CS475 Spring 2021 Homework 3

Team 0: Dongkwan Kim, Jiseon Kim, and Alice Oh

April 15, 2021

BERT [1] has become a standard architecture for NLP research ever since it was published. BERT computes the representation for every token, but for sentence-level tasks (e.g., sentiment analysis), we usually use the output representation of the special token `[CLS]`. Is this the best way to pool token-level representations to create the sentence-level representation?

In this homework, your team will implement and compare different techniques for pooling token-level representation in BERT, namely `BERTPooler` in `huggingface` library (`https://huggingface.co/`).

1 Tasks

One simple pooling method is to apply permutation-invariant operations (e.g., mean, max, sum) across all token representations. We define the output representation of `MeanMaxTokens` (shortly MMT) as:

$$C_{\text{MMT}} = \left[\sum_{i=0}^{L-1} T_i \;||\; \max_i T_i\right], \quad C_{\text{MMT}} \in \mathbb{R}^{2H}, \quad T_i \in \mathbb{R}^H, \tag{1}$$

where L is the maximum length of sequence and $||$ is the vector concatenation operation. Note that we borrow notations from the original BERT paper [1].

The `BERTPooler` in `huggingface` implementation applies linear transformation with $W \in \mathbb{R}^{H\times H}$ and tanh activation (See `BertPooler` class). Similarly, we apply linear transformation with $W_{\text{MMT}} \in \mathbb{R}^{H\times 2H}$ and tanh activation.

$$C = \tanh(C_{\text{MMT}} W_{\text{MMT}}^\top),\; C \in \mathbb{R}^H \tag{2}$$

The tasks in this homework are as follows:

1. Implement the `MeanMaxTokens` pooler (See `MeanMaxTokensBertPooler` class in `bert_poolers.py`).
2. Implement your own BERT pooler (See `MyBertPooler` class) and describe its architecture and rationale in your report. It does not have to be completely novel.
3. Choose one dataset in GLUE [2], and compare the test performance of three poolers (See `run_glue.py`).
4. Discuss the result. Negative results are fine, the point is how you interpret and explain it.

2 Submission

The files you should submit are

1. Your team's `bert_poolers_{team_no}.py` (e.g, `bert_poolers_0.py`).
2. Your team's two-page `report_{team_no}.pdf` (e.g., `report_0.pdf`). Use this LaTeX file as a template, and do not change style attributes in this file. References are not included in the page-limit.

3 Grading

Comprehensive evaluation based on clarity, validity, and interestingness. You will get zero points if you violate academic integrity (e.g., plagiarism and data manipulation).

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[2] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.