates to help the model remember the long dependencies and to help the model forget the long dependencies. The LSTM helps the model remember long dependencies by introducing input, forgetting, and output gates. GRU is a simplified version of LSTM that solves the same problem of gradient disappearance and gradient explosion, but the more straightforward structure also brings higher computational efficiency. Bidirectional LSTM and GRU are equivalent to two layers, one forward to deal with the sequence and the other backward. Learn the contextual content, which is more suitable for tasks requiring understanding comprehensive sequences [6]. The experiments are performed by adding Dropout (default 0.5) and Batch normalization behind each RNN layer to

normalize the data and thus speed up the computation to improve the stability. The Optimizer is Adam by default, and the loss function is categorical crossentropy.

The model's design is simple to deep from the simplest single-layer RNN and keeps on transforming, such as changing the simple RNN to LSTM or GRU and then increasing the number of layers. If the model performs well, the hyperparameters will continue to be adjusted as much as possible to find the best-performing model at the end. If the model shows overfitting or deterioration, the study will immediately change the direction of model modification. For example, stop increasing the number of layers and change to a different type of RNN or adjust other parameters.

2.3. Evaluation Metrics

This paper evaluates the model's strength through the following three indicators: Precision, recall, and f1 score. Precision is the proportion of all samples predicted to be positive, which is positive. Recall is the proportion of all positive samples correctly predicted to be positive. The F1 score is the reconciled average of precision and recall. The F1 score is the ratio of precision and recall to the F1 score. Recall's reconciled mean is that the larger the F1 score, the better the model performance is proved [7].

3. Experiments and Results

This section shows the specific steps and results of the experiments to evaluate the model's performance by comparing the accuracy, precision, recall, and F1 scores. As Table 1 summarizes the optimal performance data of the corresponding models obtained from this experiment under different structures. Among them, the two-layer Bi-LSTM model performs the best, receiving a test accuracy of 93.21% after 200 learning epochs.

Table 1. Result comparison of different models

Model	Accuracy	Precision	Recall	F1-Scores
Single-layer RNN	0.3013	0.30	0.30	0.30
Two-layer RNN	0.3599	0.33	0.36	0.30
Single-layer GRU	0.7867	0.79	0.79	0.79
Two-layer GRU	0.6718	0.78	0.67	0.65
Two-layer Bi-GRU	0.8833	0.89	0.88	0.88
Single-layer LSTM	0.7110	0.72	0.71	0.71
Two-layer LSTM	0.5988	0.65	0.60	0.59
Two-layer Bi-LSTM	0.9321	0.93	0.93	0.93

The specific experimental procedure starts from the simplest single-layer simple RNN, and the accuracy of the test model is only 0.3013, showing very serious overfitting. This may be due to the limited ability of the simple RNN model to deal with complex text categorization tasks and its limited ability to capture long-range dependencies in long texts [8]. The experiment tries to build a two-layer simple RNN to solve the overfitting problem, but the result is not apparent with only 0.3599. Even if experiments continue to add the simple RNN layer, the accuracy can be improved, but it cannot solve the problem effectively. The improvement is not obvious, but it will waste a lot of arithmetic resources. Therefore, this work decided to change the direction and try other, more complex RNN. This time, a comparison experiment is conducted. When all other parameters were the same, this work trained a single-layer LSTM and a GRU, respectively. The result is that the accuracy of single-layer GRU is 0.7867, which is significantly larger than that of single-layer LSTM, which is 0.7110. It has been shown that comparing single-layer LSTM with single-layer GRU is not an effective way to solve the problem. It has been shown that when comparing single-layer LSTM and GRU, where the structure of GRU is simpler, it reduces the computational complexity so that it can perform better when dealing with smaller datasets [9]. Therefore, the following experiment will be based on GRU. Table 2 shows the results of pursuing the optimal model by changing the number of layers, hyperparameters, and optimizer based on GRU.

Table 2. Result comparison of different GRU settings

Model	Accuracy	Precision	Recall	F1-scores
Two-layer Bi-GRU with L2 and LR decay	0.8217	0.83	0.83	0.83
Two-layer Bi-GRU with L2 no LR decay	0.8172	0.87	0.82	0.81
Two-layer Bi-GRU with RMSprop	0.8735	0.88	0.87	0.87
Two-layer Bi-GRU with Adam	0.8833	0.89	0.88	0.88
Two-layer Bi-GRU with SGD	0.3549	0.35	0.35	0.31

When choosing to add one layer of GRU to build a two-layer unidirectional GRU model, the test result is 0.6718, but it is worse because adding one layer of GRU leads to the complexity of the whole model is too high, which leads to the overfitting problem is more serious. Therefore, this work chose to change the first layer to bi-directional GRU, and the performance was outstanding, breaking through 0.8571 for the first time. This is because the bi-directional GRU can capture both the front and the back simultaneously to help the model better learn the context and improve classification performance. With the above content as the basis, the experiment chooses to adjust the current structure by incorporating L2 regularization in the two-way GRU and also adding learning rate attenuation to prevent overfitting. But instead of making the results worse, accuracy drops to 0.8217; if the learning rate stops decaying late fixed at 0.001, it will deteriorate further to 0.8172. From here, it can be seen that the learning rate attenuation and L2 regularization do not have any sound effect, so the experiments chose to give up the adjustment of the L2 regularization and the learning rate turn to start adjusting the optimizer, which has been used for the time being is Adam.

