SENTIMENT CLASSIFICATION WITH FOCAL LOSS ON IMBALANCED DATASETS

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND COMMUNICATIONS ENGINEERING,
THE GRADUATE SCHOOL OF FUNDAMENTAL SCIENCE
AND ENGINEERING
OF WASEDA UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF ENGINEERING

July 16th 2021

Fan Zhang
5119FG27-4

Advisor: Prof. Tetsuya Sakai
Research guidance: Research on Information Access

Abstract

Sentiment classification is an important task of Natural Language Processing. For better classification capability, a sufficiently large as well as high quality dataset is often required. But in real life, the datasets are often unbalanced, which leads to a decrease in the classification ability of the model. And in the task of object detection, focal loss is a common means to solve the data imbalance problem. But in the nlp domain, this method is not widely used. To explore the ability of focal loss to handle imbalanced data in the sentiment classification domain, we use focal loss to replace the original cross-entropy loss function of the model and investigated the performance of the classifier on three imbalanced datasets with different proportions. The experimental results show that focal loss improves the performance of the classifier compared to the original model, allowing the model to focus more on hard-to-classify samples.

Contents

Chapter 1

Introduction

With the increasing popularity of the Internet, more and more users like to express their opinions about products on e-commerce and rating sites. Most of these opinions are presented as texts containing sentiment tendencies, and determining the sentiment polarity of such texts is also an important task in natural language processing. Machine learning and deep learning approaches have achieved many advanced results for sentiment analysis of text on various types of datasets. Zhang et al. [1] used two classification methods based on lexicality and lexicon to get the semantic features, and then used an improved SVM algorithm to classify text, which achieved good results, but with a strong dependence on the lexicon selection. Tang [2] used neural networks to construct relationships between sentences and classify. But high performance depend on the size and quality of training data, which is very difficult to collect.

In real life, the datasets for sentiment classification are often unbalanced. In sentiment classification tasks, smaller numbers samples are more likely to be misclassified which leads to a decrease in the performance of the model. To reduce the effect of sample imbalance on classification results, focal loss [3] is an effective method. In the target detection task, it can reduce the weights of few, easy samples thus solving the problem of sample imbalance. This method is less applied in sentiment classification tasks, so this paper is to investigate the effect of focal loss on data imbalance problem in sentiment classification tasks.

We used three neural network models TextCNN, Bi-LSTM, and Bi-LSTM with attention layer. Focal loss was used to replace the cross-entropy loss function of the models and to perform sentiment classification on three imbalanced datasets with different proportions of data and to compare with the original model. We observe that the F_1 scores of the model with focal loss are higher than the original model at different proportions. And as the gap between the number of samples of different categories increases, the F_1 scores of focal loss increases more and more. It indicates that focal loss makes the model pay more attention to the hard samples, thus improving the performance of the classifier.

Chapter 2

Related Work

There have been many studies so far to address the problem of data imbalance in text classification.

2.1 Data Augmentation

EDA [4] augments the data by editing the original sentences. It include 4 operations: synonym replacement, random insertion, random swap, and random deletion. There is a significant performance when the datasets is small and it is very simple to implement.

Edit Transformer [5] applies edits learned on a source domain with plentiful data to a data-constrained target domain. Then the generative model is used to generate new sentences to augment data.

Kobayashi [6] proposes a context-based approach for data augmentation. Given the context of the replacement word, the language model is used to predict the word, and the new word is substituted for the original word to obtain data augmentation. And remaining label information to ensure syntactic consistency.

2.2 Machine Learning

Zheng Z. [7] extracts features from the negative and positive samples separately and combined. Then they used Multinomial Naive Bayesian and regularized Logistic regression functions as classifiers.

BABoost [8] is an improved AdaBoost algorithm, which gives higher weights to the misclassified examples from the minority class. It decreases the prediction error of minority class significantly with increasing the prediction error of majority class a little bit.

Chapter 3

Proposed Methods

We use TextCNN, Bi-LSTM, and Bi-LSTM with attention layer as Baseline. Focal loss is used to replace the original cross-entropy loss function and obtaining improved models.

