AN ATTENTION-AWARE BIDIRECTIONAL MULTI-RESIDUAL RECURRENT NEURAL NETWORK (ABMRNN): A STUDY ABOUT BETTER SHORT-TERM TEXT CLASSIFICATION

Ye Wang[1], *Han Wang*[1], *Xinxiang Zhang*[2], *Theodora Chaspari*[1], *Yoonsuck Choe*[1] *and Mi Lu*[1]

[1]Texas A&M University, College Station, Texas, 77843, USA
[2]Southern Methodist University, Dallas, Texas, 75025, USA
{wangye0523, hanwang, chaspari, choe}@tamu.edu, xinxiang@smu.edu, mlu@ece.tamu.edu

ABSTRACT

Long Short-Term Memory (LSTM) has been proven an efficient way to model sequential data, because of its ability to overcome the gradient diminishing problem during training. However, due to the limited memory capacity in LSTM cells, LSTM is weak in capturing long-time dependency in sequential data. To address this challenge, we propose an Attention-aware Bidirectional Multi-residual Recurrent Neural Network (ABMRNN) to overcome the deficiency. Our model considers both past and future information at every time step with omniscient attention based on LSTM. In addition to that, the multi-residual mechanism has been leveraged in our model which aims to model the relationship between current time step with further distant time steps instead of a just previous time step. The results of experiments show that our model achieves state-of-the-art performance in classification tasks.

Index Terms— Long Short-Term Memory, recurrent neural network, attention model, natural language processing, residual network

1. INTRODUCTION

Compared with Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) are widely applied to sequential data such as natural language processing [1] and speech processing [2], while CNNs are more employed in image processing fields [3–5]. Among the existing RNN models, LSTM is one of the most widely approaches since it initially solved gradient vanishing and exploding problems during RNN training [6] by introducing forget gate and memory cell. Numerous RNNs variations [6–8] have been proposed in previous literature to achieve the state-of-the-art performance in different tasks, where LSTM is the cornerstone of those structures. With the increase in the depth of the layers, residual networks have proved their advantages in both CNNs [9] and RNNs [10]. Residual networks provide an alternative to LSTMs by connecting current and distant time steps during training.

In this paper, we propose an Attention-aware Bidirectional Multi-residual Recurrent Neural Network (ARMRNN) and have shown improved performance in existing sequential classification tasks. To summarize our contributions:

- We propose a algorithm which enables the updating of the weights combining both previous and future time steps.
- We leverage a multi-residual mechanism from existing residual network into the recurrent networks for sequence learning, through which we achieve the state-of-the-art performance in classification tasks.
- We provide comprehensive analysis of the advantages and disadvantages of current cutting-edge models including RNNs and CNNs for sequence learning, especially in short-term text classification tasks.

2. RELATED WORK

Regarding improving the performance of classification tasks, there are some directions towards networks exploration. First, an increasing number of layers is employed for capturing features. Second, various feature extraction methods such as word2vec [11] and doc2vec [12] have been invented for better words representations learning. Third, some variations towards the interior structure units such as LSTM and GRU [7] are proposed. With the development of neural networks, a novel trend is to combine deeper networks and multiple neural network variations.

Since general CNNs or RNNs architectures do not fit well in some tasks such as short-term text classification, the contribution of this work lies in the fact that, it integrates advantages of residual networks for the tasks of interest.

3. RESIDUAL LSTM PRELIMINARIES

LSTM solves gradient vanishing and exploding problems. However, if the time sequence is too long, the dependency between the former and latter information is neglected in LSTM because current time step only depends on previous time step. To enhance such a distant relationship, residual network based on LSTM has been proposed [10, 13]. Figure 1 shows the general structure of a residual network. The basic

978-1-5386-4658-8/18/$31.00 ©2019 IEEE

idea of the recurrent residual network is to add a direct line between different time steps to strengthen the connection.

