Frontiers in Signal Processing, Vol. 5, No. 1, January 2021
https://dx.doi.org/10.22606/fsp.2021.51003

Improved Dynamic Routing Algorithm for Information Aggregation

Gongbin Chen, Wei Xiang and Yansong Deng*

Key Laboratory of Electronic and Information Engineering, Southwest Minzu University, State Ethnic Affairs Commission, Chengdu, China
Email: cgb727@foxmail.com

Abstract Information aggregation is an essential component of text encoding, but it has been paid less attention. The pooling-based (max or average pooling) aggregation method is a bottom-up and passive aggregation method, and loses a lot of important information. Recently, attention mechanism and dynamic routing policy are separately used to aggregate information, but their aggregation capabilities can be further improved. In this paper, we proposed an novel aggregation method combining attention mechanism and dynamic routing, which can strengthen the ability of information aggregation and improve the quality of text encoding. Then, a novel Leaky Natural Logarithm (LNL) squash function is designed to alleviate the "saturation" problem of the squash function of the original dynamic routing. Layer Normalization is added to the dynamic routing policy for speeding up routing convergence as well. A series of experiments are conducted on five text classification benchmarks. Experimental results show that our method outperforms other aggregating methods.

Keywords: information aggregation, dynamic routing, attention, squash function, text classification

1 Introduction

A primary challenge is how to encode text sequence in natural language processing. The process of text encoding is how to encode the variable-length text sequence into a fixed-length vector, which should fully capture the semantics of text. Generally, many successful text encoding methods contain three key steps: word embeding, feature representation and information aggregation. In past studies, more attention has been paid to the first two steps, while the critical aggregation step has obtained less attention. The information aggregation is the process of summarizing previously extracted feature information into a fixed-length vector.

The traditional information aggregation methods utilize max pooling or avg pooling to sum up the output of Convolutional Neural Network (CNN) or Recurrent Neural Network (RNN) layers [2,3,9]. In the process of max pooling, only the most active neurons will be selected to pass through the next layer, this is the reason why valuable spatial information between layers is lost. Although pooling can significantly reduce the number of parameters and maintain a certain invariance (rotation, translation, expansion, etc.), it loses a lot of important information, which have an enormous influences on the performance of downstream tasks.

With the popularity of attention mechanism [12,23], researchers also use attention mechanism to replace pooling for information aggregation. The attention mechanism imitates the human visual system, selectively focusing on the useful part of all information, while ignoring other visible information. For example, when people are reading, usually a few words are paid attention to and processed. The attention mechanism has two main aspects: deciding which part of the input needs to be paid attention to; allocating limited information processing resources to the important part. When processing text information, the attention mechanism assigns a relevance value to the words and sentences, telling how relevant they are for the task at hand. Although the attention mechanism can handle with long text sequence, it will generate redundant information.

In recent promising work of capsule network [19], a dynamic routing policy is proposed and proven to be more effective than the max pooling. The capsule network was first proposed to overcome the limitations and deficiencies of CNN, especially its insensitivity to spatial information and the loss of a lot

of information after pooling. A metaphor (also as an argument) they made is that human visual system intelligently assigns parts to wholes at the inference time without hard-coding patterns to be perspective relevant. Dynamic routing policy was first used to encode the intrinsic spatial relationship between a part and a whole of the image. However, when dealing with text information, the dynamic routing policy which dynamically decides that what and how much information need be transferred from each word to the final encoding of the text sequence could effectively avoid the loss and the redundancy of feature information. The dynamic routing is a up-bottom, more active aggregation method than pooling and attention mechanism. Due to the good characteristics of dynamic routing, some researchers have begun to use it for information aggregation and have obtained good results [4].

