

CSC594 Advanced Deep Learning: Assignment 3

Clayton Cohn

October 30, 2020

1 Assignment Specifications

This assignment involved finding source code from our chosen topics (BERT, in my case), annotating it, and implementing it. I chose to annotate `run_pretraining.py`, which is the code that Google provided for pretraining BERT's contextual word embeddings. I have included the annotated code in the enclosed Jupyter notebook. Also included is my implementation of `run_pretraining.py`. Please note that in order to run `run_pretraining.py`, I first had to filter the data through `create_pretraining_data.py`, and that file was not annotated or included in my submission.

2 Code Annotations

`run_pretraining.py` is the file that is responsible for conducting BERT's pretraining. It takes a corpus of sentence-separated documents (formatted via `create_pretraining_data.py`), and it performs both Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) on the corpus in order to train the model. The code initially generates a model via a `model_fn_builder` function, and then performs training on each batch via an `input_fn_builder` function. All of the pertinent information (weights, biases, masks, loss, etc.) is passed to the TPU via closures.

The authors of this code are the same as the BERT authors (although the code itself was uploaded by the first author): Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. It was published October 31, 2018, roughly three weeks after the publication of the original BERT paper to arXiv. The file is in Python and is 493 lines (including spacing and commenting). Please note that the provided code is *not* the code used by Google in the original paper (the original training was done via C++, according to their [GitHub repository](#)).

3 Reflections

Code annotation was an interesting idea for an assignment, and it was also something that I had not done before. Going through the code gave me a very good understanding of how it actually worked. Additionally, I gained a lot of insight into training with TensorFlow and TPUs. I have never used TPUs before, so it was interesting to see how everything was passed back and forth between the TPUEstimator and also how the TensorFlow flags were used for parameterization. What was also interesting was that both the model and the batches were passed to the TPU via closures. I am curious as to whether or not this is standard practice for TPU implementation or if it was just done for efficiency's sake.

I thought the implementation component was the most interesting part of the assignment. Personally, I tend to learn best by doing, so this was a great opportunity to see the code in action. I was able to implement the file correctly, but I did not conduct pretraining on a corpus large enough to meaningfully affect BERT's vectors (I used the Coral Bleaching dataset gathered by the NIU students that consists of about 8,000 sentences). The only difficulty I encountered was related to syntactical discrepancies, as the code provided was designed for TensorFlow 1 and Colab now defaults to TensorFlow 2. However, once I specified that TensorFlow 1 should be used, I stopped having problems. Altogether, I thought this was a very informative, enjoyable assignment.