Dissect Sliced-RNN in Multi-Attention View

Zeping Yu[1 [0000−0002−3684−2512]], Linqing Shi[1 [0000−0003−0414−6691]], and Gongshen Liu✉[1 [0000−−0001−−5194−1570]]

Shanghai Jiao Tong University, Shanghai Minhang 200240, China
zepingyu@foxmail.com, {shilinqing,lgshen}@sjtu.edu.cn
https://www.sjtu.edu.cn/

Abstract. Sliced recurrent neural network (sliced-RNN) has achieved much faster speed and better performance than standard RNN. Training RNN used to be very slow, the slice structure successfully deals with this problem. The slice structure is a major breakthrough, however, the explanation of sliced-RNN is weak. It is difficult to understand why sliced-RNN can get performance improvement, because intuitively the slice structure destroys the relation of several adjacent words. To tackle this problem, we propose the multi-attention mechanism to explain why the slice structure could achieve good performance, and the visualization of the attention weights shows that the slice structure could obtain more information from the sequences. The slice structure is able to capture information on different levels. Additionally, we propose a shortcut connection over the multi-attention mechanism, which could help transfer important information effectively. We propose experiments on six text classification datasets, and the results show that the multi-attention mechanism and the shortcut connection are both useful.

Keywords: sliced-RNN · multi-attention · shortcut connection.

1 Introduction

Recurrent neural network (RNN) is useful on many NLP tasks. However, it is difficult to be parallelized because of the recurrent structure, so the speed of RNN is very slow. Sliced-RNN improves the structure by slicing the sequence into many subsequences and transmitting the information through multiple layers [26]. Because of the improved slice structure, sliced-RNN has much faster speed and higher accuracy on the document classification task. The model structure of sliced-RNN is shown in Figure 1.

Although the experimental results showed that sliced-RNN has better performance than RNN, the reason why the slice structure is good has not been explained convincingly. The speed advantage is easy to understand, but the performance improvement is not. Intuitively, the slice structure may destroy the connection between some adjacent words and phrases. A possible reason is that the slice structure could obtain high-level information rather than only word-level information through the multi-layer structure. But this explanation has not been verified. To explore this problem, we implement the multi-attention mechanism on each layer of sliced-RNN. The attention mechanism is a widely used function to focus on the most important part of the sequences because not all parts of a sequence are equally relevant [25]. The attention mechanism is widely used on a large amount of NLP tasks, such as machine translation [5], image caption [24], and text classification [25,28].

In this study, we propose the multi-attention mechanism, which could obtain information on different levels. We slice the sequence into many subsequences and use bidirectional RNN on the subsequences. And then we use the attention mechanism on each layer to capture the most important information of the subsequences. The representation of each subsequence is used as the input of the next layer, and this operation is repeated until we get the final representation of the sequence. In addition, we propose a shortcut connection over the multi-attention mechanism by paying attention to the input representations and the hidden states simultaneously on each layer. Moreover, we implement the multi-layer shortcut connection, which could be added across many layers. This method can help the information to be transmitted more efficiently.

The multi-attention mechanism could pay attention to the important information on different levels, and the information could be transmitted layer by layer. The model structure is similar to hierarchical attention network (HAN) [25]. Our model is more flexible than HAN, because the documents do not have to be split into sentences. Also, the attention mechanism could be added on more than 2 layers, so it could be much faster and could capture high-level information from the sequence.

We compare our model with attention-based bidirectional RNN (att-BRNN) [28]. Figure 2 is a review in Yelp 2013, and the task is predicting sentiment labels from 1-5. The key words in this sentence are great and like. Att-BRNN pays too much attention to the word great and predicts the wrong label 5. Our model captures both 2 words and predicts the correct label 4, and it also finds the "!" in the word 'vegus'. In this example, we could find the advantage of the multi-attention sliced-RNN. The slice structure could capture more comprehensive information from the sequence, so it could achieve better performance.

