Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 11; April-June, 2015 pp. 59-63

http://www.krishisanskriti.org/acsit.html

Sentiment Analysis with Gated Recurrent Units

Shamim Biswas[1], Ekamber Chadda[2] and Faiyaz Ahmad[3]

[1,2,3]Department of Computer Engineering Jamia Millia Islamia New Delhi, India
E-mail: [1]shamimbiswas93@gmail.com, [2]ekamberc @gmail.com, [3]ahmad.faiyaz@gmail.com

Abstract—*Sentiment analysis is a well researched natural language processing field. It is a challenging machine learning task due to the recursive nature of sentences, different length of documents and sarcasm. Traditional approaches to sentiment analysis use count or frequency of words in the text which are assigned sentiment value by some expert. These approaches disregard the order of words and the complex meanings they can convey. Gated Recurrent Units are recent form of recurrent neural network which have the ability to store information of long term dependencies in sequential data. In this work we showed that GRU are suitable for processing long textual data and applied it to the task of sentiment analysis. We showed its effectiveness by comparing with tf-idf and word2vec models. We also showed that GRUs are faster in convergence than LSTM, another gating network. We applied a number of modifications to the standard GRU to make it train faster and yet less prone to over training. We found the better performimg hyperparameters of the GRU-net through extensive cross-validation testing. Finally we ensembled the best performing GRU models for even better performance.*

Keywords*: Sentiment analysis; gated recurrent unit ; kaggle; ensemble;*

1. INTRODUCTION

Sentiment analysis is the process of extracting the attitude of the author towards the entity the text is written. At the simplest level sentiment analysis is a binary classification task which involves deciding whether a text is positive or negative. Sentiment analysis is done at both at phrase level and document or paragraph level. Both the levels offer unique challenges and hence require different techniques to tackle them.

There is an increasing availability of various websites which allows customers to write reviews on various products, movies or hotels. In the reviews the customers share their experience of using that particular service. Sentiment analysis is applied to such reviews to discover the customers opinion on that service. This is of important value to companies, stocks and politics.

The advent of deep learning approaches has given rise to a number of new methods for sentiment analysis. The availability of large unlabeled textual data can be used to learn the meanings of words and the structure of sentence formation. This has been attempted by word2vec [1] which learns word embeddings from unlabeled text samples. It learns both by predicting the word given its surrounding words(CBOW) and predicting surrounding words from given word(SKIP-GRAM). These word embeddings are used for creating dictionaries and act as dimensionality reducers in traditional method like tf-idf etc. More approaches are found capturing sentence level representations like recursive neural tensor network (RNTN) [2]

Convolution neural network which has primarily been used for image related task has been shown effective in text classification too [3]. The main problem is the variable length of the natural language. Some of it is solved by fixed size context windows but it fails to capture dependencies which extend longer than the context window. Recurrent neural network have the ability to take variable length of text sequence but they are extremely tricky to learn. Hence new types of rnn were employed like LSTM and GRU. LSTM was proposed in 1997 by Hochreiter et al.[4] and is making news in many nlp task like sentiment analysis, translation and sequence generation. GRUs is quite recent development proposed by K. Cho[5] in 2014. GRU are much simpler in structure and probably more practical than LSTM. We attempt to show its advantages over LSTM in sentiment analysis in this work.

In this study we did sentiment analysis at paragraph level of movie reviews as positive or negative. Different approaches used by us are Tf-idf, Word2Vec (vector average), Word2Vec (k – means dictionary), GRU (gated recurrent units) and Ensemble model. Ensemble model consisted of different three GRU learning models whose results were combined using logistic regression. It was found that among all the single models GRU outperformed all of them, whereas this result of GRU model was further improved after using ensemble model.

We found that GRUs were effective in the task of sentiment analysis because of their ability to remember long time dependencies. GRU are specially useful for large texts. We showed that LSTM networks were quite slow to train and did not converge to better accuracy even after many epochs. Hence we propose GRUs as better replacements to LSTM.

The organization of the rest of the paper is as follows. In section 2, we describe the general recurrent neural networks

and their specializations used here. We also explain various modification done to make GRU more suitable for sentiment analysis. In section 3 we describe our dataset, experimental setup and evaluate our technique. In section 4 we conclude the project and describe future work.