In this paper, in order to solve some problems of original dynamic routing policy [19], we propose the improved dynamic routing policy to aggregate information. The contributions of this work are presented as follows:

- The attention mechanism and the dynamic routing policy are combined to aggregate information, which can obtain better text encoding vectors.
- We propose a novel Leaky Natural Logarithm (LNL) squash function, which can alleviate the "saturation" phenomenon of the squash function in original dynamic routing.
- Layer Normalization is added to our routing policy, which enables routing to converge faster.
- Experimental results on five text classification tasks indicate that our aggregation method outperforms other aggregation methods by a significant margin.

The rest of our paper is structured as follows: Section 2 discusses related works, Section 3 gives a detailed description of our model, Section 4 shows the experimental setup, Section 5 reports and discusses the obtained results, and Section 6 summarizes this work and the future direction.

2 Related Work

In many text encoding models, there are three methods for aggregating information: pooling, attention, and dynamic routing. For text classification tasks, many traditional CNN/RNN models that use pooling to aggregate information have a simple structure [9,13]. [7] proposes a CNN model that provides an alternative mechanism for effective use of word order for text categorization through direct embedding of small text regions, different from the traditional bag-of-n-gram approach or word-vector CNN. [21] address this issue that traditional models only use semantics of texts by incorporating user-level and product-level information into a neural network approach for document level sentiment classification. With the emergence of various problems, more complex neural network models have emerged to handle with text classification tasks [5,6,22,25].

Recently, with the popularity of the attention mechanism [12,23,27], more and more models that use attention mechanism to aggregate information have emerged in endlessly. [26] proposes an Attention-based Long Short-Term Memory Network which establishes the connection between an aspect and the content of a sentence for aspect-level sentiment classification. [14] proposes an attention-gated convolutional neural network (AGCNN) for sentence classification, which makes full use of limited context information to extract and enhance the influence of important features in predicting sentence categories. [18] proposes two improved Hierarchical Attention Networks (HAN) models, which solve the noise problem caused by irrelevant words or sentences and can better handle with important distributions. The attention mechanism based aggregation collects information in a bottom-up way without considering the state of the final encoding.

With the emergence of capsule network [19], its ability of text processing has also been explored by many researchers. Different from the pooling and the attention mechanism, dynamic routing can determine that what and how much information need be transferred based on specific tasks, rather than passively accepting information. [4] uses original dynamic routing to aggregate information and proposes hierarchical routing to classify documents. [28] proposes three strategies to stabilize the dynamic routing process to alleviate the disturbance of some noise capsules which may contain "background" information. Due to these advantages of dynamic routing, applications in the field of text processing will also increase.

3 Our Model

Generally, these models of text classification task mainly consists of the following parts: Embedding Layer, Feature Extraction Layer, Aggregation Layer and Prediction Layer. Our model is shown in Fig.1.

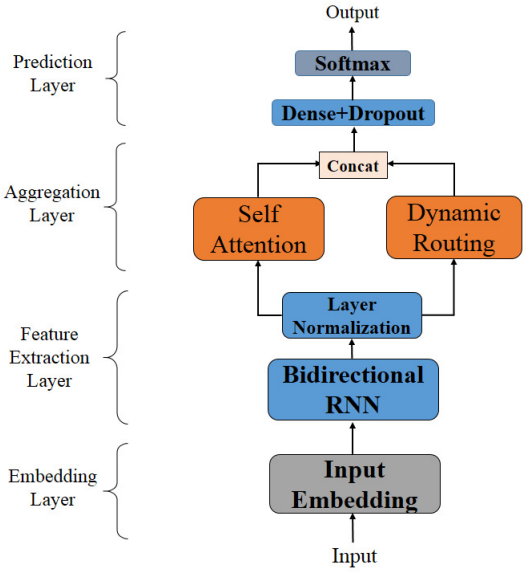

Figure 1. Combination of Self Attention and Dynamic Routing for information aggregation.

3.1 Embedding Layer

In order to obtain knowledge from a vast unlabeled corpus, the embeddings can be taken from the pre-trained word embedding, such as Glove [16], Word2vec [15]. We map each word into a N dimensional embedding vector, then transport it to the next layer.

