Sentiment Analysis with Tree-Structured Gated Recurrent Units

Marcin Kuta(✉), Mikołaj Morawiec, and Jacek Kitowski

Department of Computer Science, AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Krakow, Poland
mkuta@agh.edu.pl

Abstract. Advances in neural network models and deep learning mark great impact on sentiment analysis, where models based on recursive or convolutional neural networks show state-of-the-art results leaving behind non-neural models like SVM or traditional lexicon-based approaches. We present Tree-Structured Gated Recurrent Unit network, which exhibits greater simplicity in comparison to the current state of the art in sentiment analysis, Tree-Structured LSTM model.

Keywords: Sentiment analysis · Recursive neural network · Gated Recurrent Unit · Tree-Structured GRU · Long Short-Term Memory

1 Introduction

Sentiment analysis is the problem of assigning sentiment to a document, sentence or a phrase. A document may be a movie review or an opinion about a particular product. Finding an accurate solution to the sentiment analysis problem has strong economic justification, as it allows companies to find opinions about their products and recognize their characteristics.

Currently neural network based models achieve the most competitive results in sentiment analysis. One challenge for neural network architectures is handling an input of variable length. Recurrent Neural Networks (RNNs) easily process sequences of variable length, unfortunately are hard to train, due to vanishing or exploding gradient problems. Long Short Term Memory (LSTM) units were proposed [6] as a remedy to vanishing gradient, the most ubiquitous problem encountered during RNNs training. Over the years, numerous extensions and refinements of the original LSTM, including adding peephole cells, have been developed [5].

Recently, significant simplification over LSTM, known as Gated Recurrent Unit (GRU), was introduced [2]. GRUs contain less subcells and are described by much simpler set of equations, thus require less computational power. Relation between GRU and LSTM effectiveness is an open issue and an area of research. Evaluation of GRU-based neural networks on sequence modelling [3] showed effectiveness similar to those build from LSTMs, while in [8] GRU outperformed the LSTM on nearly all tasks except language modelling with the naive initialization.

K. Ekštein and V. Matoušek (Eds.): TSD 2017, LNAI 10415, pp. 74–82, 2017.
DOI: 10.1007/978-3-319-64206-2_9

This paper proposes the Tree-Structured Gated Recurrent Unit (TS-GRU) model for sentiment analysis and compares it with the state-of-the-art Tree-Structured Long Short Term Memory (TS-LSTM) [16] and other models in terms of effectiveness. The model is evaluated on the Stanford Sentiment TreeBank (SST) for the binary (number of classes, $C = 2$) and fine-grained ($C = 5$) sentiment classification problems.

2 Related Work

The promising avenue of research in sentiment analysis opened up with expansive growth of deep learning. A broad range of neural networks was already harnessed to the sentiment analysis problem, including recurrent and recursive neural networks, convolutional neural networks, autoencoders and Restricted Boltzmann Machines (RBMs). The common aspect of these models is that they do not rely on man-made features.

Recursive autoencoders (RvAEs), introduced into sentiment analysis by Socher [14], work over a parse tree of a sentence, build by an external parser. RvAEs were further extended into Matrix-Vector Recursive Neural Networks (MV-RNNs) [13]. The original idea of this model is that an embedding is not represented by a vector, but by a (vector, matrix) pair. This increases representational power of phrases, but very large number of additional parameters introduced with matrices makes the model more prone to overfitting. The next model, Recursive Neural Tensor Network (RNTN) [15], tries to alleviate this problem. Embeddings are represented back with vectors only, but equations define the model using tensors instead of matrices. Tensor multiplications give the model enough representational power, at the same time solving the problem with overfitting.

Current state of the art in the fine-grained sentiment analysis, measured on the Stanford Sentiment TreeBank, is achieved by Tree-Structured LSTMs [16]. Similarly to other recursive neural networks, this approach uses a parse tree as a backbone for sentiment signal propagation. An alternative to Tree-Structured LSTMs way of combining in a parent node sentiment signals from children, named S-LSTMs, was proposed in [19].