2. MODELS

2.1 Recurrent Neural Network

A recurrent neural network (RNN) is similar to conventional feed forward network, with the difference that it has connections to units in the same layer. This provides them an internal memory which is able to handle a variable length input sequence. It handles the variable length sequences by having a recurrent hidden state whose activation at each time is dependent on that of previous time. The human brain is a recurrent neural network, a network of neurons with feedback connections and hence using them brings us closest to natural design.

However, RNNs are difficult to train and suffer from vanishing and exploding gradients problem. Either the gradients become so small that learning stops or the gradient becomes so large that the weights overflow the max limit. The most effective solution to this problem is adding a gating mechanism to the RNN. Two gated RNN are found in literature

- Long Short Term Memory(LSTM)
- Gated Recurrent Unit(GRU)

3. LSTM

The Long Short-Term Memory (LSTM) unit was initially proposed by Hochreiter and Schmidhuber[4]. It has undergone many changes over the years like the addition of forget gate.

LSTM consists of a memory cell to store information. It computes input gate, forget gate and output gate to manage this memory. LSTM units can propagate an important feature that came early in the input sequence over a long distance, hence, capturing potential long-distance dependencies.

LSTM despite being complex are very successful in various tasks like handwriting recognition, machine translation and of course sentiment analysis.

4. GRU

The gated recurrent unit GRU is relatively recent development proposed by Cho et al. [5]. Similar to the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without having a separate memory cells. Gated Recurrent Unit (GRU) calculates two gates called update and reset gates which control the flow of information through each hidden unit. Each hidden state at time-step t is computed using the following equations:

Update gate:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$$

Reset gate:

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$$

New memory:

$$\tilde{h_t} = tanh(Wx_t + r_t \, o \, Uh_{t-1})$$

Final memory:

$$\tilde{h_t} = z_t \, o \, h_{t-1} + (1 - z_t) \, o \, \tilde{h}_{t-1}$$

where o is element wise multiplication and σ is the sigmoid function

The update gate is calculated from the current input and the hidden state of previous time step. This gate controls how much of portions of new memory and old memory should be combined in the final memory. Similarly the reset gate is calculated but with different set of weights. It controls the balance between previous memory and the new input information in the new memory.

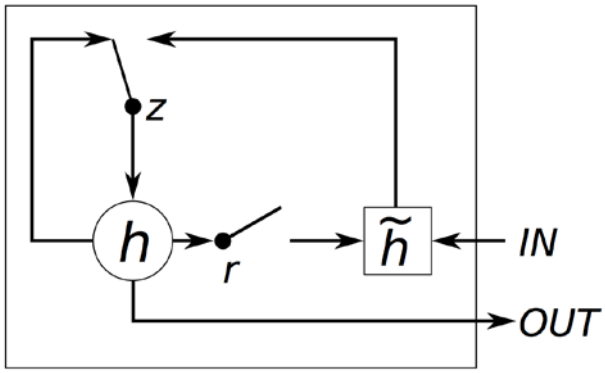

Fig. 1: Internal structure of a gated recurrent unit [5]

5. GRU FOR TEXT CLASSIFCIATION

The GRU can be used for text classification in the similar manner as it has been used for LSTM by Le P. et al. [6] and Cho K [5,17]. First a embedding layer of appropriate size is created. Then the words in the phrase are fed into the first layer as 1 of K encoding while maintaining their order. The embedding layer will learn to represent each word by a real valued vector of size equal to the fixed dimension. These values are the weights between the embedding layer and the hidden layer on top of it. The hidden layer consists of gated recurrent units. These are not only connected to the layer below and above them but also connected to units within their own layer. We used only single layered GRUs even though it is possible to stack them. At the end of the hidden layer we get the representation of the entire sequence which can be used as input to linear model or classifier. We used sigmoid function

for binary classification in the dense layer(output layer). Similarly a softmax classifier can be used for multiclass classification task.