The main contribution of this paper is using the multi-attention mechanism to explain why the slice structure achieves better performance than the standard recurrent structure and propose a shortcut connection for multi-attention. We visualize the multi-attention weights and compare them with att-BRNN, and the results show that the slice structure could capture more information and transmit it through the layers. Our study clearly explains why the slice structure achieves good results.

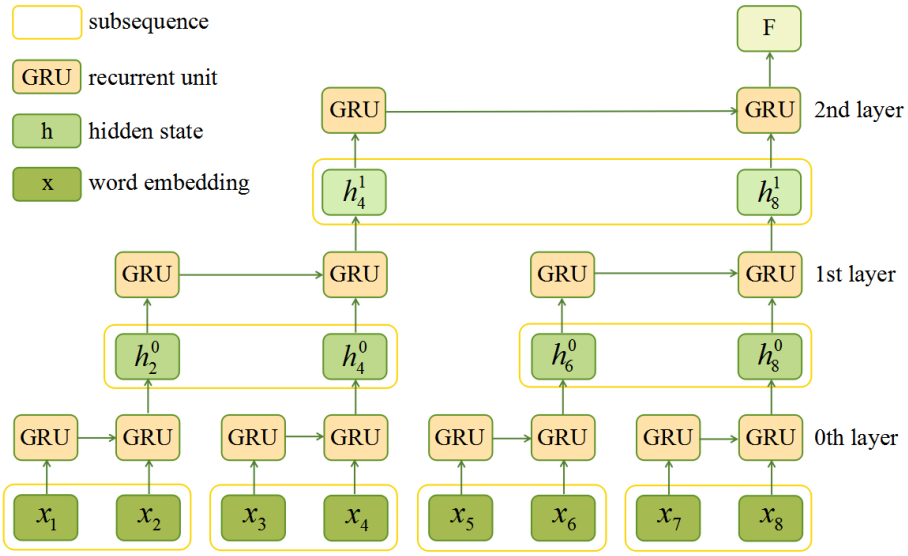


Fig. 1. The model structure of sliced-RNN. The recurrent units are GRU. Each subsequence is circled by a hollow rec- tangle.

2 Model Structure

In this section, we introduce the model structure of multi-attention sliced-RNN. As shown in Figure 3, our model has multiple layers. On the 0th layer, the input is the word embeddings of the sequence, and they are sliced into many subsequences. On each layer, a bidirectional RNN is added on the subsequences. Then the multi-attention mechanism pays attention to the hidden states of the bidirectional RNN, the important information is obtained in this way. Especially, if the shortcut connection is used, the multi-attention mechanism pays attention to both the bidirectional RNN and the input of this layer. Then the representation of each subsequence is used as the next layers input. This operation is repeated until the final representation F is obtained. Similar to sliced-RNN, the number of the layers and the length of each subsequence are hyper-parameters which need to be chosen. The details of the model will be described in this section.

2.1 Bidirectional RNN

Gated recurrent unit (GRU) [5] is a simplified version of long short-term memory (LSTM) [7]. Both of the two recurrent units have a gating mechanism, which could solve the vanishing gradient problem. GRU has a reset gate r and an update gate z, these gates could decide whether to keep the previous memory or accept the current input. The equations of GRU are as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{1}$$
$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{2}$$
$$c_t = tanh(W_c x_t + U_c(r_t \cdot h)) \tag{3}$$
$$h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot c_t \tag{4}$$

The reset gate controls the candidate hidden state, and the hidden state is controlled by the update gate. The standard GRU only processes the previous information and ignores the future information. Bidirectional GRU could get both the future context and the previous context. The input sequence is processed by two GRUs from both directions. The two hidden states from both directions are concentrated.

