Sentence Modeling with Gated Recursive Neural Network

Xinchi Chen, Xipeng Qiu*, Chenxi Zhu, Shiyu Wu, Xuanjing Huang
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, China
{xinchichen13,xpqiu,czhu13,syu13,xjhuang}@fudan.edu.cn

Abstract

Recently, neural network based sentence modeling methods have achieved great progress. Among these methods, the recursive neural networks (RecNNs) can effectively model the combination of the words in sentence. However, RecNNs need a given external topological structure, like syntactic tree. In this paper, we propose a gated recursive neural network (GRNN) to model sentences, which employs a full binary tree (FBT) structure to control the combinations in recursive structure. By introducing two kinds of gates, our model can better model the complicated combinations of features. Experiments on three text classification datasets show the effectiveness of our model.

1 Introduction

Recently, neural network based sentence modeling approaches have been increasingly focused on for their ability to minimize the efforts in feature engineering, such as Neural Bag-of-Words (NBoW), Recurrent Neural Network (RNN) (Mikolov et al., 2010), Recursive Neural Network (RecNN) (Pollack, 1990; Socher et al., 2013b; Socher et al., 2012) and Convolutional Neural Network (CNN) (Kalchbrenner et al., 2014; Hu et al., 2014).

Among these methods, recursive neural networks (RecNNs) have shown their excellent abilities to model the word combinations in sentence. However, RecNNs require a pre-defined topological structure, like parse tree, to encode sentence, which limits the scope of its application. Cho et al. (2014) proposed the gated recursive convolutional neural network (grConv) by utilizing the directed acyclic graph (DAG) structure instead of parse tree to model sentences. However, DAG structure is relatively complicated. The number of the hidden neurons quadraticly increases with the length of sentences so that grConv cannot effectively deal with long sentences.

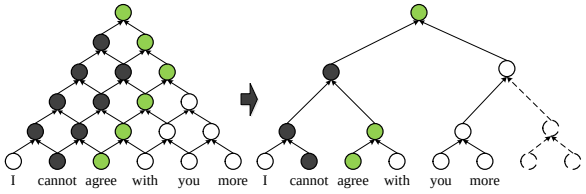

Figure 1: Example of Gated Recursive Neural Networks (GRNNs). Left is a GRNN using a directed acyclic graph (DAG) structure. Right is a GRNN using a full binary tree (FBT) structure. (The green nodes, gray nodes and white nodes illustrate the positive, negative and neutral sentiments respectively.)

Inspired by grConv, we propose a gated recursive neural network (GRNN) for sentence modeling. Different with grConv, we use the full binary tree (FBT) as the topological structure to recursively model the word combinations, as shown in Figure 1. The number of the hidden neurons linearly increases with the length of sentences. Another difference is that we introduce two kinds of gates, reset and update gates (Chung et al., 2014), to control the combinations in recursive structure. With these two gating mechanisms, our model can better model the complicated combinations of features and capture the long dependency interactions.

In our previous works, we have investigated several different topological structures (tree and directed acyclic graph) to recursively model the semantic composition from the bottom layer to the top layer, and applied them on Chinese word segmentation (Chen et al., 2015a) and dependency parsing (Chen et al., 2015b) tasks. However, these structures are not suitable for modeling sentences.

*Corresponding author.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 793–798,
Lisbon, Portugal, 17-21 September 2015.

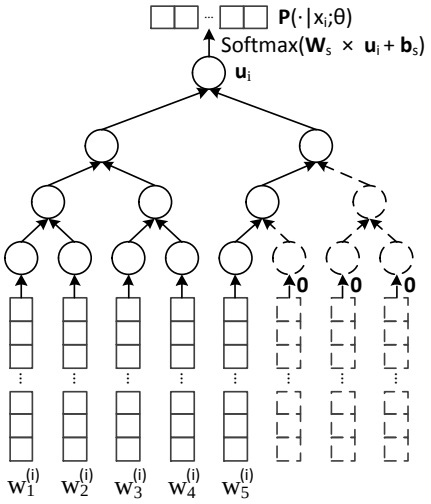

Figure 2: Architecture of Gated Recursive Neural Network (GRNN).

In this paper, we adopt the full binary tree as the topological structure to reduce the model complexity.

Experiments on the Stanford Sentiment Treebank dataset (Socher et al., 2013b) and the TREC questions dataset (Li and Roth, 2002) show the effectiveness of our approach.

2 Gated Recursive Neural Network

2.1 Architecture

The recursive neural network (RecNN) need a topological structure to model a sentence, such as a syntactic tree. In this paper, we use a full binary tree (FBT), as showing in Figure 2, to model the combinations of features for a given sentence.