3.1 TextCNN

TextCNN [9] is a text convolutional neural network with parallel convolutional structure. Figure 3.1 shows the structure.

The input to the network is a word-based document. Each word is represented by a word vector such that the whole document is mapped as a matrix of size $n \times d$, where n is the number of words in the document and d is the dimension of the word vector. Thus all documents have the same matrix dimension $X \in R^{(n' \times d)}$. Then the convolution operation is performed on the matrix. And a maximum pooling operation is used for the pooling layer of the convolutional neural network to extract the most important features of each convolution and reduce the dimension of the text vector. Finally, c training set data are obtained in the pooling layer as $\{x^{(1)}, y^{(1)}, x^{(2)}, y^{(2)}, ..., x^{(c)}, y^{(c)}\}$, where $x^{(i)}$ is text feature vector, $y^{(i)}$ is text category, $i \in \{1, 2, ..., k\}$. Finally the classification is done by a fully connected softmax layer.

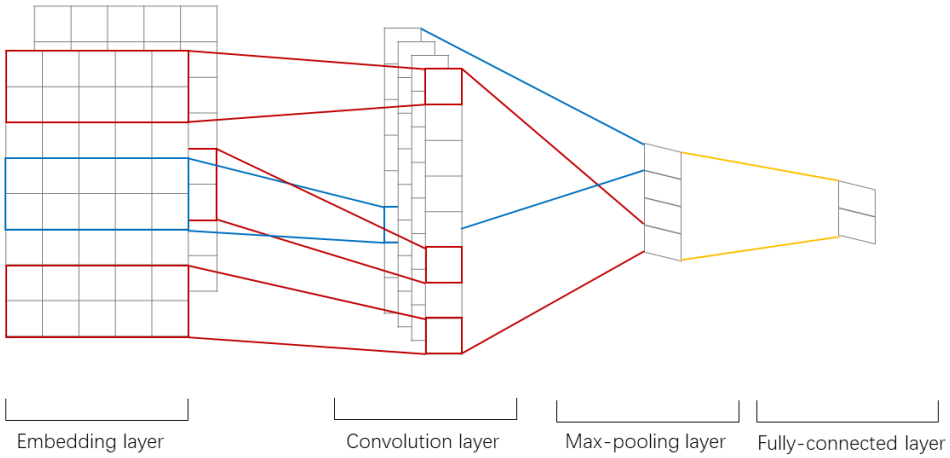

Fig. 3.1: The structure of TextCNN

3.2 Bi-LSTM

LSTM [10] is a special kind of recurrent neural network, which is a neural network that processes continuous data by sharing internal weights in a sequence, and which uses the current word vector and its previous hidden states to compute the next hidden state.

Bi-LSTM was proposed by Graves et al. [11] It is a combination of forward LSTM and backward LSTM. By this structure, Bi-LSTM could solve the problem of LSTM which is difficult to encode information from back to front. Figure 3.2 shows the structure of Bi-LSTM. After forward LSTM inputs I_0, I_1, I_2, vector $[\vec{h_0}, \vec{h_1}, \vec{h_2}]$is obtained.After backward LSTMinputs I_0, I_1, I_2, vector $[\overleftarrow{h_0}, \overleftarrow{h_1}, \overleftarrow{h_2}]$is obtained. Then the forward and backward vectors are concatenated to be vector $[h_0, h_1, h_2]$.