2.2 Multi-Attention

The attention mechanism could focus on the important parts of the sequences. Because the slice structure has many layers, the attention mechanism has different weights on different layers. In this way, the multi-attention mechanism could get multi-level

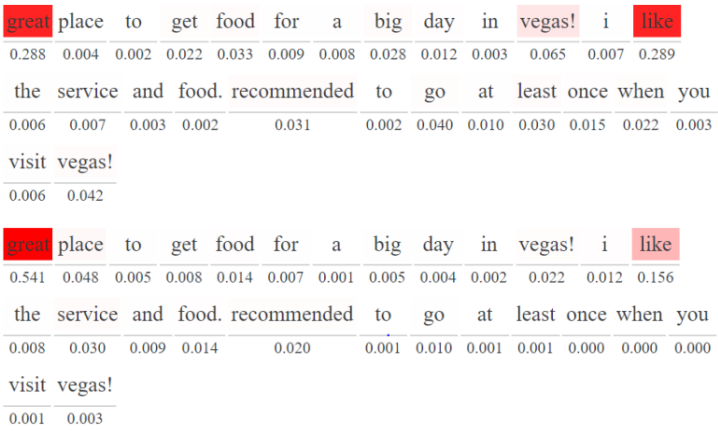


Fig. 2. An example from Yelp 2013 dataset. The word "great" and "like" has the largest contribution. The label of this document is 4 (1-5). The prediction of multi-attention sliced-RNN (above) is the correct label 4, and the prediction of attention-based bidirectional RNN (below) is 5.

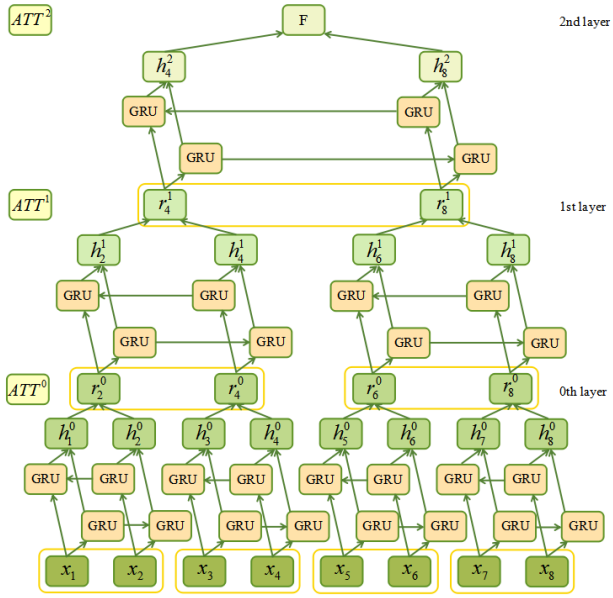


Fig. 3. Model structure of multi-attention sliced-RNN. The attention mechanism on multiple layers could capture more information on different levels. Here ATT means the attention mechanism, r is the representation of each subsequence.

information from the sequence. The important information could be transmitted through the layers. The attention mechanism on each layer is:

$$u = tanh(W_h h + b_h) \tag{5}$$

$$\alpha = softmax(v^T u) \tag{6}$$

$$s = \alpha^T h \tag{7}$$

where h is the hidden states produced by the bidirectional GRU. The representation s is the average of h. The attention mechanism could learn the weights.

shortcut connection The shortcut connection pays attention to both the input representations and the hidden states. If shortcut connection is adopted, the input of the multi-attention mechanism is the concatenation of the input representations and the

Table 1. Dataset information.

	Yelp 13	Yelp 14	Yelp 15	Yelp P	Amz.F	Amz.P
Documents	469k	670k	898k	598k	3.65M	4M
Average Lengths	129	116	108	153	93	91
Max Lengths	1060	1053	1092	1073	441	257
Classes Number	5	5	5	2	5	2

hidden states on each layer rather than the hidden states only. The important information could be transmitted easier through multiple layers. This idea is similar to the shortcut connection of Resnet [6], which is a commonly used method in CNN. With shortcut connection, this equation should be added before equation (5)-(7):

$$h = Concat(x, h) \tag{8}$$

where x is the input of this layer, and h is the hidden states of the bidirectional RNN.