3.2 Feature Extraction Layer

Usually, CNN and RNN are commonly used to extract features information. CNN can capture the local information of the text sequences and train in parallel, but cannot obtain the related information between adjacent words. Although RNN cannot be parallelized like CNN, it can capture semantic information between adjacent words and deal with the dependency of long text sequences. In our model, in order to obtain better inter-word information, we use BiLSTM/BiGRU layer as the feature extraction layer to incorporate forward and backward context information of a sequence.

Studies have shown that Layer Normalization[1] can reduce the impact of covariate shift by fixing the input mean and variance of a layer of neurons. Therefore, we choose the Layer Normalization to normalize output of the RNN layer.

3.3 Aggregation Layer

This layer is the most important part of our model. Although the attention mechanism and the dynamic routing policy have excellently independent information aggregation capabilities, there is not an attempt to combine the two way to aggregate information. We use a combination of attention mechanism and dynamic routing policy to aggregate the feature information from the previous layer.

Attention Mechanism The attention mechanism assigns attention weights according to the importance of input words to output nodes. A few input words that are crucial to the output will be emphasized while the other words will be ignored. Generally, the procedure of the attention mechanism is as follows:

$$b_i = \mathbf{q}^{\mathrm{T}} \cdot \mathbf{g}_i \tag{1}$$

$$a_i = \frac{\exp(b_i)}{\sum_j \exp(b_j)} \tag{2}$$

$$\mathbf{v}_{\mathrm{att}} = \sum_{i=1}^{\mathrm{T}} a_i \cdot \mathbf{g}_i \tag{3}$$

where, the trainable query $\mathbf{q}$ is used to calculate the similarity weight with each word in the context, $\mathbf{g}$ represents the output of the previous layer, $a \in (0, 1)$ means attention weight, $\mathbf{v}_{\mathrm{att}}$ is the sum of the final attention score of each context word.

Dynamic Routing The basic idea of dynamic routing is to construct a non-linear map in an iterative manner ensuring that the output of each capsule gets sent to an appropriate parent in the subsequent layer:

$$\left\{\hat{u}_{n|m} \in R^d\right\}_{m=1,\ldots H, n=1,\ldots N} \overset{routing}{\longrightarrow} \left\{\hat{v}_{n|m} \in R^d\right\}_{n=1}^{N} \tag{4}$$

Table 1. Dynamic Routing Algorithm.

Algorithm 1: Dynamic Routing Algorithm
1 **procedure** Routing($\hat{\mathbf{u}}_{n\|m}$,t,L)
2 Initialize the coupling coefficients b_{mn}=0
3 **for** t iterations **do**
4 for all capsule m in layer L: a_{mn} = **Softmax**(b_{mn}) ⇒ **Softmax** computes Eq.7
5 for all capsule n in layer (L+1): $\mathbf{s}_n = \sum_m a_{mn}\ \hat{\mathbf{u}}_{n\|m}$
6 for all capsule n in layer (L+1): $\mathbf{p}_n$ = **LN**($\mathbf{s}_n$) ⇒ **LN** denotes Layer Normalization
7 for all capsule n in layer (L+1): $\mathbf{v}_n$ = **Squash** ($\mathbf{p}_n$) ⇒ **Squash** computes Eq.9
8 for all capsule m in layer L and capsule n in layer (L+1): $b_{mn} = b_{mn}+\hat{\mathbf{u}}_{n\|m} \cdot \mathbf{v}_n$
return $\mathbf{v}_n$

The dynamic routing policy is the process of iteratively updating the log probabilities b_{mn} to generate the coupling coefficient a_{mn}. The specific algorithm is as Algorithm 1.

According to the definition of capsule network [19], we call each encoding vector, or a group of neurons, as a capsule.