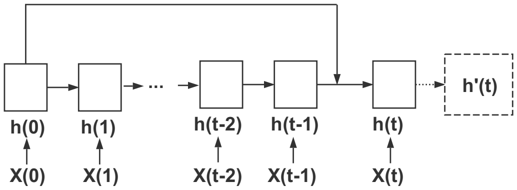

Fig. 1. Residual Network. The dashed box means the updated state in current time step.

Regarding the implementation of the residual network, for each current time step t, we consider both of the previous time step $t-1$ and the additional specific previous time step (e.g. we assume it is $t(0)$ as Figure 1 shows).

$$C_{t-1}^{new} = C_{t-1}^{old} + \alpha * C_0 \tag{1}$$

$$h_{t-1}^{new} = h_{t-1}^{old} + \alpha * h_0 \tag{2}$$

where α represents the specific weight of how much information is imported to current time step, C and h represent cell states and hidden states respectively. RNN residual networks [10] leverage CNN residual network [9] and indeed improve the performance of LSTM.

4. PROPOSED SCHEME

4.1. Attention Model

We leverage the attention model to enhance previous system states correlation with the current state [14–16]. We define each weighted summation (WS) as the whole attention within the current time step and we illustrate the equations as below:

$$WS = \Sigma_{T=t1}^{tn}(a_T \times h_T) \tag{3}$$

$$a_T = \frac{exp(W \cdot h_T)}{\Sigma_{T=t1}^{tn} exp(W \cdot h_T)} \tag{4}$$

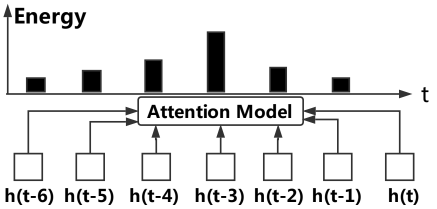

Fig. 2. Attention Model

In Equation 3, h_T represents the hidden state value in LSTM at time step T. a_T is a scalar value representing the weight at time step T. We compute a_T by softmax form and W is a parameter which needs to be learned. $exp(W \times h_T)$ represents the potential energy at time step T. Ideally, for every time step T, we look back the whole previous states to check the relationship with each others. However, due to the limitation of computational power, we select a few past states as a sliding window. Figure 2 shows one example of the attention model. The highest attention we acquired is in h_{t-3} regarding to the current time step t.

4.2. Multi-residual LSTM

[9] initially proposed a promising residual learning framework for deep learning network. They attempted to build a block as:

$$y = \mathcal{F}(x, \{W_i\}) + x \tag{5}$$

where x and y are input and output vectors. The function $\mathcal{F}(x, \{W_i\})$ represents the residual mapping to be learned. The operation $\mathcal{F} + x$ is performed by a shortcut connection and element-wise addition. The residual network is initially proposed in CNNs. However, we leverage the residual networks into RNNs as Figure 3 illustrates, finding the promising results. We get inspired by residual networks because of the limitation of traditional LSTM, where the original LSTM only considers the output from the previous time step as an input of current time step. By combing residual and LSTM, we connect any two distant states. Besides, we add an attention model to explore the relationships among the whole past states.

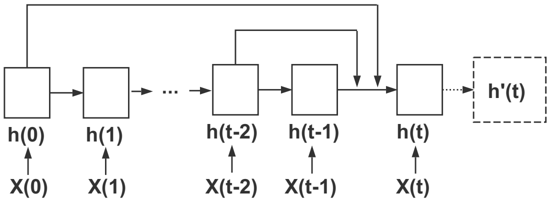

Fig. 3. Multi-residual LSTM with attention Model

We illustrate our idea as Figure 3 shows. We connect $h(t-2)$ and $h(0)$ with current time step $h(t)$ since $h(t-2)$ and $h(0)$ gained more attentions compared with other states.