Multi-layer Shortcut Connection The shortcut connection could help transmit important information directly across adjacent layers. When having many layers, an intuitive idea is to transmit information directly across many layers. Multi-layer shortcut connection could achieve this by paying attention to all the previous layers, which is similar to Densenet [8]. When using multi-layer shortcut connection, the mechanism should pay attention to:

$$h = Concat(x, x_p, h, h_p) \tag{9}$$

where x_p and h_p are the inputs and hidden states on previous layers.

2.3 Classification

In order to predict the labels, a softmax layer is added after the final representation F:

$$p = softmax(W_F F + b_F) \tag{10}$$

and the loss function is negative log-likelihood with document d:

$$L = -\sum \log p_d \tag{11}$$

3 Experiments

3.1 Datasets

Yelp reviews The Yelp reviews datasets are obtained from Yu and Liu [26]. They separate the dataset into Yelp 2013, Yelp 2014 and Yelp 2015. There are five sentiment labels from 1 to 5. Yelp P. is a polarity dataset with sentiment labels 1 and 2, which are proposed by Zhang et al. [27].

Amazon reviews The Amazon reviews datasets are also obtained from Zhang et al. [27]. It also has a polarity dataset. The document length of Amazon datasets is less than that of Yelp datasets.

3.2 Baselines

We compare our model with sliced-RNN, att-BRNN, HAN and standard RNN. We use GRU as the recurrent unit.

GRU The whole document is the input, and we take the last hidden state as the representation for classification.

HAN HAN was proposed by Yang et al. [25]. The hierarchical structure has word-level and sentence-level, which could be seen as a special slice structure. Intuitively, HAN could achieve better results because it has semantic features of the documents.

Attention-based bidirectional-RNN Attention-based bidirectional RNN was proposed by Zhou et al. [28]. They use the attention mechanism after the bidirectional RNN layer. We compare our method with this model because we want to explain why the slice structure could achieve better accuracy than the standard recurrent structure.

3.3 Model Training

In order to compare with the baseline models, the training method is similar to Yu and Liu [26]. We pad the sequences into a proper length. We pad zero on sequences shorter than that length, and sequences longer than that length are truncated. To compare with HAN, we set the number of the layers to be 2. The word vocabulary is set to be 30000. We use the Glove embeddings [17] to initialize the embeddings.

We tune the hyper parameters on the validation set. On all the datasets, the embedding dimension is 200 and the GRU dimension is 50, so the bidirectional GRU has 100 dimensions on each layer. The final representation also has 100 dimensions. The mini-batch size is set to be 100, and we use the Adam optimizer [10] having $\alpha = 0.001, \beta 1 = 0.9, \beta 2 = 0.999, and \epsilon = 10e(-8)$.

3.4 Experimental Results

The results are shown in Table 2. The first block is some deep CNN models with many layers. The second block contains some RNN-based models. The third block is our models with multi-attention and shortcut connection. Here MA means multi-attention, SC means shortcut connection, MLSC means multi-layer shortcut connection.