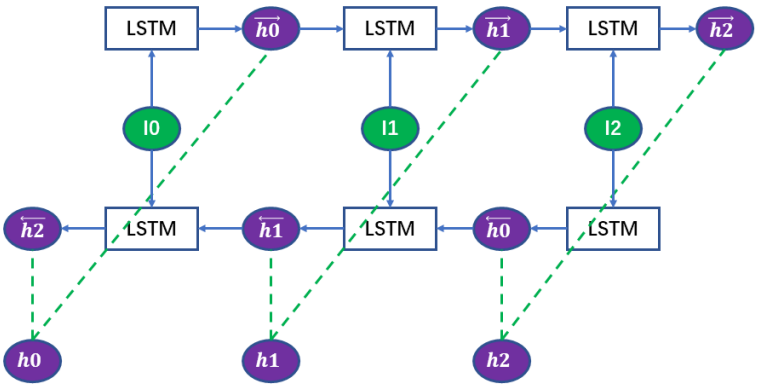

Fig. 3.2: The structure of Bi-LSTM

3.3 Attention Layer

The Attention mechanism [12] has been used in a wide variety of natural language processing tasks. It is a approach that mimics human attention, which means that it can helps the model to learn which part of text should be paid more attention, resulting in more reasonable sentence representations. A very important application of the attention mechanism is to use it with the LSTM model. It can be used to solve the problem about the difficulty in obtaining a final reasonable vector representation when the input sequence of the LSTM model is long.

We use the hidden outputs of Bi-LSTM as the feature vector. Then we take these feature vectors as input and compute the weight vectors in the attention layer. The calculation formula is shown below.

$$\begin{aligned} U &= \tanh(H) \\ \alpha &= softmax(W^T U) \\ r &= H\alpha^T \end{aligned} \tag{3.1}$$

where H is a matrix of output vectors that produced by Bi-LSTM, W is the weight vector, the representation r is the output vector of this attention model.

3.4 Focal Loss

The Focal loss [3] function is an improved function based on the standard cross-entropy, which makes the model focus more on hard-to-classify samples during training, thus increas-

ing the F_1 scores of a small number categories in the model. The standard cross-entropy is

$$CE(p, y) = CE(p_t) = -log_2(p_t) \tag{3.2}$$

where $p_t = \begin{cases} 0 & \text{y} = 1 \\ 1 - p & \text{others} \end{cases}$, p is the expected output, y is the actual output.

The function FL of focal loss is

$$FL(p_t) = -(1 - p_t)^{\gamma} log_2(p_t) \tag{3.3}$$

where $(1 - p_t)^{\gamma} (\gamma \geq 0)$ is a modulating factor. The focal loss has two properties. (1)When a sample is misclassified and p_t is small, the modulating factor is close to 1 and is same as the cross-entropy function. When p_t close to 1, the factor becomes to 0 and the contribution to the total loss becomes smaller. (2)When γ goes to 0, the FL is equivalent to CE. In other words, focal loss is a function to measure the contribution of hard samples and easy samples to the total loss, thus solving the category imbalance problem.

But to solve the quantity imbalance problem, the α-balanced variant needs to be added. And the improved function is

$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} log_2(p_t) \tag{3.4}$$

where α_t is used to control the shared weight of positive and negative samples on the total loss.

Chapter 4

Experiments

4.1 Datasets

We use reviews from DianPing, Amazon and DMSC[*1] as the datasets. The datasets contains the ratings, and we classify the dataset into negative and positive categories based on the ratings. For DMSC and Amazon, ratings of 1 are set as negative samples and ratings of 5 are set as positive samples. For DianPing, ratings of 0 are set as negative samples and ratings of 5 are set as positive samples. The negative samples are labeled with 0 and the positive samples are labeled with 1. Only the data with text length less than 100 were kept, and finally a total of 25436 negative samples and 26040 positive samples were obtained. And three sets of unbalanced datasets were constructed by randomly selecting samples according to the ratio of negative samples to positive samples of 1:2, 1:5 and 1:10. The training set and validation set and test set are splited according to the ratio of 8:1:1.

4.2 Experiment Settings

We preprocessed the datasets by word separation and removing the stop words. Then the word vectors are trained by Word2vec. We used TextCNN, Bi-LSTM, and Bi-LSTM with attention layer as the baselines and compared with the model using focal loss on the three unbalanced datasets. According to several experiments, we found that the models get the best performance when the $\gamma = 2$. And we used the proportions between categories as α.

4.3 Evaluation Metrics

Because precision and recall can not evaluate the performance of the model well on unbalanced datasets, and F_1 score is a harmonic mean of precision and recall, and a high F_1 score can be obtained only when both precision and recall are high. Therefore F_1 score is chosen as the evaluation metrics.