4.3. Attention-aware Bidirectional Multi-residual LSTM

We initially propose an algorithm, which updates the weights by combining both previous and future time steps. The equations are shown as below:

$$h'(t) = [\overrightarrow{h}(t), \overleftarrow{h}(t)] \tag{6}$$

$$\overrightarrow{h'}(t-1) = \overrightarrow{h}(t-1) + \Sigma_{\overrightarrow{T}}\overrightarrow{a_T}(\tanh(C_T) \otimes \sigma(x_T)) \quad (7)$$

$$\overleftarrow{h'}(t-1) = \overleftarrow{h}(t-1) + \Sigma_{\overleftarrow{T}}\overleftarrow{a_T}(\tanh(C_T) \otimes \sigma(x_T)) \quad (8)$$

where $\overrightarrow{T}, \overleftarrow{T} \in \mathbb{N}, t-n \leq \overrightarrow{T} \leq t-1, t+1 \leq \overleftarrow{T} \leq t+n$.

In Equation 7 and Equation 8 $\overrightarrow{h}(t-1)$ and $\overleftarrow{h}(t-1)$ are the original forward and backward hidden states at time step $t-1$ of bidirectional LSTM. $\overrightarrow{h'}(t-1)$ and $\overleftarrow{h'}(t-1)$ are updated forward and backward hidden states at time step $t-1$ of the proposed ABMRNN. The residual we introduced here, is the weighted summation of the hidden states from selected time steps T based on attention scalar a_T in Equation 4. The hidden states at time step $t-1$ are the input to compute the output at time step t.

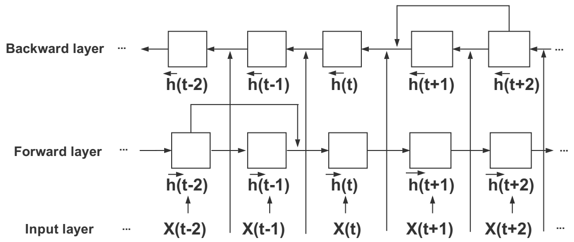

Fig. 4. Bidirectional multi-residual LSTM with attention model

Compared with previous models, we add one more layer as Figure 4 shows. Both forward and backward training sequences are taken into considerations. The advantage is we consider current time step together with both before and after time steps information. Our model allows more time flexibilities in terms of recalling distant past time steps, predicting the future pieces of information and evaluating the influence of each state.

4.4. Training procedure

ABMRNN training procedure is given as pseudo-code in Algorithm 1. The procedure takes bi-directional input sequence x_{bi}, which is composed of forward and reversed order sequences. The objective is to minimize the loss function by updating hidden states and the corresponding attention dynamics. $\mathcal{F}$ is denoted as the function of ABMRNN to obtain output states. $\mathcal{A}$ is defined as the function of updated attention and $\mathcal{H}$ is defined as the function of updated hidden states. W is defined as matrix weights while W_{tmp} is temporary updated weights after attention model. h is defined as initial hidden states while h^* is defined as updated hidden states. y is defined as initial output while y^* is defined as updated output.

Algorithm 1 ABMRNN training procedure

```
$x_{bi} \hookleftarrow \{\overrightarrow{x}, \overleftarrow{x}\}$ where $\overrightarrow{x}$ is the forward input, $t$ is the target value and $\overleftarrow{x}$ is the backward (reversed) input.
$\epsilon$ number of epochs
$e \hookleftarrow 0$
for $e < \epsilon$ do
    for $x_T \in x_{bi}$ do
        $y \hookleftarrow \mathcal{F}(x_{bi}, \{W\})$
        $W_{tmp} \hookleftarrow \mathcal{A}(\{W\}), h^* \hookleftarrow \mathcal{H}(h, \{W_{tmp}\})$
        $y^* \hookleftarrow \mathcal{F}(y, h^*, \{W_{tmp}\})$
        error $E \hookleftarrow \|t - y^*\|$
        update $W \hookleftarrow backpropagate(W, E)$
    end for
    $e \hookleftarrow e + 1$
end for
```

5. EXPERIMENTS AND RESULTS

5.1. Task introduction

We evaluated our model and other existing architectures on a challenging short-term text classification task (STCT). Unlike traditional long text documents, short-term text such as headings, news titles are usually concise, which somehow hinder the classification performance due to the short information. [17] introduces STCT which are usually construct by 20 Chinese words in coarse and refined categories. There are eight labels in coarse categories and 59 labels in refined categories and the total number of text is 400,000. Besides, STCT provides baseline performance of traditional statistical methods including support vector machine, decision tree and logistic regression.