In fact, the FBT structure can model the combinations of features by continuously mixing the information from the bottom layer to the top layer. Each neuron can be regarded as a complicated feature composition of its governed sub-sentence. When the children nodes combine into their parent node, the combination information of two children nodes is also merged and preserved by their parent node. As shown in Figure 2, we put all-zero padding vectors after the last word of the sentence until the length of $2^{\lceil log_2^n \rceil}$, where n is the length of the given sentence.

Inspired by the success of the gate mechanism of Chung et al. (2014), we further propose a gated recursive neural network (GRNN) by introducing two kinds of gates, namely "reset gate" and "update gate". Specifically, there are two reset gates, $\mathbf{r}_L$ and $\mathbf{r}_R$, partially reading the information from

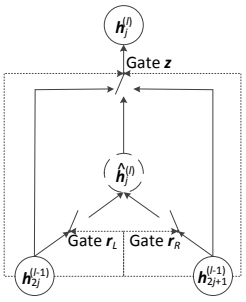

Figure 3: Our proposed gated recursive unit.

left child and right child respectively. And the update gates $\mathbf{z}_N$, $\mathbf{z}_L$ and $\mathbf{z}_R$ decide what to preserve when combining the children's information. Intuitively, these gates seem to decide how to update and exploit the combination information.

In the case of text classification, for each given sentence $x_i = w^{(i)}_{1:N(i)}$ and the corresponding class y_i, we first represent each word $w^{(i)}_j$ into its corresponding embedding $\mathbf{w}_{w^{(i)}_j} \in \mathbb{R}^d$, where $N(i)$ indicates the length of i-th sentence and d is dimensionality of word embeddings. Then, the embeddings are sent to the first layer of GRNN as inputs, whose outputs are recursively applied to upper layers until it outputs a single fixed-length vector. Next, we receive the class distribution $\mathrm{P}(\cdot|x_i;\theta)$ for the given sentence x_i by a softmax transformation of $\mathbf{u}_i$, where $\mathbf{u}_i$ is the top node of the network (a fixed length vectorial representation):

$$\mathrm{P}(\cdot|x_i;\theta) = \mathrm{softmax}(\mathbf{W}_s \times \mathbf{u}_i + \mathbf{b}_s), \qquad (1)$$

where $\mathbf{b}_s \in \mathbb{R}^{|T|}$, $\mathbf{W}_s \in \mathbb{R}^{|T| \times d}$. d is the dimensionality of the top node $\mathbf{u}_i$, which is same with the word embedding size and T represents the set of possible classes. θ represents the parameter set.

2.2 Gated Recursive Unit

GRNN consists of the minimal structures, gated recursive units, as showing in Figure 3.

By assuming that the length of sentence is n, we will have recursion layer $l \in [1, \lceil log_2^n \rceil + 1]$, where symbol $\lceil q \rceil$ indicates the minimal integer $q^* \geq q$. At each recursion layer l, the activation of the j-th ($j \in [0, 2^{\lceil log_2^n \rceil - l}$)) hidden node $\mathbf{h}^{(l)}_j \in \mathbb{R}^d$ is computed as

$$\mathbf{h}^{(l)}_j = \begin{cases} \mathbf{z}_N \odot \hat{\mathbf{h}}^l_j + \mathbf{z}_L \odot \mathbf{h}^{l-1}_{2j} + \mathbf{z}_R \odot \mathbf{h}^{l-1}_{2j+1}, & l > 1, \\ \text{corresponding word embedding}, & l = 1, \end{cases} \qquad (2)$$

where $\mathbf{z}_N$, $\mathbf{z}_L$ and $\mathbf{z}_R \in \mathbb{R}^d$ are update gates for new activation $\hat{\mathbf{h}}_j^l$, left child node $\mathbf{h}_{2j}^{l-1}$ and right child node $\mathbf{h}_{2j+1}^{l-1}$ respectively, and $\odot$ indicates element-wise multiplication.

The update gates can be formalized as:

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_N \\ \mathbf{z}_L \\ \mathbf{z}_R \end{bmatrix} = \begin{bmatrix} 1/Z \\ 1/Z \\ 1/Z \end{bmatrix} \odot \exp(\mathbf{U} \begin{bmatrix} \hat{\mathbf{h}}_j^l \\ \mathbf{h}_{2j}^{l-1} \\ \mathbf{h}_{2j+1}^{l-1} \end{bmatrix}), \quad (3)$$

where $\mathbf{U} \in \mathbb{R}^{3d \times 3d}$ is the coefficient of update gates, and $Z \in \mathbb{R}^d$ is the vector of the normalization coefficients,

$$Z_k = \sum_{i=1}^{3} [\exp(\mathbf{U} \begin{bmatrix} \hat{\mathbf{h}}_j^l \\ \mathbf{h}_{2j}^{l-1} \\ \mathbf{h}_{2j+1}^{l-1} \end{bmatrix})]_{d \times (i-1)+k}, \quad (4)$$

where $1 \leq k \leq d$.

