Review on BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

CS410 Technology Review, Fall 2021
Hui Huang NetId: huih3

I. Introduction

The paper 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding' introduces a new language representation model called BERT. The two steps of the framework include pre-training and fine-tuning. In the pre-training stage, a bidirectional attention-based transformer is trained by training with two unsupervised tasks, Masked Language Modeling and Next Sentence Prediction respectively. In the fine-tuning stage, the pre-trained BERT model can be fine-tuned by adding an additional output layer for various downstream tasks without substantial architecture modification. The BERT model achieves new state-of-the-art results on the eleven popular natural language processing tasks.

II. Comparison with OpenAI GPT and ELMO

Firstly, the paper demonstrates the advantage of pre-training a deep bidirectional representation by comparing the BERT model with the current techniques like OpenAI GPT and ELMO.

OpenAI GPT is a left-to-right attention-based transformer, it goes step-by-step and it uses attention in each step, it also has multiple layers. However, each intermediate step of the attention can only attend to whatever is to the left of it. Whenever you want to interpret a particular token, the context is only to the left of the token and you lose the information coming in. The authors argue that the unidirectional trait limits the architecture than can be used during pre-training and is the basic limitation of OpenAI GPT.

ELMO is a novel way for word embeddings. It uses 2 LSTM, one from left-to-right and one from right-to-left, and then concatenate the hidden states from both directions to get the final embeddings for each token. The paper argues that ELMO is still shallow as a simple concatenation of the left facing LSTM and the right facing LSTM intrinsically have nothing to do with each other. Ultimately a single model that can look at both the left and the right at the same time and then incorporate information from both sides simultaneously is wanted.

III. BERT Architecture

Here comes the BERT, in each layer of which, for a particular token, the model looks at every token in the input, basically all of the context. The transformer architecture in each layer of the BERT model is still attention-based, but it not only attends to the left but also attend to the right.

- **Pre-training stage: Masked Language Model task**

The logic for other models not to use bidirectional training is that in some natural language processing tasks like language generation, it's by the definition of the task that the model is supposed to only be able to look to the left. BERT deals with it by trained using Masked Language Modeling. In a masked language modeling, some words are replaced by the masked token and model is asked to predict the masked tokens. The model simply drops out some of the words some of the time.

- **Pre-training stage: Next Sentence Prediction task**

In addition to that, in order to understand the relationships that span multiple sentences, the BERT model is pre-trained on a second task at the same time where two sentences as an input are fed, the model is asked to predict a label 'IsNext' or 'NotNext'. 'IsNext' is true when the second sentence follows the first sentence. It's trained using an unsupervised approach by taking a big corpus and for 50% of the time, two continuous sentences are taken and the label is true, and for the rest 50% of the time, a random sentence is chosen as the second sentence.

The input to the BERT model is a sum of the token embeddings, the segmentation embedding and the position embeddings. A segmentation embedding is a binary label with EA being the label for the first sentence and EB being the label for the second sentence to help the model to differentiate the two sentences. The position embedding helps the model to decide if two tokens are close to each other in input or if they are actually far apart.

- **Fine-tuning stage: MNLI task**

The paper then demonstrates another advantage of BERT as being uniformly architected across different tasks. In the fine-tuning stage of the model, the model is initialized with the pre-trained parameters, then all the parameters are fine-tuned using labeled data from the downstream tasks. There's minimal difference between the pre-trained architecture and the final downstream architecture, which reduces the need for many heavily-engineered task-specific architectures. The model is evaluated on 11 different NLP tasks with the same model and the pre-trained model is fine-tuned to do all these tasks.

For example, in the MNLI task, given a pair of sentences, the model is about to predict whether the second sentence if an entailment, contradiction, or neutral with respect to the first one. What the model does is simply take the final embedding for the first one corresponding to the start token and then put a single layer of classification on it. The model only needs a matrix with size of D * 3, with D represents the dimension of the first embedding. The weight matrix is then trained altogether with the pre-trained BERT. However, the model simply needs to learn the new weights from scratch. It's very neat and by simply adding a layer on top of the pre-trained BERT, the BERT is allowed to model a downstream task.

- **Fine-tuning stage: SQuAD task**

In a question-answering task like SQuAD, we have an input question and an input paragraph which is a paragraph from Wikipedia page with answer embedded in it. Usually with this kind of task, the model goal is to predict the span, the start of the answer and the end of the answer. It can also be easily adjusted with the pre-trained BERT. We pre-trained the model with the first sentence as the question, the second sentence as the paragraph containing the answer. For each token in the output, the model adds additional layer to classify if it's the start token, the end token or none of them.

IV. Evaluation

The paper lastly demonstrates the importance of the two pre-training tasks MLM and NSP by training on 5 different tasks and compare the performance between models with the same pre-training data, fine-tuning scheme and hyperparameters but with and without MLM and NSP, it also explores the effect of model size and finds that large model size also benefits small scale tasks.

V. Conclusion

The BERT model, which takes into account both the left and right context of given word when doing the attention during pre-training, can be easily adjusted to tackle a broad set of NLP tasks during fine-tuning for downstream tasks.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova.
2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
arXiv:1810.04805