The subsequent study will be three controlled experiments with Adam, Root Mean Square Prop (RMSProp), and Stochastic Gradient Descent (SDG), with only the optimizer adjusted while keeping all other conditions the same. The results of the experiments show that Nadam has the best results under the same conditions with an accuracy of 0.8833, followed by RMSprop with 0.8735, and SDG has the worst results with 0.3549 overfitting. According to the study, Adam performs better in most cases because combining the advantages of Nesterov and Adam optimizers accelerates the convergence, and RMSprop also solves the problem of the other methods that only allow the learning rate to decrease by dynamically adjusting the learning rate [10]. After trying a single layer bi-directional GRU, the experiment chose to then try a bi-directional GRU without regularization and learning rate decay while choosing Nadam as the optimizer, which was the best performer before, the result was instead poor with an accuracy of only 0.8354. Therefore, this is the end of the study on GRU. Next, this work chose to replace the GRU in the model with LSTM after 50 epochs of learning; as shown in Figure 3 shows that the accuracy rate is improving and reached the current maximum of 0.8923. Therefore, this work continued the training for 100 epochs, and the result is improving. It broke through 90% for the first time, reaching 0.9237. Finally, after 200 epochs of learning, the accuracy rate came to the current experimental maximum of 0.9321. The highest accuracy of the experiment is 0.9321. Comparisons are demonstrated in Table 3.

Table 3. Result comparison of different epochs

Epochs	Accuracy	Precision	Recall	F1-scores
50	0.8923	0.89	0.89	0.89
100	0.9237	0.92	0.92	0.92
200	0.9321	0.93	0.93	0.93

4. Discussion

The experimental results show that Bi-LSTM performs better in text categorization tasks, proving that it performs better in learning extended text context. For now, bi-directional Bi-LSTM is the best-performing model among all the RNN structures studied. However, this work should also point out that there are many defects in this experiment, such as insufficient experimental data, not selecting several datasets for comparison, and there may be a situation where a single dataset is particularly

suitable for specific model learning. There are also fewer open-source datasets related to text sentiment categorization. Almost all of them are from Twitter, so there is the problem of poor applicability of the model in other occasions or environments.

5. Conclusion

The approach taken in this paper is more of a comparison experiment, where all other parameters are the same and only the required comparison parameters are changed, such as LSTM, GRU or some hyperparameters and then the optimizer. Through this comparison approach, it is found that the two-layer bidirectional LSTM model performs best in solving the sentiment classification problem in the current dataset, which is supposed to be due to the more complex model of LSTM and the addition of bi-directional allows the model to learn the context of the article better, which in turn improves the model's performance and finally achieves an accuracy of 93.21%. However, the accuracy of the model is still increasing when the training reaches 200 epochs, so it is expected to obtain a better model by training for a longer period of time in the future when enough resources could be achieved. It is also hoped to obtain text training sets from other platforms so that the model can be adapted to more environments and used in more situations.

References

- [1] Devlin Jacob, Chang Ming-Wei, Lee Kenton, et al. Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint, 2018: 1810.04805.
- [2] Liu Yinhan, Ott Myle, Goyal Naman, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint, 2019: 1907.11692.
- [3] Sanh Victor, Debut Lysandre, Chaumond Julien, et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint, 2019: 1910.01108.
- [4] Nidula Elgiriye withana, Emotions Dataset, Kaggle, URL: <https://www.kaggle.com/datasets/nelgiriye withana/emotions>. Last Accessed: 2024/08/02
- [5] Chen Wuxing, Yang Kaixiang, Yu Zhiwen, et al. A survey on imbalanced learning: latest research, applications and future directions. Artificial Intelligence Review, 2024, 57 (6): 1-51.
- [6] Shahid Farah, Aneela Zameer, Muhammad Muneeb. Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. Chaos, Solitons & Fractals, 2020, 140: 110212.
- [7] Sokolova Marina, Guy Lapalme. A systematic analysis of performance measures for classification tasks. Information processing & management, 2009, 45 (4): 427-437.
- [8] Minaee Shervin, Kalchbrenner Nal, Cambria Erik, et al. Deep learning--based text classification: a comprehensive review. ACM computing surveys, 2021, 54 (3): 1-40.
- [9] Cahuantzi Roberto, Xinye Chen, Stefan Güttel. A comparison of LSTM and GRU networks for learning symbolic sequences. Science and Information Conference, 2023: 771-785.
- [10] Maurya, Manju, Neha Yadav. A comparative analysis of gradient-based optimization methods for machine learning problems. International Conference on Data Analytics and Computing, 2022: 85-102.