Distilling knowledge from BERT into Simpler Machine Learning Models

Praveen Sridhar, Rakesh Varma Siri, Rashmi Nagpal, Ribhu Lahiri

Plaksha TechLeaders Fellowship, Gurugram, India

AI at Scale Project

Submitted on 21st January 2020

Abstract

As a core task of natural language processing and information retrieval, performance of algorithms plays a vital role. This survey paper draws comparison between various distillation models, which generates predictions from the whole ensemble of models less computationally expensive. We propose a simple compression pipeline which achieves considerable amount of reduction in model size without distorting accuracy. We also explore methods of interpretability of complex models as a future line of work.

Keywords: Knowledge Distillation, BERT, Model Compression

I. Introduction

Recent advancements in neural networks like BERT and GPT-2 have led to a rise in the field of NLP in the past few years. Identifying bias from advanced neural neural networks has become harder as the models are not easily interpretable. By distilling neural networks and using them as backbone, with interpretable Machine Learning models as head, we attempted to achieve interpretability.

For this experiment we considered sentiment analysis on IMDB movie review dataset as a binary classification problem. We explored by providing knowledge distilled from BERT embeddings as input for more familiar interpretable machine learning models of Logistic Regression, Decision trees and SVM. Once the accuracy is similar to state-of-art, those models are applied with LIME to interpret.

The rest of the paper is organized as follows. Section II comprises related work. In Section III the proposed approach is presented in detail. Further, we showed the results and analysis in Section IV. Section V comprises a conclusion as well as future work.

II. Related Work

Distilling Task-Specific Knowledge from BERT into Simple Neural Networks [1] explains how deep language representation model, which includes BERT, ELMo, and GPT are distilled for knowledge into a single-layer BiLSTM while using roughly 100 times fewer parameters and 15 times less inference time. In the past, researchers have developed and applied various neural architectures for NLP, including convolutional neural networks [2], recurrent neural networks [3], and recursive neural networks [4]. These generic architectures can be applied to tasks like sentence classification [5] and sentence matching [6] but the model is trained only on data of a particular task.

Recently [7] introduce Embeddings from Language Models (ELMo), an approach for learning high-quality, deep contextualized representations using bidirectional language models. With ELMo, they achieve large improve- ments on six different NLP tasks [8] propose Bidirectional Encoder Representations from Transformers (BERT), a new language representation model that obtains state-of-the-art results on eleven natural language processing tasks. Trained with massive corpora for language modeling, BERT has strong syntactic ability and captures generic language features. A typical downstream use of BERT is to fine-tune it for the NLP task at hand. This improves training efficiency, but for infer- ence efficiency, these models are still considerably slower than traditional neural networks.

Model compression : A prominent line of work is devoted to compressing large neural networks to accelerate inference. Early pioneering works include [9], who propose a local error-based method for pruning unimportant weights. Recently, [10] proposed a simple compression pipeline, achieving 40 times reduction in model size without hurting accuracy. Unfortunately, these techniques induce irregular weight sparsity, which precludes highly optimized computation routines. Some studies examine quantization neural networks [13]; in the extreme [14] propose binarized networks with both binary weights and binary activations.

Unlike the aforementioned methods, the knowledge distillation approach [12] enables the transfer of knowledge from a large model to a smaller, "student" network, which is improved in the process. The student network can use a completely different architecture, since distillation works at the out- put level. This is important in our case, since our research objective is to study the representation power of shallower neural networks for language understanding and thereby simultaneously compressing models like BERT thus, we follow this approach in our work.

III. Approach

Our approach involved in first fine tuning the pre-trained BERT model for Binary classification task. This fine-tuned model is then used as the teacher model for distillation into simpler models.

We chose 3 baseline Machine Learning models - Logistic Regression, Decision Trees and Support Vector Machines for the binary classification task of sentiment analysis. After analysing the baseline performance of these algorithms using the scikit-learn library, we created new distilled models, where the labels are the output of the BERT fine tuned model.

Model Architecture

```
class BertBinaryClassifier(nn.Module):
    def __init__(self, dropout=0.1):
        super(BertBinaryClassifier, self).__init__()

        self.bert = BertModel.from_pretrained('bert-base-uncased')
        self.linear = nn.Linear(768, 1)

    def forward(self, tokens, masks=None):
        _, pooled_output = self.bert(tokens, attention_mask=masks)
        linear_output = self.linear(pooled_output)
        return linear_output
```

The architecture for BERT finetuning is a simple linear layer stacked after the pooled output of the pretrained BERT layer. This has been achieved using the PyTorch framework using Transformers library.

Distillation Architecture

Scikit-learn does not provide interfaces for distillation or label smoothed inputs. Hence we use the equivalent regression models for each algorithm, predict the logits and then use a custom sigmoid function to obtain the final classification.