First, for all capsule m in the bottom layer, all "prediction vectors" $\hat{\mathbf{u}}_{n|m}$ are produced by multiplying the output $\mathbf{u}_m$ of a capsule by a weight matrix $\mathbf{W}_{mn}$

$$\hat{\mathbf{u}}_{n|m} = \mathbf{W}_{mn}\mathbf{u}_m \tag{5}$$

Then, for all capsule n in the bottom layer, a capsule $\mathbf{s}_n$ is a weighted sum over all "prediction vectors" $\hat{\mathbf{u}}_{n|m}$

$$\mathbf{s}_n = \sum_m a_{mn}\hat{\mathbf{u}}_{n|m} \tag{6}$$

where the a_{mn} are coupling coefficients that are determined by the iterative dynamic routing process.

While, the coupling coefficient a_{mn} is calculated by

$$a_{mn} = \frac{\exp(b_{mn})}{\sum_k \exp(b_{mk})} \tag{7}$$

The coefficients between capsule m and all the capsules in the layer above sum to 1 and are computed by a softmax function. It represents the probability that the bottom capsules are routed to the top capsules.

After that, we adopt Layer Normalization which can be able to accelerate routing convergence for each capsule $\mathbf{s}_n$ in the top layer

$$\mathbf{p}_n = \mathbf{LayerNormalization}(\mathbf{s}_n) \tag{8}$$

and then squashes $\mathbf{p}_n$ to confine $|\mathbf{p}_n| \in(0,1)$ to a probability

$$\mathbf{v}_n = \frac{\ln(1+\lambda\|\mathbf{p}_n\|)}{1+\ln(1+\lambda\|\mathbf{p}_n\|)} \frac{\mathbf{p}_n}{\|\mathbf{p}_n\|} \tag{9}$$

where, $\lambda \in(0,1]$ is the Leaky coefficient, $\mathbf{v}_n$ is the vector output of n capsule in the top layer.

Finally, the log probabilities b_{mn} is updated by

$$b_{mn} = b_{mn} + \hat{\mathbf{u}}_{n|m} \cdot \mathbf{v}_n \tag{10}$$

where, $\hat{\mathbf{u}}_{n|m} \cdot \mathbf{v}_n$ represents the dot product of $\hat{\mathbf{u}}_{n|m}$ and $\mathbf{v}_n$, which means similarity between the bottom capsules and the top capsules.

Leaky Natural Logarithm squash function. Aiming at the "saturation" problem of the original squash function, we propose a Leaky Natural Logarithm squash function as shown in Eq.9. We assume that the input of the squash function is a scalar x, then the original squash function and the Leaky Natural Logarithm squash function are computed by

$$squash_{Original}(x) = \frac{x^2}{1+x^2} \tag{11}$$

$$squash_{LNL}(x) = \frac{\ln(1+\lambda|x|)}{1+\ln(1+\lambda|x|)} \tag{12}$$

We draw the curves of these two functions and the curves of their gradients in a two-dimensional coordinate system as shown in Figure 2 and Figure 3.

From the Figure 2, we can observe the curve of the original squash function [19] is relatively steep in the early stage, then it tends quickly to be stable. In other words, the original squash function is easy to reach "saturation" status. When the model is trained by backpropagation, the gradients of those neurons with larger values will tend to 0, and the trend of the gradient can be seen from the Figure 3, which makes it difficult to update the parameters in the Adam algorithm [10], leading to the poor performance of the model.

However, the curve of the Leaky Natural Logarithm squash function is relatively gentle throughout as shown in Figure 2, so it can postpone the "saturation" status of the original squash function. As can be seen from Figure 3, the Leaky Natural Logarithm squash function still has a certain gradient compared with the original squash function when the input is large or small, which can ensure that the model is better trained.