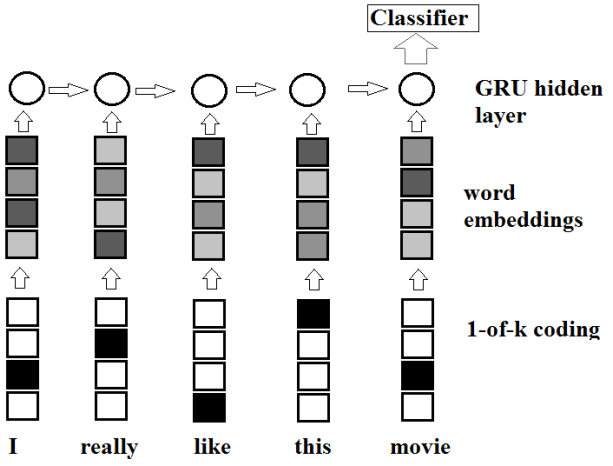

Fig. 2: Using GRU net for text classification

6. OPTIMIZATIONS

Recurrent Neural Networks like GRU require special learning algorithms and other modifications to make them practical.

As suggested by Radford in his fantastic presentation [7]

GRU network can be optimized in the following as:

- STEEPER GATES
- The slope of usual sigmoid functions is quite less and hence when weights are initialized randomly, most of values come close to zero on the y axis. This can be avoided by using a sigmoid function with a higher slope. This results in quicker learning because of higher separation in sigmoid outputs
- BETTER OPTIMIZERS
- A variety of modern optimizer shave been developed in recent times which have both faster convergence rates and also better long term accuracy.. We used Adadelta[8] for this task.
- DROPOUT
- It is very effective method of preventing overtraining in neural networks Srivastava N. et al. 2014 [9]. The key idea is to randomly leave units with a probability p during training the neural network. The loss is recovered by multiplying net output by product of probability p and number of units in the dropout layer. This prevents the units from adapting too much.
- ORTHOGONAL INITIALIZATION
- It has been shown that initialization of weight matrices with random orthogonal matrices works better than random Gaussian (or uniform) matrices[10].

7. RESULTS

7.1 DATASET

The results for this paper were obtained using the IMDB dataset originally collected by Andrew Maas [11] . It consists of the labeled data set of 50,000 IMDB movie reviews, specially selected for sentiment analysis. The sentiment of reviews is binary, meaning the IMDB rating < 5 results in a sentiment score of 0, and rating >=7 have a sentiment score of 1. Additionally it has 50,000 unlabelled movie reviews which we useful for unsupervised training.

7.2 EXPERIMENTAL SETUP

All our experiments were performed on a system running Ubuntu 14.04 with 2.4GHz dual core Intel i3 CPU , 4 GB RAM and nvidia GeForce 620m 1GB GPU. We used Passage[12], a python based library for text analysis with RNNs for implementing our GRU and LSTM models. We used the Gensim[13] package for python implementation of word2vec we used for comparison. The computation power of the GPU was exploited by using the theano[14] python library. The average learning time of GRU-NET with 10 epochs was 40 minutes.

7.3 EXPERIMENTS

We ran our implementations of LSTM and GRU nets on IMDB dataset with identical parameters. We fixed both the word embedding dimwnsions and number of units to 64. And ran the models for 10 epochs. We found that GRU net converged faster than LSTM and attained lower error even after many epochs. It was also found that GRU net learned without any oscillations like in LSTM.

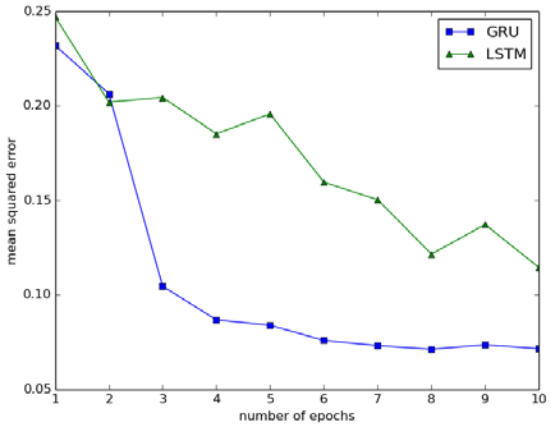

Fig. 3: Comparison of GRU and LSTM on cross-validation test error.