The new activation $\hat{\mathbf{h}}_j^l$ is computed as:

$$\hat{\mathbf{h}}_j^l = \tanh(\mathbf{W}_{\hat{\mathbf{h}}} \begin{bmatrix} \mathbf{r}_L \odot \mathbf{h}_{2j}^{l-1} \\ \mathbf{r}_R \odot \mathbf{h}_{2j+1}^{l-1} \end{bmatrix}), \quad (5)$$

where $\mathbf{W}_{\hat{\mathbf{h}}} \in \mathbb{R}^{d \times 2d}$, $\mathbf{r}_L \in \mathbb{R}^d$, $\mathbf{r}_R \in \mathbb{R}^d$. $\mathbf{r}_L$ and $\mathbf{r}_R$ are the reset gates for left child node $\mathbf{h}_{2j}^{l-1}$ and right child node $\mathbf{h}_{2j+1}^{l-1}$ respectively, which can be formalized as:

$$\begin{bmatrix} \mathbf{r}_L \\ \mathbf{r}_R \end{bmatrix} = \sigma(\mathbf{G} \begin{bmatrix} \mathbf{h}_{2j}^{l-1} \\ \mathbf{h}_{2j+1}^{l-1} \end{bmatrix}), \quad (6)$$

where $\mathbf{G} \in \mathbb{R}^{2d \times 2d}$ is the coefficient of two reset gates and σ indicates the sigmoid function.

Intuiativly, the reset gates control how to select the output information of the left and right children, which result to the current new activation $\hat{\mathbf{h}}$. By the update gates, the activation of a parent neuron can be regarded as a choice among the the current new activation $\hat{\mathbf{h}}$, the left child, and the right child. This choice allows the overall structure to change adaptively with respect to the inputs.

This gate mechanism is effective to model the combinations of features.

2.3 Training

We use the Maximum Likelihood (ML) criterion to train our model. Given training set (x_i, y_i) and the parameter set of our model θ, the goal is to minimize the loss function:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \log \mathrm{P}(y_i|x_i;\theta) + \frac{\lambda}{2m}\|\theta\|_2^2, \quad (7)$$

Initial learning rate	$\alpha = 0.3$
Regularization	$\lambda = 10^{-4}$
Dropout rate on input layer	$p = 20\%$

Table 1: Hyper-parameter settings.

where m is number of training sentences.

Following (Socher et al., 2013a), we use the diagonal variant of AdaGrad (Duchi et al., 2011) with minibatchs to minimize the objective.

For parameter initialization, we use random initialization within (-0.01, 0.01) for all parameters except the word embeddings. We adopt the pre-trained English word embeddings from (Collobert et al., 2011) and fine-tune them during training.

3 Experiments

3.1 Datasets

To evaluate our approach, we test our model on three datasets:

- **SST-1** The movie reviews with five classes in the Stanford Sentiment Treebank[1] (Socher et al., 2013b): negative, somewhat negative, neutral, somewhat positive, positive.

- **SST-2** The movie reviews with binary classes in the Stanford Sentiment Treebank[1] (Socher et al., 2013b): negative, positive.

- **QC** The TREC questions dataset[2] (Li and Roth, 2002) involves six different question types.

3.2 Hyper-parameters

Table 1 lists the hyper-parameters of our model. In this paper, we also exploit dropout strategy (Srivastava et al., 2014) to avoid overfitting. In addition, we set the batch size to 20. We set word embedding size $d = 50$ on the TREC dataset and $d = 100$ on the Stanford Sentiment Treebank dataset.

3.3 Experiment Results

Table 2 shows the performance of our GRNN on three datasets.

[1] http://nlp.stanford.edu/sentiment

[2] http://cogcomp.cs.illinois.edu/Data/QA/QC/

Methods	SST-1	SST-2	QC
NBoW (Kalchbrenner et al., 2014)	42.4	80.5	88.2
PV (Le and Mikolov, 2014)	44.6*	82.7*	91.8*
CNN-non-static (Kim, 2014)	48.0	87.2	93.6
CNN-multichannel (Kim, 2014)	47.4	**88.1**	92.2
MaxTDNN (Collobert and Weston, 2008)	37.4	77.1	84.4
DCNN (Kalchbrenner et al., 2014)	**48.5**	86.8	93.0
RecNTN (Socher et al., 2013b)	45.7	85.4	-
RAE (Socher et al., 2011)	43.2	82.4	-
MV-RecNN (Socher et al., 2012)	44.4	82.9	-
AdaSent (Zhao et al., 2015)	-	-	92.4
GRNN (our approach)	47.5	85.5	**93.8**

Table 2: Performances of the different models. The result of PV is from our own implementation based on Gensim.