Document level sentiment can be determined by hierarchically building document neural representation on the basis of neural representations of its sentences. Gated recurrent neural networks with LSTM or convolutional components achieve here state-of-the-art results, as measured on Yelp and IMBD datasets [18].

3 Tree-Structured GRU Model

3.1 LSTM and GRU

Let $\boldsymbol{x}$ be an input sequence of length T, i.e., $\boldsymbol{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(T)})$, and let N be dimensionality of its elements, i.e., each $x^{(t)} \in \mathbb{R}^N$.

LSTMs are able to remember long-term dependencies due to the presence of memory cell $s^{(t)}$. The flow of a signal is controlled by three gates: input gate $i^{(t)}$, forget gate $f^{(t)}$ and output gate $o^{(t)}$. Cell $g^{(t)}$ denotes candidate hidden state on the basis of which hidden state $h^{(t)}$ is computed. Dimensionality of $h^{(t)}$ and other LSTM components equals M. Values of LSTM cells and gates are determined according to the following equations ($\odot$ denotes the element-wise multiplication – the Hadamard product):

$$\begin{aligned} \begin{bmatrix} g^{(t)} \\ i^{(t)} \\ f^{(t)} \\ o^{(t)} \end{bmatrix} &= \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} W \begin{bmatrix} x^{(t)} \\ h^{(t-1)} \end{bmatrix} + b \\ s^{(t)} &= g^{(t)} \odot i^{(t)} + s^{(t-1)} \odot f^{(t)} \\ h^{(t)} &= \tanh(s^{(t)}) \odot o^{(t)} \end{aligned} \tag{1}$$

where $W \in \mathbb{R}^{4M \times (N+M)}$, bias $b \in \mathbb{R}^{4M}$ and σ is the sigmoid logistic function.

Structure of GRU, shown in Fig. 1, is simpler than the LSTM one. GRU contains only two gates: reset gate $r \in \mathbb{R}^M$ and update gate $z \in \mathbb{R}^M$. Compared to LSTM, GRU does not have an output gate and is defined by a simpler set of equations:

$$\begin{aligned} z &= \sigma\left(U^{\mathrm{z}} x^{(t-1)} + V^{\mathrm{z}} h^{(t-1)}\right) \\ r &= \sigma\left(U^{\mathrm{r}} x^{(t-1)} + V^{\mathrm{r}} h^{(t-1)}\right) \\ \bar{h} &= \tanh\left(U^{\mathrm{h}} x^{(t-1)} + V^{\mathrm{h}}(h^{(t-1)} \odot r)\right) \\ h^{(t)} &= (\mathbf{1} - z) \odot \bar{h} + z \odot h^{(t-1)} \end{aligned} \tag{2}$$

Before computing hidden state $h^{(t)} \in \mathbb{R}^M$, candidate hidden state $\bar{h} \in \mathbb{R}^M$ must be determined. Vector $\mathbf{1}$ denotes M-dimensional vector composed of ones.

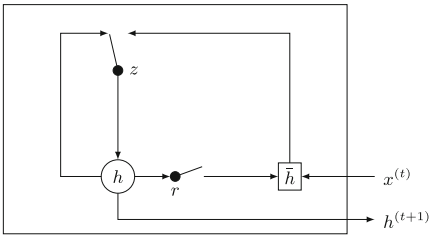

Fig. 1. Structure of Gated Recurrent Unit. Source: [2]

3.2 Model

GRU is designed for work with linear structures like sequences coped with RNNs. GRU version adjusted to operations on branching structures, called grConv, was proposed and applied to machine translation [1].