When we look at the second block and the third block, models with the attention mechanism achieve better performance than the non-attention models. Take Yelp 2013 dataset as an example. SRNN+MA has 1.46% higher accuracy than SRNN. Att-BRNN has 1.8% higher accuracy than GRU. So we could see that the attention mechanism is very useful, it could pay attention to the most important parts of the sequences and extract the important information. When exploring the slice structure, we could find that the slice structure achieves higher accuracy than the standard recurrent structure. The accuracy improvement is not easy to understand. Intuitively, the connection between the words may be destroyed by the slice structure. However, the experimental results have shown that the slice structure achieves very good performance. An explanation is that the slice structure could get high-level information from the sequence, instead of only extracting the word-level information. In order to validate whether this explanation is credible, we will visualize the attention weights of multi-attention without shortcut connection and explore what the slice structure could learn. When we explore the shortcut connection and multi-layer shortcut connection, we could find that they are very useful and greatly improve the performance. On all the datasets sliced-RNN with one-layer or multi-layer shortcut connection outperforms the other models. The shortcut connection over the multi-attention mechanism is helpful for the information to be transmitted through multiple layers. The shortcut connection could choose to extract information from the hidden states or just the input representations on each layer, so the model is stronger and more flexible. On all datasets, HAN has a little higher accuracy. than sliced-RNN. This is because of the semantic knowledge of the documents. The slice structure is more flexible because it does not need to be split into sentences. Moreover, the hierarchical structure of HAN sometimes does not work because the input sequence does not have any semantic features. For example, when we use RNN on user modeling and recommender system, the input sequences are the web pages that the users clicked before. The sequences cannot be split into sentences and words, while the slice structure is still useful in this case.

3.5 Visualization and Analysis

In order to verify whether our model could extract the important information from the input sequence, we visualize the attention weights and compare it with att-BRNN. We use multi-attention without short connection in order to explore the slice structure. We choose several examples from the Yelp 2013 dataset. Because the model has multiple levels on different layers, we multiply the attention weights together to compare with att-BRNN, which only has the word-level attention. The results are shown in Figure 4, Figure 5 and Figure 6. The larger the weight is, the darker the color is shown. The weight of each word is printed under the words.

The visualization of multi-attention is shown on the left, and that of att-BRNN is shown on the right. In order to compare with att-BRNN, we only show the word-level weights instead of showing the high-level weights. But when we look at this visualization, we could find that even if one word has large weight, the words near it have very small weights. It means that the slice structure has the ability to transmit important information through multiple layers. The weights of the multi-level

Table 2. Accuracy on document classification datasets, in percentage.

	Yelp 13	Yelp 14	Yelp 15	Yelp P	Amz.F	Amz.P
Char CNN(9 layer)[27]	-	-	-	94.46	57.41	94.15
WOrd CNN(9 layer)[27]	-	-	-	95.40	55.60	94.16
Char CNN(9 layer)[27]	-	-	-	95.72	63.00	95.59
GRU	66.12	70.63	72.89	95.96	61.36	95.22
SRNN[26]	66.80	70.76	73.30	95.99	61.41	95.18
HAN[25]	68.34	71.46	74.03	96.43	63.49	95.98
Att-BRNN[28]	67.92	71.14	73.62	96.17	62.73	95.46
SRNN+MA	68.26	71.32	73.84	96.26	63.24	95.53
SRNN+MA+SC	68.44	71.82	74.19	96.46	63.58	96.16
SRNN+MA+MLSC	68.56	71.64	74.47	96.48	63.44	96.28

attention could validate this. Take Figure 2 as an example. It is a 3-layer sliced-RNN model with multi-attention. The input length is 512, the sequence length of Figure 2 is 27, and the length of the subsequences on each layer is 8. The words great and like have the largest weight. The word great is the 6th word of the 5th slice on the 2nd layer, and like is the 2nd word of the 7th slice on the 2nd layer. We extract the attention weights on the 2nd layer: [0.00, 0.00, 0.00, 0.00, 0.30, 0.18, 0.35, 0.17], which means that the 5th and the 7th slice are the most 2 important slices. And the attention weights of the 5th slice are: [0.00, 0.00, 0.00, 0.00, 0.00, 0.97, 0.02, 0.01], and the 7th slice: [0.02, 0.83, 0.02, 0.01, 0.01, 0.01, 0.09, 0.01]. These two large weights correspond to the words great and like. This means our model could indeed find the most important information from the bottom layer and pass them layer by layer.