[*1] https://github.com/SophonPlus/ChineseNlpCorpus

Chapter 5

Results and Discussions

Table 5.1 shows the result. We can observe that as the number of negative samples decreases, the F_1 scores of the negative samples gradually decrease, but the F_1 scores of the positive samples all remain high and increase by a certain amount. It indicates that the more unbalanced the data is, the more influence it has on the classifier. The model using focal loss has approximately the same F_1 scores for positive samples compared to baseline. The F_1 scores for negative samples improves under different datasets, and the improvement increases gradually as the number of negative samples decreases. It shows that the model reduces the weights of the easy samples and makes the model focus more on the hard samples, thus improving the classification ability of the smaller number samples.

Bi-LSTM achieves the best results compared to other models for data ratios of 1:2 and 1:10, and Bi-LSTM with attention layer achieves the best results for data ratio of 1:5. And when the ratio is 1:10, Bi-LSTM with focal loss achieves the biggest improvement of 5.28. This illustrates that focal loss can effectively solve the hard classification problem when the samples are not balanced.

Table 5.1: The results of three models with and without focal loss on imbalanced datasets with positive and negative class ratios of 1:2,1:5,1:10. N is negative sample. P is Positive sample.

		N:P = 1:2		N:P = 1:5		N:P = 1:10	
		BL	FL	BL	FL	BL	FL
TextCNN	N	46.85	47.46	31.63	32.71	21.23	23.90
	P	83.07	83.01	91.45	91.39	95.54	95.51
Bi-LSTM	N	48.58	**49.94**	32.20	34.17	23.15	**28.43**
	P	82.91	82.79	91.35	91.33	95.56	95.49
Bi-LSTM+Att	N	48.47	49.41	32.85	**34.31**	24.03	27.69
	P	82.87	82.97	91.33	91.31	95.51	95.48

Chapter 6

Conclusion

We constructed three imbalanced datasets of different proportions. Then we used the focal loss function to replace the cross-entropy loss function and conducted experiments on these three datasets to test the ability of focal loss to handle the problem of unbalanced datasets in sentiment classification tasks. The experimental results show that focal loss solves the problem of the contribution of positive and negative sample weights to the loss when the datasets are unbalanced, and improves the classification ability of the model.

Chapter 7

Future Work

For future work, we can go further in two directions: (1)Since sentiment classification tasks can involve multiple sentiment polarities, we need to investigate solutions for unbalanced datasets in multiple classification tasks. (2)Combining focal loss with other methods to further improve the accuracy of sentiment classification.

Acknowledgements

This research was undertaken with the support from The Real Sakai Laboratory [13], Waseda University. I would like to express my thanks of gratitude to members in The Real Sakai Laboratory, who helped me and gave me many useful pieces of advice for continuing the research. Also, I would like to thank Professor Tetsuya Sakai, who gave me this great opportunity and lots of supports to do research on this topic.

References

[1] Dongwen Zhang, Hua Xu, Zengcai Su, and Yunfeng Xu. Chinese comments sentiment classification based on word2vec and svmperf. *Expert Systems with Applications*, 42(4):1857–1863, 2015.

[2] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432, 2015.

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[4] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[5] Guillaume Raille, Sandra Djambazovska, and Claudiu Musat. Fast cross-domain data augmentation through neural sentence editing. *arXiv preprint arXiv:2003.10254*, 2020.

[6] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*, 2018.

[7] Zhaohui Zheng, Xiaoyun Wu, and Rohini Srihari. Feature selection for text categorization on imbalanced data. *ACM Sigkdd Explorations Newsletter*, 6(1):80–89, 2004.

[8] Jie Song, Xiaoling Lu, and Xizhi Wu. An improved adaboost algorithm for unbalanced classification data. In *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 1, pages 109–113. IEEE, 2009.

[9] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

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

[11] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[13] RSL. The sakai laboratory. Last updated: 26th March 2019.