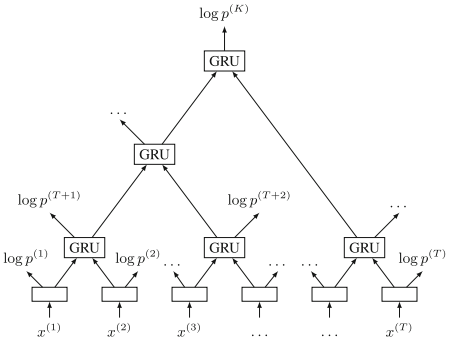

Fig. 2. Tree-Structured GRUs network over a parse tree

Tree-Structured GRU extension proposed in the paper is able to cope with recursive structures like parse trees (Fig. 2). The model is determined by parameters θ: $W^{\mathrm{x}} \in \mathbb{R}^{M \times N}$; $U^{\mathrm{z}1}, V^{\mathrm{z}1}, U^{\mathrm{z}2}, V^{\mathrm{z}2}, U^{\mathrm{r}}, V^{\mathrm{r}}, U^{\mathrm{h}}, V^{\mathrm{h}} \in \mathbb{R}^{M \times M}$; $W^{\mathrm{y}} \in \mathbb{R}^{C \times M}$; and $F \in \mathbb{R}^{N \times \#W}$, where $\#W$ is the number of different words in the training set.

Tree-Structured GRU is governed by the following equations:

$$\begin{aligned} z^i &= \sigma\big(U^{\mathrm{z}i} h_1 + V^{\mathrm{z}i} h_2\big), \qquad i = 1, 2 \\ r &= \sigma\big(U^{\mathrm{r}} h_1 + V^{\mathrm{r}} h_2\big) \\ \bar{h} &= \tanh\big(U^{\mathrm{h}}(h_1 \odot r) + V^{\mathrm{h}}(h_2 \odot r)\big) \\ h &= (\mathbf{1} - \sum_{i=1}^{2} z^i) \odot \bar{h} + \sum_{i=1}^{2} z^i \odot h_i \end{aligned} \tag{3}$$

States h_1 and h_2 denote hidden states of the left and right child unit of a parent GRU. Gate z was implemented as two binary gates z^1 and z^2.

The model works as follows. At first, each element $x \in \mathbb{R}^N$ of input sequence $\boldsymbol{x}$, represented with a GloVe vector [11], is projected onto the input hidden state $h \in \mathbb{R}^M$:

$$h = \tanh\big(W^{\mathrm{x}} x\big). \tag{4}$$

Hidden states are propagated up the tree according to TS-GRU definition (3). On the basis of the hidden state, h, each GRU computes its output signal, $\log p \in \mathbb{R}^C$, with the logsoftmax function according to (6). To improve network effectiveness regularization dropout layer (5) is applied in-between h and p. In experiments dropout ratio was set to $1/2$.

$$\tilde{h} = \mathrm{dropout}\big(h\big) \tag{5}$$

$$p = \mathrm{softmax}(W^{\mathrm{y}} \tilde{h}) = \frac{\exp(W^{\mathrm{y}} \tilde{h})}{\sum_{j'} \exp(W^{\mathrm{y}}_{j' \cdot} \tilde{h})} \tag{6}$$

$$\hat{c} = \arg\max_k p_k \tag{7}$$

Output signal is compared with the true sentiment and error is backpropagated down a parse tree. At the input, error signal is backpropagated through fine-tuning matrix F to pre-input storing fine-tuned version of the input vectors.

The cost function $J(\theta)$ used as the training criterion is defined for one sentence as follows:

$$J(\theta) = -\frac{1}{K}\sum_{i=1}^{K} \log p_c^{(i)} + \frac{\lambda}{2}||\theta||_2^2, \tag{8}$$

where sum goes over all nodes of a parse tree. The number of such nodes, K, equals $2T-1$, as the network is spanned over a binary constituency tree. Weight decay parameter, λ, determines the importance of the regularization term $||\theta||_2^2$.

Vector $p^{(i)}$ (6) contains probabilities a word or phrase belongs to each of C classes, and $p_c^{(i)}$ is just the probability corresponding to the true class $c \in \{1, \ldots, C\}$, denoting the correct word or phrase sentiment read from the SST gold standard. The sentiment class $\hat{c}$ of a word or phrase returned by the model is determined according to (7).