Combination of Attention and Dynamic Routing In our model, we use a combination of attention mechanism and dynamic routing policy as aggregation layer. The two methods are combined by

$$\mathbf{v} = \text{Concat}[\mathbf{v}_{att}, \mathbf{v}_{rout}] \tag{13}$$

where, $\mathbf{v}_{att}$ and $\mathbf{v}_{rout}$ are the output vector of the Attention and the Dynamic Routing. Concat is a way of dimensional splicing.

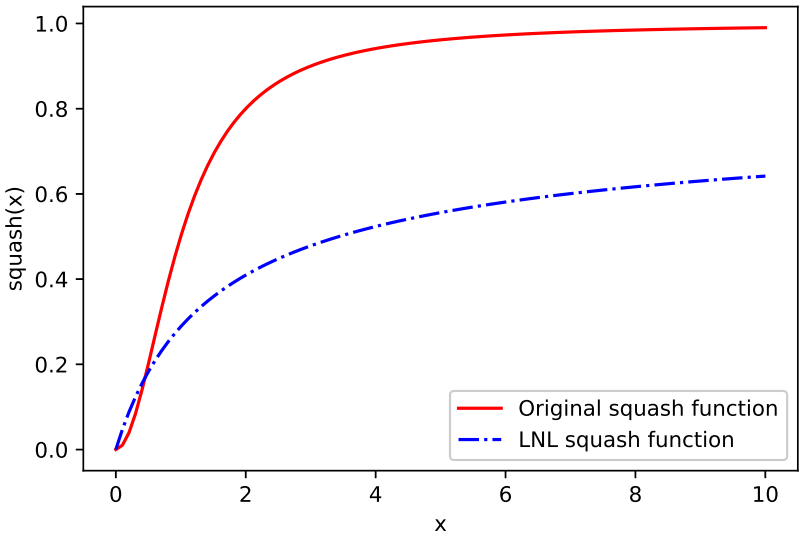

Figure 2. Original squash function and Leaky Natural Logarithm squash function in the two-dimensional coordinate system..

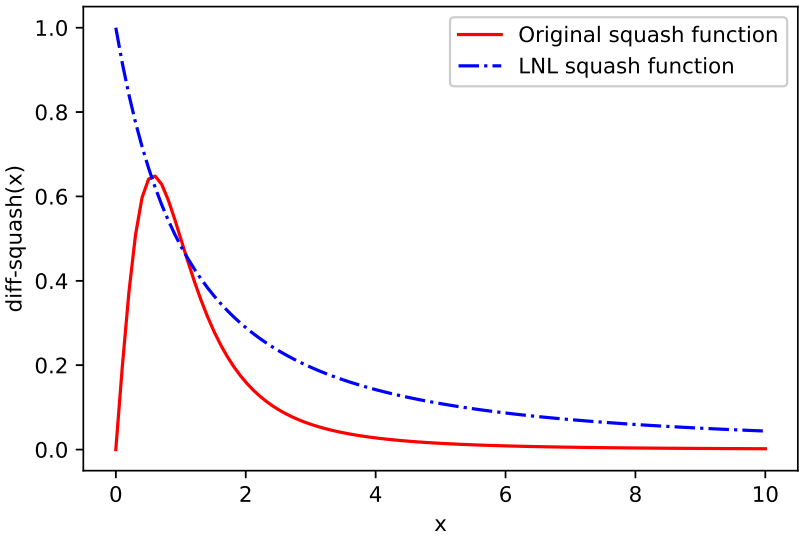

Figure 3. Gradient of Original squash function and Leaky Natural Logarithm squash function in the two-dimensional coordinate system.

3.4 Prediction Layer

We feed output of the Aggregation Layer to the input of a Dense (or Fully connected) layer with dropout, followed by a softmax classifier.

$$\mathbf{p}(\bullet|\mathbf{v}) = \text{Softmax}(\text{Dense}(\mathbf{v})) \tag{14}$$

where, $\mathbf{v}$ is output of the Aggregation Layer. $\mathbf{p}(\bullet|\mathbf{v})$ represents the predicted distribution of different classes given the representation vector $\mathbf{v}$.