Figure 4 is an example that both multi-attention sliced-RNN and att-BRNN predict the correct label. We could find that our model finds more information than att-BRNN. Our model not only finds the words good and fair but also captures the words properly, little rich and still, which are also important. Our model could capture more information because it has more layers. The receptive field on each layer is small, so it is easier to capture the information and transmit it layer by layer. In contrast, the receptive field of att-BRNN is too large, it is hard to capture all the important information, so it just extracts some information and sets the weights very large. We could look at the weights on Figure 4, Figure 5 and Figure 6. In our model, the weights are smooth and distributed, the largest weight is less than 0.15. But in att-BRNN, the weights are large on very few words, such as good and not. The largest weight is more than 0.2. Once their model captures the wrong word, it will predict the wrong model, just like the mistake made in Figure 2.

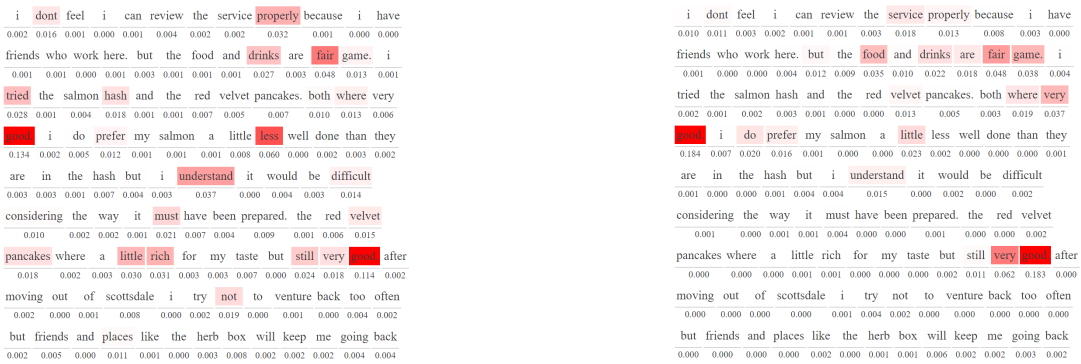
Figure 5 shows an example with label 1. Our model predicts the correct label, and att-BRNN predicts label 2. We could find an amazing result from the visualization: Even if the word love has the largest weight, our model still predicts the correct label. This means our model definitely understands the word love is not to praise, but to criticize. Although att-BRNN captured the word not and set the weight very large, it regards the word love as a positive word, so att-BRNN thinks the review is an implied criticism and predicts the wrong label 2.

Figure 6 is an example with label 1. Att-BRNN predicts the correct label, and our model predicts label 2. However, when we look at the review, we may have the same choice as multi-attention sliced-RNN. Our model captures the words on a good note and great, which talk something positive, that is, the onion rings. Our model recognizes them, while Att-BRNN does not. Amazingly, although the words on a good note and rings where great are actually in two subsequences, the slice structure captures this information. This means the slice structure does not break the word connection and could transfer the information through layers.

In addition to the examples already shown, there are many other examples that demonstrate the validity of the slice structure. We will open source our data and code later.

4 Related Work

Deep learning methods have great results on text classification. Kim uses convolutional neural network (CNN) for text classification [9]. Zhang et al. build a character-level CNN for text classification [27]. RNN is the most widely used model for sequences because it could capture the order information, so it is used on many NLP tasks, including machine translation [5], image caption [24], and question answering [4,23]. Several works combine CNN and RNN for text classification [12,**?**].



(a) pic1. (b) pic2.

Fig. 4. An example with label 4 from Yelp 2013. Both multi-attention sliced-RNN (left) and att-BRNN (right) predict the correct label.



(a) pic1. (b) pic2.

Fig. 5. An example with label 1 from Yelp 2013. Multi-attention sliced-RNN (left) predicts the correct label, att-BRNN (right) predict label 2.



(a) pic1. (b) pic2.