4 Experiments and Results

Experiments were conducted on the Stanford Sentiment TreeBank (SST) [15], containing movie reviews. SST consists of 11 855 sentences parsed with the Stanford Parser into 239 232 phrases. Each sentence and phrase goes with the assigned sentiment, s, being a real number in interval [0,1]. Using appropriate cutoffs, this number can be mapped into one of 5 fine-grained sentiment classes: very negative, negative, neutral, positive and very positive. Removing neutral opinions and merging together very negative and negative classes into one class, and similarly merging positive and very positive classes we obtain coarse-grained, binary sentiment classification version of the problem. The mapping is done with functions f_V and f_{II}:

$$f_V(s) = \begin{cases} 1, & \text{for } s \in [0, 0.2) \\ 2, & \text{for } s \in [0.2, 0.4) \\ 3, & \text{for } s \in [0.4, 0.6) \\ 4, & \text{for } s \in [0.6, 0.8) \\ 5, & \text{for } s \in [0.8, 1] \end{cases} \quad \text{and} \quad f_{II}(s) = \begin{cases} 1, & \text{for } s \in [0, 0.4) \\ 2, & \text{for } s \in [0.6, 1] \end{cases} \tag{9}$$

SST comes with the predefined split to the training, optimization, and test set, containing 8544, 1101 and 2210 sentences respectively for the fine-grained version. In the binary sentiment classification 6920, 872 and 1821 sentences from SST were used, respectively.

There are 24860 different words in SST, of which 15665 were initialized with GloVe vectors and remaining words, not present in the GloVe dictionary, got random initialization. The model parameters, θ, were initialized from the uniform distribution $\mathcal{U}(-\frac{1}{\sqrt{D}}, \frac{1}{\sqrt{D}})$, where D is a dimension of the given layer input,

i.e., $D = N$ for W^{x}, $D = M$ for U^{r}, etc. The size of GloVe embedding was fixed to $N = 300$.

The best model was selected through the full grid search of the following hyperparameters: learning rate $\varepsilon \in \{0.005, 0.05, 0.1\}$, size of the hidden layer $M \in \{40, 60, 80, 100, 150, 200\}$, and weight decay $\lambda \in \{10^{-4}, 2 \cdot 10^{-4}, 10^{-3}\}$. Both in the binary and fine-grained problem the highest effectiveness on the optimization set was achieved for hyperparameters $\varepsilon = 0.05$, $M = 100$, and $\lambda = 10^{-4}$.

The network was trained with the AdaGrad algorithm in mini-batches of 25 samples. The optimization algorithm itself was model selected from 3 algorithms: SGD, AdaGrad and Adam. The error signal was propagated with the Backpropagation Through Structure (BTS) algorithm [4]. The parameters of the optimal model, θ, were found for the network trained 4 epochs (the fine-grained problem) and 6 epochs (the binary problem), when the highest accuracy was achieved on the optimization set.

Accuracy of various approaches to sentiment classification is compared in Table 1. For sentences TS-GRU achieved 49.28% accuracy in the fine-grained classification and 76.16% accuracy in the binary classification. Analysis of sentiment of phrases is always simpler, TS-GRU classified them with 66.50% (fined-grained sentiment) and 77.80% (binary sentiment) accuracy.

TS-GRUs revealed significant impact of proper initialization of input vectors – initialization with subsequently fine-tuned GloVe vectors showed over 6% improvement over a random initialization. This behaviour is similar to TS-LSTMs, where 7% improvement was achieved. Exact impact of input vector initialization on sentiment accuracy is shown in Table 2.

