GRAPH CONVOLUTIONAL NEURAL NETWORKS FOR SENTIMENT ANALYSIS

Vojtech Myska
Doctoral Degree Programme 2^{nd} year, FEEC BUT
E-mail: xmyska04@stud.feec.vutbr.cz

Supervised by: Radim Burget
E-mail: burgetrm@feec.vutbr.cz

Abstract: Commonly used approaches based on deep learning for sentiment analysis task operating over data in Euclidean space. In contrast with them, this paper presents, a novel approach for sentiment analysis task based on a graph convolutional neural networks (GCNs) operating with data in Non-Euclidean space. Text data processed by the approach have to be converted to a graph structure. Our GCNs models have been trained on 25 000 data samples and evaluated 5 000 samples. The Yelp data set has been used. The experiment is focused on polarity sentiment analysis task. Nevertheless, a relatively small training data set has been used, our best model achieved 86.12% accuracy.

Keywords: sentiment analysis, graph neural networks, deep learning

1 INTRODUCTION

Sentiment analysis belongs to one of the important task of natural language processing (NLP). The aim is to reveal the author's opinion and classify it into one of the considered classes, e.g. 5 star task takes into account positive, negative, neutral, more positive or more negative class.

The most of current approaches solve this task by using models based on convolutional (CNNs) or recurrent neural networks (RNNs). CNN-based approaches take advantage of computer vision success and treat text as an image (transform text into a 2D matrix). A different approach is used in RNN-based models which consider text as time-series.

Mentioned approaches work with text transformed into Euclidean space. These data have limited expressive power in compare of graphs. For this reason it is more useful to work with text in graph domain. It is necessary to use a different kind of neural networks. Graph neural networks (GNNs) presented in work [2] is capable to operate over a graph structure.

This paper introduces a novel approach designed for sentiment classification task based on graph convolutional networks operating over graph structure. The introduced approach is not working only with word meaning but also with part-of-speech as well as relations between words. This information is used in process of building a graph structure and in classification process.

2 RELATED WORKS

The most important approach based on CNNs has been published in 2015 [9]. In this work has been demonstrated the ability to successfully classify converted text into the 2D matrix by using a CNN-based model. This model has reached a relatively high accuracy (95.07%) in the polarity sentiment analysis task. The proposed model works on character-level what leading to language independence.

Paper [5] presents bidirectional hierarchical LSTMs (RNNs) used for sentiment analysis task. Presented neural networks model captures relationship between sentences in document [5]. Evaluation of 5 star sentiment analysis task has been done on the Yelp data set [8].

Approach based on the GNNs has been introduced in [4]. Text is converted to a graph, where each node of the graph denotes word in vector space. The nodes are connected to each other within a distance 2, i.e. node representing word "neural" in sentence "Graph neural networks are awesome." is connected to node "Graph", "networks" and "awesome". This leads to connecting unrelated nodes (words).

3 EXPERIMENT DESCRIPTION

This section provides detailed information of the introduced approach, description of the proposed graph convolutional neural networks models. Further, in this section is described building a graph structure from text and methodology how the models are evaluated. Achieved results are also presented.

3.1 DATA SET

The proposed models have been trained and tested on the Yelp data set [4] purposed for deep learning. It contains a millions of user's reviews divided into 5 classes. In the experiment only two classes are taken into account, i.e. positive and negative class. The proposed models are trained on 25 000 samples. The validation and test has been done on 5 000 samples.

3.2 BUILDING A GRAPH FROM TEXT

Graph neural networks operate over a graph $G = (V, E)$ in Non-Euklidean space. The $V = \{\mathbf{e}\}_{i=1:N^v}$ denotes set of graph's nodes, where $\mathbf{v}_i$ represents node's feature vector. Each node from V represents single word. Set $E = \{\mathbf{e}_i, v_{start}, v_{end}\}_{i=1:N^e}$ denotes graph's edges, where $\mathbf{e}_i$ is edge features vector. Each edge captures relationship between nodes (words).