Competitor Models Neural Bag-of-Words (NBOW) model is a simple and intuitive method which ignores the word order. Paragraph Vector (PV) (Le and Mikolov, 2014) learns continuous distributed vector representations for pieces of texts, which can be regarded as a long term memory of sentences as opposed to short memory in recurrent neural network. Here, we use the popular open source implementation of PV in Gensim[1]. Methods in the third block are CNN based models. Kim (2014) reports 4 different CNN models using max-over-time pooling, where CNN-non-static and CNN-multichannel are more sophisticated. MaxTDNN sentence model is based on the architecture of the Time-Delay Neural Network (TDNN) (Waibel et al., 1989; Collobert and Weston, 2008). Dynamic convolutional neural network (DCNN) (Kalchbrenner et al., 2014) uses the dynamic k-max pooling operator as a non-linear sub-sampling function, in which the choice of k depends on the length of given sentence. Methods in the fourth block are RecNN based models. Recursive Neural Tensor Network (RecNTN) (Socher et al., 2013b) is an extension of plain RecNN, which also depends on a external syntactic structure. Recursive Autoencoder (RAE) (Socher et al., 2011) learns the representations of sentences by minimizing the reconstruction error. Matrix-Vector Recursive Neural Network (MV-RecNN) (Socher et al., 2012) is a extension of RecNN by assigning a vector and a matrix to every node in the parse tree. AdaSent (Zhao et al., 2015) adopts recursive neural network using DAG structure.

[1] https://github.com/piskvorky/gensim/

Moreover, the plain GRNN which does not incorporate the gate mechanism cannot outperform the GRNN model. Theoretically, the plain GRNN can be regarded as a special case of GRNN, whose parameters are constrained or truncated. As a result, GRNN is a more powerful model which outperforms the plain GRNN. Thus, we mainly focus on the GRNN model in this paper.

Result Discussion Generally, our model is better than the previous recursive neural network based models (RecNTN, RAE, MV-RecNN and AdaSent), which indicates our model can better model the combinations of features with the FBT and our gating mechanism, even without an external syntactic tree.

Although we just use the top layer outputs as the feature for classification, our model still outperforms AdaSent.

Compared with the CNN based methods (MaxTDNN, DCNN and CNNs), our model achieves the comparable performances with much fewer parameters. Although CNN based methods outperform our model on SST-1 and SST-2, the number of parameters[2] of GRNN ranges from 40K to 160K while the number of parameters is about 400K in CNN.

4 Related Work

Cho et al. (2014) proposed grConv to model sentences for machine translation. Unlike our model, grConv uses the DAG structure as the topological structure to model sentences. The number of the

[2] We only take parameters of network into account, leaving out word embeddings.

internal nodes is $n^2/2$, where n is the length of the sentence. Zhao et al. (2015) uses the same structure to model sentences (called AdaSent), and utilizes the information of internal nodes to model sentences for text classification. Unlike grConv and AdaSent, our model uses full binary tree as the topological structure. The number of the internal nodes is $2n$ in our model. Therefore, our model is more efficient for long sentences. In addition, we just use the top layer neurons for text classification.

Moreover, grConv and AdaSent only exploit one gating mechanism (update gate), which cannot sufficiently model the complicated feature combinations. Unlike them, our model incorporates two kind of gates and can better model the feature combinations.

Hu et al. (2014) also proposed a similar architecture for matching problems, but they employed the convolutional neural network which might be coarse in modeling the feature combinations.

5 Conclusion

In this paper, we propose a gated recursive neural network (GRNN) to recursively summarize the meaning of sentence. GRNN uses full binary tree as the recursive topological structure instead of an external syntactic tree. In addition, we introduce two kinds of gates to model the complicated combinations of features. In future work, we would like to investigate the other gating mechanisms for better modeling the feature combinations.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work was partially funded by the National Natural Science Foundation of China (61472088, 61473092), National High Technology Research and Development Program of China (2015AA015408), Shanghai Science and Technology Development Funds (14ZR1403200).

References

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 2015a. Gated recursive neural network for Chinese word segmentation. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Xinchi Chen, Yaqian Zhou, Chenxi Zhu, Xipeng Qiu, and Xuanjing Huang. 2015b. Transition-based dependency parsing using two heterogeneous gated recursive neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

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

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of ICML*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 556–562.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*.

Jordan B Pollack. 1990. Recursive distributed representations. *Artificial Intelligence*, 46(1):77–105.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211.

Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. 2013a. Parsing with compositional vector grammars. In *In Proceedings of the ACL conference*. Citeseer.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. 1989. Phoneme recognition using time-delay neural networks. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(3):328–339.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. *arXiv preprint arXiv:1504.05070*.