Neural Networks as Explicit Word-Based Rules

Jindřich Libovický
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
libovicky@ufal.mff.cuni.cz

Abstract

Filters of convolutional networks used in computer vision are often visualized as image patches that maximize the response of the filter. We use the same approach to interpret weight matrices in simple architectures for natural language processing tasks. We interpret a convolutional network for sentiment classification as word-based rules. Using the rule, we recover the performance of the original model.

1 Introduction

When using convolutional neural networks (CNNs) for computer vision (CV), the convolutional filters can be visualized as small image patches that maximize the response to the filter (Krizhevsky and Hinton, 2009; Krizhevsky et al., 2012). Intuitively, the more similar a window of the image to the filter visualization, the higher the neuron activation is.

In natural language processing (NLP), discrete network inputs are first embedded into a continuous vector space. The projection that follows the embedding can be interpreted in a similar way as the filters in CV. We can retrieve the words whose embeddings have the highest response to the projection. In this abstract, we use this principle to reconstruct a CNN for sentence classification using explicit rules. We present a case study of this approach using models for sentiment analysis.

2 CNN for Sentiment Analysis

The goal of sentiment analysis is to decide if a snippet of text speaks positively or negatively about whatever its content is. We train and evaluate our models on the IMDB dataset (Maas et al., 2011) that contains 17k training, 7.5k validation and 25k test examples with a balanced number of positive and negative examples of movie reviews.

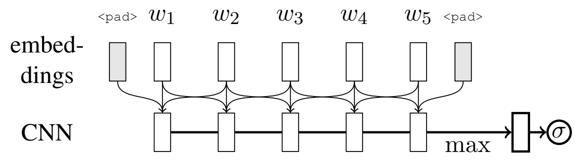

Figure 1: Architecture used for the sentiment analysis.

For our experiments, we use a convolutional network with max-pooling (Kim, 2014) depicted in Figure 1. We use word embeddings with dimension $d = 300$, kernel widths k from 1 up to 5 of $n = 500$ filters.

Formally, for a sequence of word embeddings x_i of dimension d, the output of the network is:

$$\mathbf{v} \cdot \underset{j=1..k}{\mathrm{Concat}} \left[\max_i \left(\mathrm{ReLU}(\mathbf{W}_j [x_{i-\frac{j}{2}}, \ldots x_{i+\frac{j}{2}}]) \right) \right]$$

where $\mathbf{W}_j \in \mathbb{R}^{d \times n}$ and $\mathbf{v} \in \mathbb{R}^{kn}$ are trainable parameters. We apply the sigmoid function over the output and train the network towards the cross-entropy loss.

We trained the models until convergence and analyzed the learned weights. Our best model reached 89% accuracy, the state-of-the-art result with pre-trained sentence representation is 95% (Howard and Ruder, 2018).

3 Model Interpretation

For each weight vector in each filter, we find words whose embeddings have the highest dot-product with the weight vector. We interpret filters of size 1 as sets of these words. We interpret kernels of sizes larger than 1 both either as conjunctions or disjunctions of the neighboring words. In the conjunction case, we interpret a filter as a set of n-grams which consists of all combinations of the words extracted from the weight vectors. In the

Width 1, weight $8 \cdot 10^{-3}$:

- 1 (8.345), pointless (7.664), incohrerent (7.270)

Width 1, weight $9 \cdot 10^{-3}$:

- perfect (12.186), brilliant (5.268), innocent (5.040)

Width 3, weight $-1 \cdot 10^{-3}$:

- yawn (7.549), incoherent (6.338), ludicrous (6.117)
- disappointing (4.312), acquit (4.241), appalled (4.233)
- heather (7.362), boredom (5.949), pasolini (5.109)

Table 1: Examples of words extracted from the convolutional filters.

k	CNN	Rules (&)	Rules ($\vee$)	Classifier
1	83.4	83.0	83.0	86.2
2	85.9	82.4	81.5	86.2
3	87.2	81.8	81.4	86.0
4	87.7	81.9	81.0	86.1
5	87.8	82.0	81.6	85.9

Table 2: Quantiative results of the sentiment classifier and its reconstruction for different kernel sizes k.

disjunction case, we interpret the filters as multiple independent filters of size 1. Examples of the words extracted from the filters are shown in Table 1.

We interpret the max-pooling over time as an existential quantifier and thus the whole sentence representation as asking for the presence of particular words or n-grams, i.e., as a set of binary features.

We conduct two experiments with extracted features. First, based on the weight vector $\mathbf{v}$, we sort the features as contributing to positive or negative sentiment and label the sentences with the prevailing class. Second, we train a linear classifier based on the binary features.

The quantitative results of the experiments are shown in Table 2. There is only a minor difference between interpreting the filters as conjunctions and disjunctions. This shows that the filters of width 1 are the most important ones and also that neither of our interpretation of the wider filters is entirely correct.

The experiments with the linear classifier show that when the filters are interpreted as simple feature extractors, the model performance can be fully recovered.

	Kernel size				
	1	2	3	4	5
ADJ	46.1	43.7	37.2	33.2	30.7
ADV	15.8	9.2	11.2	11.9	10.7
NOUN	14.6	15.8	19.1	20.5	24.6
VERB	11.2	16.6	19.4	20.1	19.0
PROPN	1.1	6.1	5.7	8.2	8.5
NUM	6.9	4.9	4.4	3.7	4.0
rest	4.4	3.8	2.9	2.6	2.6

Table 3: Percentage of POS tags in extracted words for different kernel sizes.

4 Filter Analysis

We analyze the part-of-speech of the extracted words. We computed the most frequent POS tag for each word based on English Web Treebank (Silveira et al., 2014).We then computed statistics of the most frequent POS tag for words extracted from the network filters.

The statistics are shown in Table 3. The most frequent POS tag among the extracted word is adjective. With increasing network capacity, the model becomes sensitive to nouns and proper nouns. The proportion of function words decreases with the increasing kernel size which suggests that it is unlikely that the filters of large kernel size would capture more complex phrases.

We also compared the words extracted from the filters using Opinion Lexicon (Hu and Liu, 2004) containing 4.8k words contributing to negative and 2.0k contributing to positive sentiment. Regardless of the model, approximately 60 % of the extracted words appear in the lexicon. If we label the words the sign of the corresponding weight from vector $\mathbf{v}$, we get precision over 99 % for both words contributing to the negative and positive sentiment with respect to the lexicon.

5 Conclusions & Future Work

We showed that the first layer of a CNN for sentiment analysis can be interpreted as responding to particular words on input. Using these rules, we fully reconstruct a model for sentiment classification.

As future work, we would like to extend this approach for more complex architectures and other NLP tasks.

Acknowledgements

This work has been supported by the grant 18-02196S of the Czech Science Foundation.

References

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Alex Krizhevsky and Geoffrey Hinton. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, 6, pages 1097–1105, Red Hook, NY, USA. Curran Associates, Inc.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.