Fig. 6. An example with label 1 from Yelp 2013. Multi-attention sliced-RNN (left) predicts label 2, att-BRNN (right) predict the correcct label.

The attention mechanism is used in many NLP tasks including machine translation [5], image caption [24], and text classification [28]. Another attention mechanism called self-attention could capture the relationship of different positions in a sequence [22], and it has been used successfully in a variety of tasks including semantic role labeling [19], abstractive summarization [16], textual entailment [15] and learning task-independent sentence representations [14].

Several works try to improve the RNN speed by improving the recurrent units [1,2,11,13]. Some architectures are proposed to improve the recurrent structure of RNN [3,11,18]. The slice structure proposed by Yu and Liu 2018 [26] could greatly improve the speed of RNN. The structure is similar to wavenet [21], and the difference is that sliced-RNN uses recurrent units. The slice structure is also related to the hierarchical structure [20,25]. The slice structure is more flexible because it does not need to split the documents into many sentences, and it is also much faster because it could have more than 2 layers and shorter subsequence length.

5 Conclusions

In this paper, we present the multi-attention mechanism for sliced-RNN, which explains why sliced-RNN has better performance than RNN. Also, we propose the shortcut connection and multi-layer shortcut connection, which could transmit the information effectively and improve the performance. By visualizing the attention weights and comparing with attention-based bi-RNN, we could find that the slice structure could capture more information, and the information could be transmitted

layer by layer. Experimental results on six document classification datasets have shown that sliced-RNN with multi-attention mechanism achieves the best performance.

6 Acknowledgements

This research work has been funded by the National Natural Science Foundation of China (Grant No.61772337, U1736207), and the National Key Research and Development Program of China NO. 2016QY03D0604 and 2018YFC0830703.

References

1. Balduzzi, D., Ghifary, M.: Strongly-typed recurrent neural networks. arXiv preprint arXiv:1602.02218 (2016)
2. Bradbury, J., Merity, S., Xiong, C., Socher, R.: Quasi-recurrent neural networks. arXiv preprint arXiv:1611.01576 (2016)
3. Chang, S., Zhang, Y., Han, W., Yu, M., Guo, X., Tan, W., Cui, X., Witbrock, M., Hasegawa-Johnson, M.A., Huang, T.S.: Dilated recurrent neural networks. In: Advances in Neural Information Processing Systems. pp. 77–87 (2017)
4. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading wikipedia to answer open-domain questions. arXiv preprint arXiv:1704.00051 (2017)
5. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
8. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
9. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Koutnik, J., Greff, K., Gomez, F., Schmidhuber, J.: A clockwork rnn. arXiv preprint arXiv:1402.3511 (2014)
12. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Twenty-ninth AAAI conference on artificial intelligence (2015)
13. Lei, T., Zhang, Y., Artzi, Y.: Training rnns as fast as cnns. arXiv preprint arXiv:1709.02755 (2017)
14. Lin, Z., Feng, M., Santos, C.N.d., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130 (2017)
15. Parikh, A.P., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933 (2016)
16. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304 (2017)
17. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
18. Sutskever, I., Hinton, G.: Temporal-kernel recurrent neural networks. Neural Networks **23**(2), 239–243 (2010)
19. Tan, Z., Wang, M., Xie, J., Chen, Y., Shi, X.: Deep semantic role labeling with self-attention. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
20. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 conference on empirical methods in natural language processing. pp. 1422–1432 (2015)
21. Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. SSW **125** (2016)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
23. Xiong, C., Merity, S., Socher, R.: Dynamic memory networks for visual and textual question answering. In: International conference on machine learning. pp. 2397–2406 (2016)
24. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)
25. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1480–1489 (2016)
26. Yu, Z., Liu, G.: Sliced recurrent neural networks. arXiv preprint arXiv:1807.02291 (2018)
27. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in neural information processing systems. pp. 649–657 (2015)
28. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). vol. 2, pp. 207–212 (2016)