Experimental Setup

Datasets

1. IMDB Movie reviews sentiment dataset - A sentiment analysis dataset on movie reviews. This was the main dataset we performed our experiments on. Essentially the task is a binary classification problem to label incoming text as having Positive or Negative sentiment.
2. SNLI Dataset : The Stanford Natural Language Inference dataset. We tried experiments on the SNLI dataset, but were unable to produce proper results. It was challenging to perform distillation on this task and hence we have kept it as a future task.

Baseline Models

1. Logistic Regression
2. Decision Trees
3. Support Vector Machines

Interpretability

We use open source implementation of LIME [15] for visual interpretations of baseline models

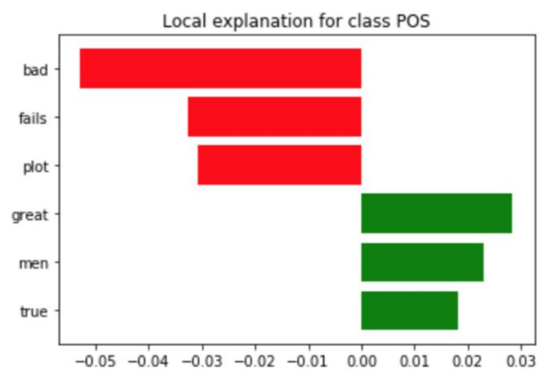
IV. Results and Analysis

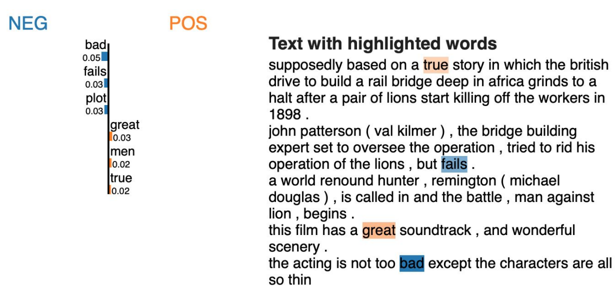
From the experiments, we can see that the distilled models are not performing better than the baseline models. We suspect that this is due to the size of the dataset. Additional work is being conducted to use unlabelled dataset into the pipeline by getting the BERT predictions and using that as additional data for distillation.

Model	BERT	Logistic	Decision Trees	SVM
Baseline	0.93	0.89	0.63	0.88
Distilled	-	0.87	0.65	0.70

Table 1. Experimental Results - Accuracy

Visual Interpretations using LIME





V. Conclusion

In this survey paper, we explored distilling the knowledge from BERT into simpler models like Logistic Regression, Decision Trees and SVM. Those distilled models achieved comparable results with BERT fine-tuned classification models. Our results suggest that Logistic Regression is more expressive for natural language tasks than previously thought.

The code and experiments can be found on Colab

One direction of future work is to explore extremely simple architectures in the extreme, and interpret those models using LIME and Shapely frameworks. Another direction is to explore slightly more complicated architectures using tricks like pairwise word interaction and attention.

References

[1] Tang, Raphael & Lu, Yao & Liu, Linqing & Mou, Lili & Vechtomova, Olga & Lin, Jimmy. (2019). Distilling Task-Specific Knowledge from BERT into Simple Neural Networks

[2] 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

[3] Tomas Marin Karafia’t Luka Burget Jan Cˇernocky`,and Sanjeev Khudanpur.2010.Recurrent neural network based language model. In Eleventh annual conference of the international speech com- munication association

[4] Richard Socher, Cliff C. Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natu- ral language with recursive neural networks. In Pro- ceedings of the 28th international conference on ma- chine learning (ICML-11), pages 129–136

[5] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text clas- sification. In Advances in neural information pro- cessing systems, pages 649–657

[6] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A deep ar- chitecture for semantic matching with multiple po- sitional sentence representations. In Thirtieth AAAI Conference on Artificial Intelligence

[6] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A deep ar- chitecture for semantic matching with multiple po- sitional sentence representations. In Thirtieth AAAI Conference on Artificial Intelligence

[7] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word repre- sentations. In Proceedings of the

2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), volume 1, pages 2227–2237

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language under- standing. arXiv:1810.04805

[9] Alexis Conneau, Holger Schwenk, Lo ¨ıc Barrault, and Yann Lecun. 2016. Very deep convolutional net- works for text classification. arXiv:1606.01781

[10] Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural net- works with pruning, trained quantization and Huffman coding. arXiv:1510.00149

[11] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning efficient convolutional networks through network slimming. In Proceedings of the IEEE In- ternational Conference on Computer Vision, pages 2736–2744

[12] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In Advances in neural information processing systems, pages 2654–2662

[13] Shuang Wu, Guoqi Li, Feng Chen, and Luping Shi. 2018. Training and inference with integers in deep neural networks. In International Conference on Learning Representations

[14] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. arXiv:1602.02830

[15] Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin, Why Should {I} Trust You?": Explaining the Predictions of Any Classifier, SIGKDD 2016