AG_NEWS is a collection news articles labelled in four categories. We randomly select 8,000 samples for training and 1000 samples for testing and the average length of each title is 30.

IMDB movie review dataset is a binary sentiment classification task which contains movie reviews with positive and negative labels. The maximum length in IMDB review is up to 3000 and the average length is about 300. There are 50000 samples selected, and we use half for training and another half for testing.

MNIST is an image classification task (10 categories). We regard the image pixels as the sequential data and the flatted images of MNIST are fed into the networks to predict the image label. Therefore, sequential MNIST is assumed as a solid task for long time dependencies modelling (up to 784). There are 60000 training samples and 10000 testing samples.

After selecting the feature, for better illustrating the improvement, various RNN models are evaluated and compared such as plain RNNs, LSTMs, Bidirectional LSTM, single residual and multi-residual networks. Besides, we also utilize 1-D CNNs into those given sequential tasks to compare the performance with RNNs.

Model	IMDB	AG_NEWS	Seq. MNIST	STCT
Plain LSTM	88.77%	82.33%	97.01%	93.01%
Bi-LSTM	89.91%	83.13%	98.31%	94.10%
2-layer LSTM	88.42%	82.27%	98.03%	93.16%
1-layer IndRNN	80.60%	84.98%	97.58%	93.02%
5-layer IndRNN	76.39%	84.74%	97.71%	88.89%
Plain RNN	77.12%	80.33%	97.66%	78.89%
5-layer RNN	50.00%	77.76%	97.45%	87.23%
1-D CNN	88.70%	84.61%	98.01%	94.50%
Attention-LSTM	89.50%	82.17%	98.31%	95.88%
Residual-LSTM	90.80%	84.71%	98.03%	93.55%
Proposed model	**90.91%**	**86.31%**	**98.53%**	**96.50%**

Table 1. Accuracy in classification Results

Our model applies two layers with 128 forward and 128 backward LSTM units. For better optimization, we utilized [18] with gradient clipping. All the weights are randomly initialized by the isotropic Gaussian distribution of variance 0.1. The dropout rate is 0.2 for each layer [19] and the batch size is 64. Regarding the other models, we keep the consistent settings, which have 128 hidden units in hidden layers. The kernel size of 1-D CNNs is three.

5.2. Analysis

Results of the experiment are shown in Table 1. Our model achieves the state-of-the-art performance in STCT. In STCT, the highest accuracy rate is 96.50%, where the baseline performance provided by [17] is 69.03%. Therefore we advance the ground truth about 39.7%. Even the plain RNN model outperforms the statistical classification models (SVM 69.03% vs Plain RNN 78.89%). With the model becoming more advanced, the performance increasingly improved (Plain RNN 78.89% - LSTM 93.01% - Bi-LSTM 94.10%). We also attempt other architectures in STCT such as IndRNN [8] and multi-layer RNNs. Our model outperforms all the existing methods.

We are also concerned about the training loss and we select four models because they represent typical structures. We only show the result of STCT because the training loss in other tasks is similar with STCT. In Figure 5, the training loss in plain RNNs keeps oscillating, which means it is hard to converge. Both LSTM and IndRNN converge after only a few epochs, however, LSTM converges slower than IndRNN. Although the training loss of ABMRNN is a little higher than that of LSTM, training loss can only guarantee marginally lower bound but not upper bound regarding the accuracy rate.