Each node feature vector contains 2 information. The first one is representation of word in vector space. For this purpose has been used pre-trained word2vec Google News corpus word vector model [3]. It converts words into 300-dimension vector space. The second information is normalized POS tag. POS tagging is task related to NLP. Therefore, it has been necessarily to use any existing tool for POS tagging and relationship between words. In the experiment has been used spaCy library [6].

Graph edges indicate relationship between words. In the experiment are edges used only as information to create adjacency matrix. SpaCy library has been used for extracting information about words relationship (edges) from input text.

In the figure 1 is example of transformed sentence "Graph neural networks are awesome" to a graph. It can be seen relationships between each node (word) as well as POS tag.

Input text is limited to length of 60 words, more words are not taken into account. Built graphs must be converted into a form suitable as input to the proposed model. In this step it is necessary to create matrix containing feature vectors of each node and also adjacency matrix. Examples of both matrix can be seen in the figure 2.

For this experiment has been used library, facilitating experiments with a graph neural networks, based on Keras and TensorFlow 2 called Spektral [1].

3.3 PROPOSED NEURAL NETWORKS MODELS

In the experiment have been developed and trained three text classifier models based on GCNs. Core of the models is trio of GCNs as presented in work [7]. Kernel L1 and L2 regularization methods are used in GCNs, because models reported a huge over-fitting. The proposed models differ in number of filters of each GCNs layers. The next part of proposed architecture contains 4 dense layers. Between

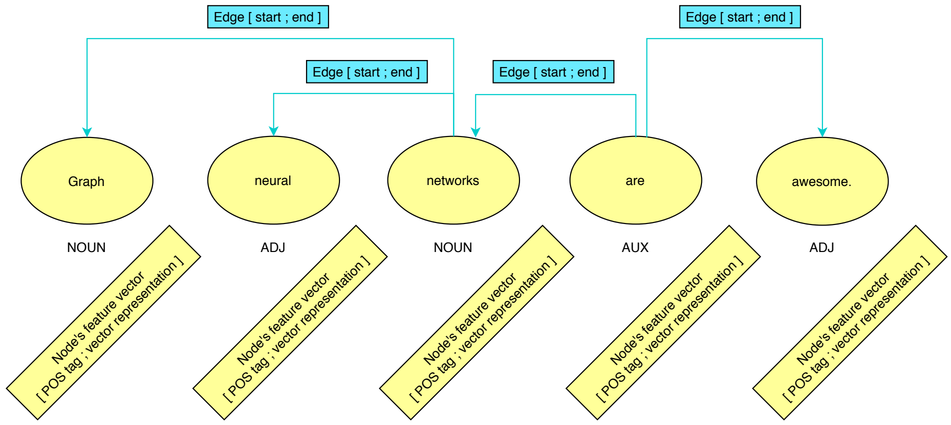

Figure 1: Graph representation of sentence "Graph neural networks are awesome.".

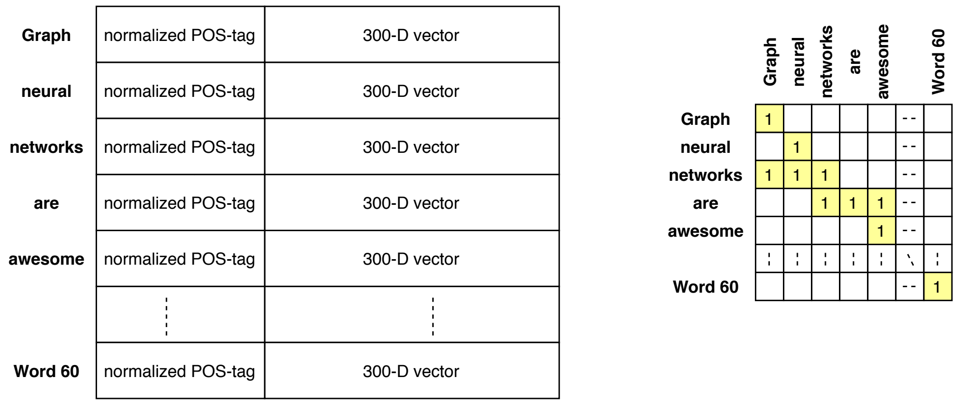

Figure 2: Node feature matrix and adjacency matrix of sentence "Graph neural networks are awesome".

them are inserted dropout layers to prevent model over-fitting. Last layer is dense with softmax activation function in order to predict input graph as positive or negative text. The architecture can be seen in the figure 3. Setting for the trained models can be seen in the table 1.