Table 1. Accuracy of sentiment classification of sentences on the test set of SST, [%]

Method	Fine-grained	Binary
TS-LSTM [16]	**51.0**	88.0
TS-GRU	49.2	76.1
S-LSTM [19]	48.0	n/a
Bidirectional LSTM [16]	49.1	87.5
CNN-multichannel [10]	47.4	**88.1**
DCNN [9]	48.5	86.8
CharSCNN [12]	48.3	85.7
Deep RvNN [7]	49.8	86.6
RNTN [15]	45.7	85.4
MV-RNN [15]	44.4	82.9
RvAE [15]	43.2	82.4
Naïve Bayes [15]	41.0	81.8
SVM [15]	40.7	79.4

Table 2. Impact of initialization of input tokens on sentiment classification accuracy, [%]

Gate	Representation	Fine-grained	Binary
GRU	GloVe fine-tuned	49.2	76.1
GRU	GloVe fixed	48.9	74.1
GRU	Random	42.8	73.3
LSTM	GloVe fine-tuned	51.0	88.0
LSTM	GloVe fixed	49.7	87.5
LSTM	Random	43.9	82.0

The number of parameters of the TS-GRU network for $C = 5$ equals $8M^2 + MN + CM + N \cdot \#W$, i.e., 7568500 parameters when fine-tuning is applied and $8M^2 + MN + CM$, i.e., 110500 parameters when fixed vectors are used. The corresponding TS-LSTM network [16] needed 316800 parameters for the input represented with $N = 300$ dimensional GloVe vectors.

5 Conclusions

In this paper we proposed the TS-GRU network, which adapts GRU to recursive networks, spanned over a parse tree. The model was inspired by the TS-LSTM network and the gated recursive convolutional neural network (grConv).

TS-GRU network achieved high accuracy on the binary sentiment classification, ranking third, behind TS-LSTM and Deep RvNN, although it performed badly on the fine-grained sentiment classification. Without input vectors fine-tuning, the TS-GRU network needed, however, three times less parameters than the TS-LSTM network.

Obtained accuracies are also comparable with human judgments, which vary between 70%–90% effectiveness, in particular [17] reports 83.6%–87.9% human accuracy on a different evaluation set.

Acknowledgments. This research was supported in part by PL-Grid Infrastructure. The research was also partially financed by AGH University of Science and Technology Statutory Fund.

References

1. Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder-decoder approaches. In: Wu, D., Carpuat, M., Carreras, X., Vecchi, E.M. (eds.) Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pp. 103–111 (2014)
2. Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1724–1734 (2014)

3. Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR abs/1412.3555 (2014)
4. Goller, C., Küchler, A.: Learning task-dependent distributed representations by backpropagation through structure. In: Proceedings of the International Conference on Neural Networks, ICNN 1996, pp. 347–352 (1996)
5. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: a search space Odyssey. arXiv preprint arxiv:1503.04069 (2015)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Irsoy, O., Cardie, C.: Deep recursive neural networks for compositionality in language. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 27, pp. 2096–2104 (2014)
8. Józefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: Bach, F.R., Blei, D.M. (eds.) Proceedings of the 32nd International Conference on Machine Learning, pp. 2342–2350 (2015)
9. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Toutanova, K., Wu, H. (eds.) Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 655–665 (2014)
10. Kim, Y.: Convolutional neural networks for sentence classification. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751 (2014)
11. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1532–1543 (2014)
12. dos Santos, C., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: Tsujii, J., Hajič, J. (eds.) Proceedings of COLING 2014, The 25th International Conference on Computational Linguistics, Technical Papers, pp. 69–78 (2014)
13. Socher, R., Huval, B., Manning, C.D., Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: Tsujii, J., Henderson, J., Paşca, M. (eds.) Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, pp. 1201–1211 (2012)
14. Socher, R., Pennington, J., Huang, E.H., Ng, A.Y., Manning, C.D.: Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Barzilay, R., Johnson, M. (eds.) Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 151–161 (2011)
15. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)
16. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Zong, C., Strube, M. (eds.) Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 1556–1566 (2015)

17. Takala, P., Malo, P., Sinha, A., Ahlgren, O.: Gold-standard for topic-specific sentiment analysis of economic texts. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC, vol. 2014, pp. 2152–2157 (2014)
18. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Màrquez, L., Callison-Burch, C., Su, J. (eds.) Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1422–1432 (2015)
19. Zhu, X., Sobhani, P., Guo, H.: Long short-term memory over recursive structures. In: Bach, F.R., Blei, D.M. (eds.) Proceedings of the 32nd International Conference on Machine Learning, pp. 1604–1612 (2015)