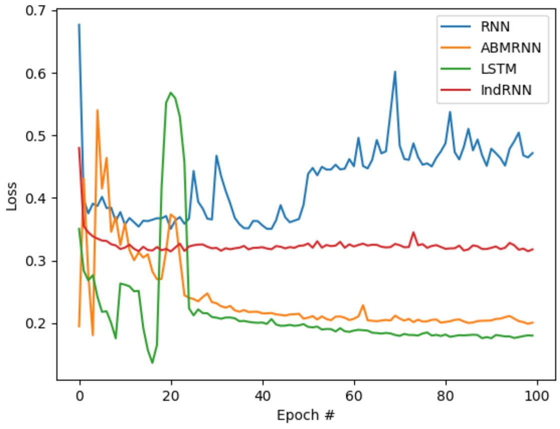

Fig. 5. STCT Training loss

Our model still outperforms the other RNN-based models in IMDB, MNIST and AG's news corpus. In IMDB datasets, with the text length increasing up to 3000, the performance is impacted due to more redundancies and noises are introduced. However, some recent algorithms leverage very deep CNN-related frameworks as feature extraction. The numbers of parameters of [20] (7.8M) [21] (11.3M and 50M) are at least ten times more than the number of our parameters (0.5M). Our model demonstrates high efficiency in training and comparatively top accuracy. In sequtial MNIST, we don't compare with current popular 2-D CNNs because those models treat MNIST as images instead of sequence.

6. CONCLUSION AND FUTURE WORK

We initially proposed a model towards short-term text classification, leveraging residual network and attention module. The result has shown that our ABMRNN model outperforms other conventional RNN models such as LSTM, IndRNN and their variations, and achieves the state-of-the-art performance in STCT. Compared with existing RNNs models, our model is applied more to the short-term text classification because each current time step considers both past and future distant time steps to correct the relationship more precisely. Our future work includes utilizing our ABMRNN in other tasks and further optimizing our model.

7. REFERENCES

[1] Han Wang, Ye Wang, Mi Lu, and Yoonsuck Choe, "English out-of-vocabulary lexical evaluation task," *arXiv preprint arXiv:1804.04242*, 2018.

[2] Fei Tao and Carlos Busso, "Gating neural network for large vocabulary audiovisual speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 7, pp. 1286–1298, 2018.

[3] Yue Zhang and Xinxiang Zhang, "Effective real-scenario video copy detection," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016, pp. 3951–3956.

[4] Ye Wang, Yuanjiang Huang, Wu Zheng, Zhi Zhou, Debin Liu, and Mi Lu, "Combining convolutional neural network and self-adaptive algorithm to defeat synthetic multi-digit text-based CAPTCHA," in *Industrial Technology (ICIT), 2017 IEEE International Conference on*. IEEE, 2017, pp. 980–985.

[5] Ye Wang and Mi Lu, "An optimized system to solve text-based CAPTCHA," *arXiv preprint arXiv:1806.07202*, 2018.

[6] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[8] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao, "Independently recurrent neural network (INDRNN): Building a longer and deeper RNN," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5457–5466.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] Yiren Wang and Fei Tian, "Recurrent residual learning for sequence classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 938–943.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[12] Quoc Le and Tomas Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, 2014, pp. 1188–1196.

[13] Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass, "Highway long short-term memory rnns for distant speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5755–5759.

[14] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[15] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4945–4949.

[16] Fei Tao and Gang Liu, "Advanced LSTM: A study about better time dependency modeling in emotion recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2906–2910.

[17] Ye Wang, Zhi Zhou, Shan Jin, Debin Liu, and Mi Lu, "Comparisons and selections of features and classifiers for short text classification," in *Materials Science and Engineering Conference Series*, 2017, vol. 261, p. 012018.

[18] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, p. 14, 2012.

[19] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[20] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun, "Very deep convolutional networks for text classification," *arXiv preprint arXiv:1606.01781*, 2016.

[21] Xiang Zhang, Junbo Zhao, and Yann LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.