	GCNs			Dropout	1st Dense	Other Denses	Softmax
Model	Filters	Activation	Kernel reg.	Drop	Neurons	Neurons	Neurons
1	196	relu	L1 = L2 = 0.01	0.05	512	256	2
2	256	relu	L1 = L2 = 0.01	0.05	512	256	2
3	384	relu	L1 = L2 = 0.01	0.05	512	256	2

Table 1: Settings of the models used in the experiment.

4 RESULTS

Table 2 shows achieved results by trained models. All models have been trained on 12 500 data samples of each class in 150 iterations. The validation and test data set contains 5 000 data samples. Figure 4 shows dependency of loss function and classification accuracy on iterations during training process. This graph is related to the first model, see table 1. The peak of the loss function in the begin-

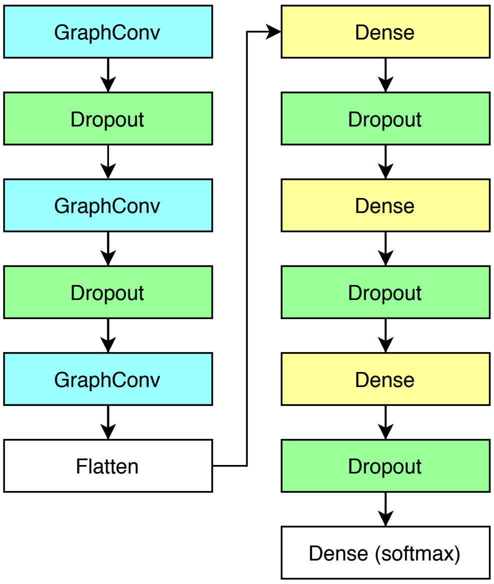

Figure 3: The proposed architecture of the text classifier based on the GCNs.

ning of the training process achieves up to 6. This is due to using the L1 and L2 kernel regularization methods to prevent model over-fitting.

