Creating a very small BERT pretrained model Main Quest 04

Anonymous authors
Paper under double-blind review

Abstract

The abstract paragraph should be indented 1/2 inch on both left and right-hand margins. Use 10 point type, with a vertical spacing of 11 points. The word **Abstract** must be centered, in bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

The field of Natural Language Processing (NLP) has undergone significant transformation with the introduction of the Transformer architecture by Google in 2017, which subsequently enabled the advent of transfer learning. Transfer learning typically consists of two phases: pre-training and fine-tuning. In the pre-training phase, a model is trained on a large corpus to gain a general understanding of language, and this pre-trained model is then fine-tuned for specific NLP tasks. This approach has yielded astonishing improvements in performance across a variety of NLP tasks. Despite the significant advancements in this area, there is an increasing need to understand these models better and make them computationally efficient. The research presented here aims to contribute to this discourse by focusing on BERT (Bidirectional Encoder Representations from Transformers), a groundbreaking model that has shown remarkable success in capturing the intricate semantics of language.

The contributions of this paper are twofold. Firstly, it aims to enhance the understanding of BERT's inner workings by constructing a very small version of the pretrained model. Secondly, it evaluates the capabilities and limitations of this small-scale BERT model in capturing complex linguistic features. The paper is organized into sections that cover background information on unidirectional drawbacks of models like ELMo and GPT, the methodology involving BERT's architecture and pretraining phases, and a conclusion highlighting the merits and demerits of BERT, with a particular focus on the challenges of fine-tuning.

2 Background

2.1 Transformer and Transfer Learning in NLP

The introduction of the Transformer architecture enabled more effective techniques for pre-training language models, thereby paving the way for transfer learning in NLP. Notable models that utilize the Transformer architecture for transfer learning include GPT, which is trained with a Left-To-Right (LTR) structure, and ELMo, which emphasizes contextual features.

2.2 Feature-Based vs Fine-Tuning Approaches

There are two primary techniques for leveraging pre-trained language models: feature-based learning and fine-tuning. ELMo exemplifies feature-based learning by employing two LSTM language models trained in LTR and Right-To-Left (RTL) directions. The hidden states from each layer are combined to offer a contextual representation of words in a sentence. This approach addresses a significant limitation in traditional word2vec embeddings, which do not consider the contextual meaning of words. In contrast, GPT follows a fine-tuning approach by adjusting all the pre-trained model parameters for specific downstream tasks.

2.3 Unidirectional Language Models and Their Limitations

BERT has highlighted a crucial drawback with unidirectional models, particularly those built on an LTR structure, like GPT. Such models often fail to capture context effectively, especially in tasks requiring bidirectional understanding, such as Question Answering tasks. BERT addresses this limitation by introducing a new Masked Language Model, which aims to capture richer contextual information during the pre-training phase. While ELMo also attempts to incorporate bidirectional information, it does so by simply combining the outputs of individually trained LTR and RTL models. This approach falls short in capturing deeper contextual relationships among words in a sentence.

3 Method

In the "Method" section, we will delve into the architecture of BERT, focusing on its unique pretraining techniques that allow it to capture bidirectional context more effectively than its predecessors. BERT is trained in two distinct phases: Pre-training and Fine-tuning. In the Pre-training phase, the model performs two types of tasks on unlabeled data. Then, in the Fine-tuning phase, the pre-trained parameters are loaded and updated to fit a specific labeled task. Therefore, even though the same pre-trained model is used, it undergoes fine-tuning to adapt to the problem at hand, resulting in a specialized model.

3.1 Pre-training

3.1.1 Masked Language Model

MLM (Masked Language Model) involves randomly altering some of the input tokens during training, which the model then learns to predict. During this learning process, the final hidden vector corresponding to the [MASK] token is fed into the final Softmax layer that maps to the model's vocabulary to produce the output. This essentially follows the same approach used for training traditional language models.

In BERT, 15% of the tokens in the entire input are selected to be masked. Of these, 80% are converted to the original MASK tokens, 10% are changed to random tokens that are not the original word, and the remaining 10% are left unchanged. The model then uses the final hidden state vector for each word to predict the original tokens. This only accounts for 1.5% of the entire dataset, so it is believed that this doesn't significantly impact the model's overall understanding of the language. Also, adopting this approach means that the Transformer encoder is forced to maintain contextual information for the entire input sequence, as it doesn't know which words it will need to predict.

3.2 Next Sentence Prediction (NSP)

Next, let's look into Next Sentence Prediction (NSP). Among various natural language problems, there are tasks like Question Answering (QA) and Natural Language Inference (NLI) that require understanding the relationship between sentences, something that general language models may struggle with. To train on such relational aspects between sentences, an additional binary classification task called Next Sentence Prediction was introduced. Sentences A and B are selected from the pre-training data, with 50% actually being consecutive sentences (IsNext) and the other 50% being completely unrelated sentences (NotNext). The task is straightforward: simply predict whether the two sentences are consecutive or not. Despite its simplicity, this approach proves to be very effective for tasks like QA and NLI.

3.3 three types of embeddings

To achieve high performance in these two tasks, namely MLM (Masked Language Modeling) and NSP (Next Sentence Prediction), BERT utilizes three types of embeddings. These embeddings assist the model in understanding how words within a sentence or between sentences relate to each other, as well as the significance of the position of each word in the sequence.

3.4 Fine-tuning

4 Result

5 References

References

A Appendix

You may include other additional sections here.