4 Experimental Setup

4.1 Datasets

To evaluate the effectiveness of our model, we conduct a series of experiments on five benchmark datasets, which are also provided in [4]. These benchmark datasets include two sentence levels (SST-1, SST-2) and three document levels (IMDB, Yelp-13, Yelp-14). The detailed statistics are presented in Table 2.

- **SST-1** Stanford Sentiment Treebank is a movie review dataset [20].
- **SST-2** Binary-class version of SST-1.
- **IMDB** Movie review dataset extracted from IMDB website.
- **Yelp-13** and **Yelp-14** are reviews from Yelp.

Table 2. Characteristics of the datasets.

Dataset	Trainset	Devset	Testset	Classes
SST-1	8.5k	1.1k	2.2k	5
SST-2	6.9k	0.8k	1.8k	2
IMDB	67.4k	8.4k	9.1k	10
Yelp-13	62.5k	7.8k	8.7k	5
Yelp-14	183k	22.7k	25.4k	5

4.2 Implementation Details

In the experiments, we use 300-dimensional Glove [16] vectors to initialize embedding vectors. We conduct mini-batch with size 64 for SST-1 and SST-2, size 32 for other datasets. We use 0.5 dropout rate for SST-2 and 0.2 dropout rate for other datasets. We utilize Cross entropy loss function and Adam optimization algorithm [10] with 1e-3 learning rate to train the model. We use 3 iteration of routing for our model. In addition, our experiment was implemented on a NVIDIA 21080Ti.

4.3 Competitor Models

In the experiments, we mainly evaluate and compare our model with several strong baseline methods including:

- **CNN-non-static** Convolutional Neural Network [9].
- **DCNN** Dynamic Convolutional Neural Network with dynamic k-max pooling [8].
- **PV** Logistic regression on top of paragraph vectors [11].
- **MT-LSTM** Multi-Timescale Long Short-Term Memory neural network [13].
- **UPNN** User Product Neural Network to capture user- and product-level information [21].
- **AGCNN** Attention-Gated Convolutional Neural Network [14].
- **DR-AGG** Dynamic Routing Aggregation [4].

5 Experimental Results

In our experiments, the evaluation metric is classification test accuracy. We summarize the experimental results in Table 2. We mainly compare with the DR-AGG [4] model using the original dynamic routing. In addition, we also compare the effect of adding Layer Normalization in dynamic routing.

From the results, we observe that our model achieve best results on 5 benchmarks. When only using original routing with CAR method, it has been greatly improved on the IMDB dataset, and

Table 3. Comparison of different methods for information Aggregation.

Model	SST-1	SST-2	IMDB	Yelp-13	Yelp-14
CNN-non-static	48	87.2	-	-	-
DCNN	48.5	86.8	-	-	-
Paragraph-Vector	48.7	87.8	-	-	-
MT-LSTM (F2S)	49.1	87.2	-	-	-
UPNN(full)	-	-	43.5	59.6	60.8
AGCNN	49.6	87.6	-	-	-
DR-AGG	50.5	87.6	45.1	62.1	63
Original squash+CAR (No LN)	49.8	87.9	47.8	62.8	63.2
Original squash+CAR (LN)	50	88.5	48.1	62.7	64
Our squash+CAR (No LN, λ=0.3)	49.9	88.4	48.7	63.1	63.8
Our squash+CAR (LN, λ=0.3)	**50.7**	**89.2**	**48.8**	**63.6**	**64.1**

CAR: Combination of Attention and Dynamic Routing. LN: Layer Normalization. λ: Leaky coefficient.

its performance on other datasets has almost the same effect as DR-AGG. However, we added Layer Normalization to original routing, and the improvement on the two datasets of SST-2 and IMDB is obvious. Separately speaking, the original routing with CAR method improves ranging from 0.3% to 2.7% compared with DR-AGG (despite poor performance on SST-1); the original routing with CAR method and Layer Normalization improves ranging from 0.9% to 3% compared with DR-AGG (despite poor performance on SST-1). This shows that Layer Normalization is effective for dynamic routing.