We then compare the GRU network with traditional models used in sentiment analysis. Table 1 reports the scores of the various models we implemented on the IMDB dataset. The

scores are area under the curve (AUC) for binary classification. Firstly we implemented the tf-idf model and tacked on logistic regression over it. This was the baseline for our results. Secondly we implemented two word2vec models, one by averaging all the word vectors in the review and the other by clustering all similar words in the vocabulary with Kmeans and giving the same index in the dictionary. In both the models the word2vec was trained on 50,000 unlabelled reviews with 300 word vector dimensions by the skip-gram method. We found that both the word2vec models performed worse than tf-idf model. Finally we implemented the LSTM and GRU model based on Passage rnn toolkit. We fixed both the embedding layer dimensions and number of units in hidden layer as 128. LSTM network was run for 35 epochs to reach the shown accuracy while GRU was trained in just 10 epochs. We found GRU model outperformed all the other models by a significant margin.

A general trend was observed that more number of dimensions in word embeddings and gated units resulted in better performance. But the improvements decrease substantially as they are increased beyond a certain limit. We also found that keeping the embedding dimensions equal to the number of gated units performed better than networks having units much more than word embedding dimensions.

Table 1: Classification accuracies of gru compared with other models

MODEL	AUC SCORE
tf-idf	0.89588
Word2Vec(vetor average)	0.55564
Word2Vec(Kmeans dictionary)	0.80140
LSTM	0.95707
GRU	0.970000

We designed 3 models of GRU of different dimensions of embedding layer and number of recurrent units. The learning rate of adadelta was 0.5 and dropout rate was 0.75 for all three models. The hyperparameters were carefully chosen by trying different configurations and checking the cross-validation scores. Finally for even better results we ensemble the 3 models by stacking a logistic regression models on top of them. As expected it produced scores higher than any of the individual models. Table 2 reports the AUC score of the different GRU models.

Table 2: Classification accuracies of gru compared with other models

GRU MODEL		AUC SCORE
EMBEDDING SIZE	NO OF GRU	
64	22	0.96800
64	64	0.96992
128	128	0.97000
ENSEMBLE		0.97090

Finally we also used GRU to phrase level sentiment analysis using the Stanford Sentiment Treebank [2]. GRU scored a respectable 0.65201 multi classification accuracy. We found that GRU did not do well for very short texts because then it cannot take the advantage of its capacity to remember long sequences.

8. CONCLUSIONS

Sentiment Analysis remains popular and important field of natural language processing. We conclude that gated recurrent units are a suitable model for sentiment analysis especially at paragraph level. Being a recurrent network it can effectively capture long sequence data required for natural language understanding. They perform better than traditional bag of features models which disregard the order of the features. They eliminate the problem of exploding and diminishing gradient problem as effectively as the LSTM networks does with lesser computational overhead. The GRU also converges to the solution faster than LSTM. The GRU networks hence look promising and may create the state of the art in sentiment analysis and other NLP tasks.

REFERENCES

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

[2] *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank*, Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Chris Manning, Andrew Ng and Chris Potts. Conference on Empirical Methods in Natural Language Processing (EMNLP 2013).

[3] Yoon Kim. Convolution Neural Network for Sentence Classification, EMNLP 2014

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[5] Cho, K., van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

[6] Compositional Distributional Semantics with Long Short Term Memory, Phong Le and Willem Zuidema,2015

[7] Alec Radford, https://indico.io/blog/passage-text-analysis-with-recurrent-neural-nets-next-ml-cambridge/

[8] Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

[9] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.

[10] Saxe, A. M., McClelland, J. L., & Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

[11] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). "Learning Word Vectors for Sentiment Analysis." *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*

[12] Alec Radford, https://github.com/IndicoDataSolutions/Passage

[13] Řehůřek, R., & Sojka, P. (2011). Gensim—Statistical Semantics in Python.

[14] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., ... & Bengio, Y. (2010, June). Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)* (Vol. 4, p. 3).

[15] Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio

[16] https://www.kaggle.com/c/word2vec-nlp-tutorial

[17] Cho K., http://deeplearning.net/tutorial/lstm.html