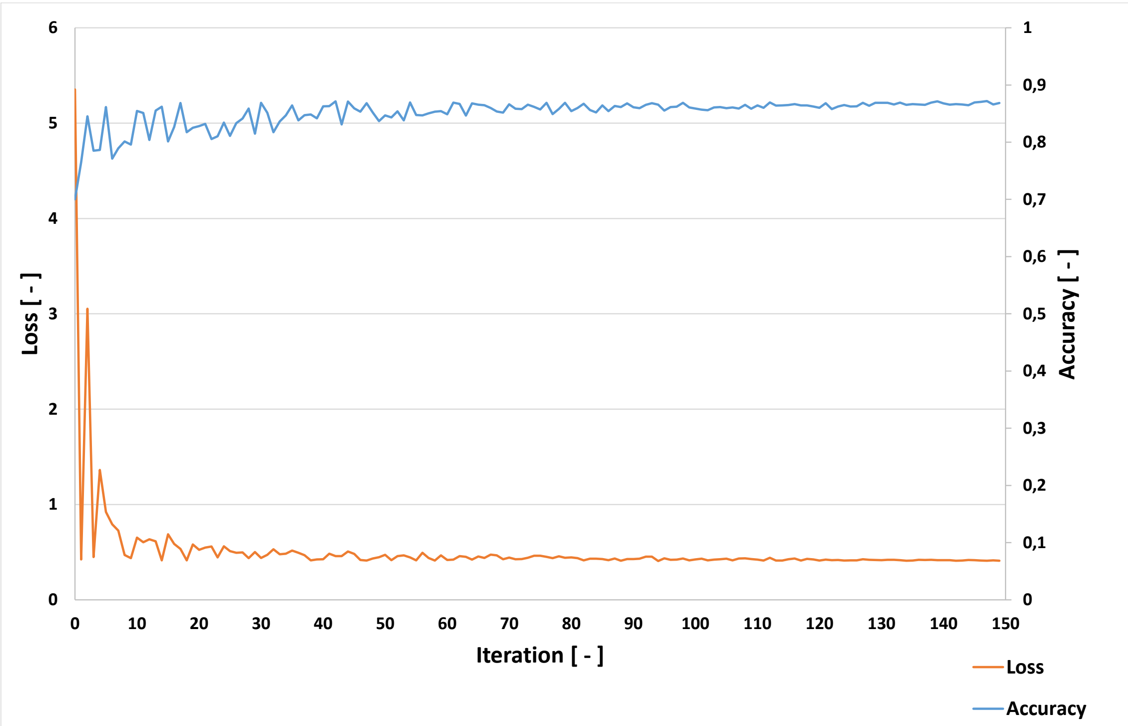

Figure 4: The proposed architecture of the text classifier based on the GCNs.

The first trained model classified test samples with 85.40% accuracy. Setting of this model can be seen in the table 1. The model number 2 where 256 GCNs filters have been used achieved 85.52% test accuracy. It is small change (0.12%) against the first model. It can be caused by initializing model's weights. The second model shows slight over-fitting, because training loss is lower than validation and test. Model 3 has more GCNs filters than previous model. It increased test accuracy in contrast with model 2 by 0.6%.

	English					
Model	Train acc.	Train loss	Valid. acc.	Valid. loss	Test acc.	Test loss
	%	-	%	-	%	-
1	87.3200	0.3843	86.6000	0.4125	85.4000	0.4283
2	88.6200	0.3661	85.0000	0.4558	85.5200	0.4324
3	89.2400	0.3618	85.9600	0.4373	86.1200	0.4337

Table 2: Achieved results in the experiment by different models.

5 CONCLUSION

This paper has proven that graph representation of text is very useful because it can capture a lot of valuable information of text structure. It can lead to increasing classification accuracy. Main disadvantage of presented approach is dependency on third party library sciPy.

The accuracy achieved by the first model is 85.40%. The second model contains more GCNs filters. It causes that this model achieved better results (85.52%). The last trained model contains the most GCNs filters (384). Therefore, the result of the last model (86.12%) has been improving accuracy in contrast of the other models in the experiment. The output of this experiment prove usefulness of converting text into the graph structure.

In the future, it is possible to improve the classification accuracy by optimizing the model architecture and extending the training data set.

REFERENCES

[1] Daniele Grattarola. *Spektral.* `https://spektral.graphneural.network/`

[2] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini. *The Graph Neural Network Model.* in IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61-80, Jan. 2009.

[3] Google Code Archive. *Long-term storage for Google Code Project Hosting.* `https://code.google.com/archive/p/word2vec/`

[4] HUANG, Lianzhe, et al. *Text Level Graph Neural Network for Text Classification.* arXiv preprint arXiv:1910.02356, 2019.

[5] SHI, Xiaoming; LU, Ran. *Attention-Based Bidirectional Hierarchical LSTM Networks for Text Semantic Classification.* In: 2019 10th International Conference on Information Technology in Medicine and Education (ITME). IEEE, 2019. p. 618-622.

[6] spaCy. *Industrial-strength Natural Language Processing in Python.* `https://spacy.io/`

[7] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks* arXiv preprint arXiv:1609.02907, 2016.

[8] Yelp. *Yelp Dataset.* `https://www.yelp.com/dataset`

[9] Zhang, Xiang a LeCun, Yann: *Text understanding from scratch. arXiv preprint arXiv:1502.01710, 2015.* Available: `https://arxiv.org/abs/1502.01710`.