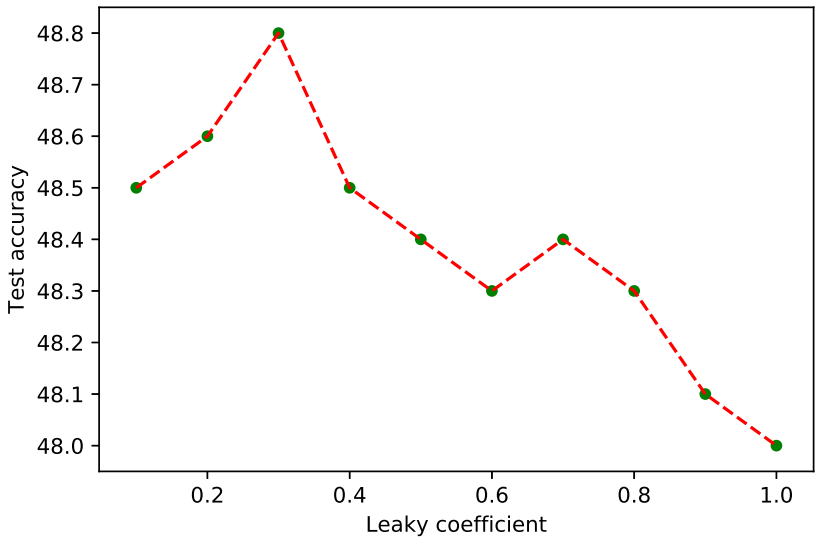

Figure 4. Relationship between Test accuracy and Leaky coefficient λ on the IMDB dataset, where the horizontal axis denotes Leaky coefficient λ, and the vertical axis denotes Test accuracy.

Furthermore, when use the CAR method that joins Layer Normalization and our squash function, our model outperforms the DR-AGG approach by 0.2%, 1.6%, 3.7%, 1.5%, and 1.1% on SST-1, SST-2, IMDB, Yelp 2013 and Yelp 2014. Moreover, our method is also better than other aggregation methods. For instance, our model improves ranging from 0.9% to 2.7% and 1.6% to 2% compared with CNN-non-static,

MT-LSTM and AGCNN on SST-1 and SST-2. It empirically shows that our proposed the improved dynamic routing policy is the best effective method on aggregating information.

Effect of Leaky coefficient. Inspired by Leaky Relu activation function, a Leaky coefficient λ is introduced to our squash function. Leaky coefficient λ can affect the updating amplitude of the parameters by adjusting the gradient of the squash function in the backpropagation, thereby affecting the result of the model. Therefore, seeking a suitable λ can improve the performance of the model to a certain extent. In our experiment, we set the value of λ from 0 to 1.

In order to search the optimal λ, we explored the relationship between Test accuracy and Leaky coefficient λ on the IMDB dataset as shown in Figure 4. We can observe that our model reach the best performance on IMDB dataset when λ is almost 0.3. Furthermore, the optimal value of λ may vary with different downstream tasks or different datasets.

6 Conclusion and Future Work

In this paper, we propose an aggregation method that combines attention mechanism and dynamic routing policy, which can take advantage of their respective characteristics to obtain better coding vectors. In addition, we propose two strategies to optimize our dynamic routing. The Leaky Natural Logarithm squash function effectively alleviates the "saturation" problem of the original squash function, and adding Layer Normalization is also proven to be effective. Experimental results of five text classification tasks show that our model outperforms other baseline models by a significant margin.

There are more and more applications of the capsule network in the field of text processing, and many effective routing policies have been proposed [17,24]. In the future, we will continue to explore the capabilities of dynamic routing in text processing.

References

1. Ba J L, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016.
2. Chung J , Gulcehre C , Cho K H , et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. Eprint Arxiv, 2014.
3. Conneau A , Schwenk H , Barrault L , et al. Very Deep Convolutional Networks for Text Classification[J]. 2017.
4. Gong J , Xipeng Qiu, Wang S , et al. Information Aggregation via Dynamic Routing for Sequence Encoding[J]. 2018.
5. Jiang M , Zhang W , Zhang M , et al. An LSTM-CNN attention approach for aspect-level sentiment classification[J]. Journal of Computational Methods in ences and Engineering, 2019, 19(4):1-10.
6. Johnson R , Zhang T . Deep Pyramid Convolutional Neural Networks for Text Categorization[C]. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017.
7. Johnson R, Zhang T. Effective use of word order for text categorization with convolutional neural networks[J]. arXiv preprint arXiv:1412.1058, 2014.
8. Kalchbrenner N , Grefenstette E , Blunsom P . A Convolutional Neural Network for Modelling Sentences[J]. Eprint Arxiv, 2014, 1.
9. Kim Y . Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.
10. Kingma D P , Ba J . Adam: A Method for Stochastic Optimization[J]. Computer ence, 2014.
11. Le Q V , Mikolov T . Distributed Representations of Sentences and Documents[J]. 2014.
12. Lin Z, Feng M, Santos C N, et al. A structured self-attentive sentence embedding[J]. arXiv preprint arXiv:1703.03130, 2017.
13. Liu P , Qiu X , Chen X , et al. Multi-Timescale Long Short-Term Memory Neural Network for Modelling Sentences and Documents[C]. Conference on Empirical Methods in Natural Language Processing. 2015.
14. Liu Y , Ji L , Huang R , et al. An Attention-Gated Convolutional Neural Network for Sentence Classification[J]. 2018.
15. Mikolov T . Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26:3111-3119.
16. Pennington J , Socher R , Manning C . Glove: Global Vectors for Word Representation[C]. Conference on Empirical Methods in Natural Language Processing. 2014.
17. Ren H, Lu H. Compositional coding capsule network with k-means routing for text classification[J]. 2018.

18. Ribeiro J G, Felisberto F S, Neto I C. Pruning and Sparsemax Methods for Hierarchical Attention Networks[J]. arXiv preprint arXiv:2004.04343, 2020.
19. Sabour S, Frosst N, Hinton G E. Dynamic routing between capsules[C]. Advances in neural information processing systems. 2017: 3856-3866.
20. Socher R, Perelygin A, Wu J, et al. Recursive deep models for semantic compositionality over a sentiment treebank[C]. Proceedings of the 2013 conference on empirical methods in natural language processing. 2013: 1631-1642.
21. Tang D , Qin B , Liu T . Learning Semantic Representations of Users and Products for Document Level Sentiment Classification[C]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015.
22. Tuan-Linh N , Swathi K , Minho L . A fuzzy convolutional neural network for text sentiment analysis[J]. Journal of Intelligent and Fuzzy Systems, 2018:1-10.
23. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in neural information processing systems. 2017: 5998-6008.
24. Wang D, Liu Q. An optimization view on dynamic routing between capsules[J]. 2018.
25. Wang R , Li Z , Cao J , et al. Convolutional Recurrent Neural Networks for Text Classification[C]. 2019 International Joint Conference on Neural Networks (IJCNN). 2019.
26. Wang Y , Huang M , Zhu X , et al. Attention-based LSTM for Aspect-level Sentiment Classification[C]. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.
27. Yang Z , Yang D , Dyer C , et al. Hierarchical Attention Networks for Document Classification[C]. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2017.
28. Zhao W , Ye J , Yang M , et al. Investigating Capsule Networks with Dynamic Routing for Text Classification[C]. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018.

