Efficient Scaling and Pre-Training of Language Models with Electra and Reformers

Keshav Bhandari, Komal Thakkar, Aleksandr Simonyan, Dimitrios Mavrofridis

Abstract

The transformer architecture has been central to a multitude of recent research work in natural language processing. Yet, scaling the network suffers from huge memory related drawbacks and doesn't come without significant computational resources or budgetary support. In this paper, we repurpose a highly efficient Reformer encoder architecture to serve as the foundational blocks for the Electra pre-training methodology, thereby allowing the network to scale to 8 times the size of its transformer counterpart while maintaining the same memory requirements. The subsequent downstream performance of this scaled up architecture is at par with the transformer based Electra benchmark, while being pre-trained using only a third of the data.

1. Introduction

The last 5 years in the field of natural language processing, has seen tremendous progress with many novel language modeling techniques built around the Transformers [6] architecture beating previous state of the art benchmarks. Bagging the top most ranks for these benchmarks are models with hundreds of millions of trainable parameters that have been fed massive volumes of data. Yet, we have also seen modeling techniques or training methodologies that focus on cutting down trainable parameters, computational memory, time or even dataset requirements while still being able to deliver comparable results or in some cases even better than these "monster models".

A prime example of this is seen with the Electra pre-training approach [1], which beats the performance of GPT with around 1/30x compute and performs comparably to RoBERTa [3] and XLNet [4] while using less than 1/4 of their compute. Similarly, Kitaev et. al [5] were able to improve upon the Transformers dot-product attention bottleneck by replacing it with an attention mechanism that uses locality-sensitive hashing. This allowed them to reduce the attention complexity from quadratic to log-linear scale for their Reformer model [9] and hence scale to longer sequence lengths.

While both the Electra and Reformer studies are revolutionary in their objective to drive down the complexity of language models, they are not used to their full potential. For example, the Electra model uses the vanilla transformer with dot-product attention which hinders its ability to scale to longer sequences due to the quadratic computational complexity. Similarly, the Reformer

model, while having the ability to scale to longer tokens by means of a log-linear attention complexity, is not pre-trained by the authors like a typical language model.

In this paper, we combine both the methods - ELECTRA and Reformer to experiment pretraining a highly efficient transformer architecture and then fine tuning it on a downstream GLUE task to check its efficacy against the ELECTRA transformer baseline. Specifically, we design our experiments by scaling both model architectures to maximize the memory and computational resources available to us. This yields an architecture that is 8 times the size of the small Electra baseline with similar computation and memory requirements.

2. Datasets

We used OpenWebText [2] corpus for pre-training. It is an open-source recreation of the WebText corpus. The text is web content extracted from URLs shared on Reddit with at least three upvotes. These links were deduplicated, filtered to exclude non-html content and then shuffled randomly by the creators of the dataset. The near-duplicate documents were identified using local-sensitivity hashing. Documents were hashed into sets of 5-grams and all documents that had a similarity threshold of greater than 0.5 were removed. The remaining documents were tokenized, and documents with fewer than 128 tokens were removed. It is 38GB of text data from 8,013,769 documents.

For fine tuning, we used MRPC (Microsoft Research Paraphrase Corpus). The MRPC corpus consists of 5,801 sentence pairs which have been extracted from news sources on the web, along with human annotations indicating whether each pair captures a paraphrase/semantic equivalence relationship.

2.1. Pre-Training

For the pretraining part of our experiment, we used ten percent of the open web text data in order to pre-train our model. We ran the small Electra model that uses the vanilla Transformer architecture as the baseline. This network consisted of around 18M trainable parameters comprising 12 encoder blocks with 128 embedding size and 30522 vocabulary size as the hyperparameters. The matching discriminator architecture differs in size from the generator through a hidden size (dimensionality of the encoder layers and the pooler layer) of 256 and intermediate size (feed-forward layer in the Transformer encoder) which are both 4 times larger. Note, that scaling to a larger version wouldn't have been possible given our GPU memory capacity. However, the computational and memory bottlenecks of the Electra Transformer model did not apply to our Electra Reformer model as we were able to scale the latter to 8 times the size of the former, resulting in a total of 152M trainable parameters. Specifically, the majority of the differences in parameters between the two models lies in the following aspects:

1. The feedforward intermediate dimensions of the Electra Reformer model is scaled twice in the generator and 4 times in the discriminator as the original Electra Transformer model.
2. The number of attention heads for the generator and discriminator encoders in Electra Reformer is 4 times that of Electra Transformer.
3. The dimensionality of the attention weight matrices is also scaled up by 4X between the two models.

While the model was scaled up 8 times, the memory and computations do not scale linearly and we were well under our total GPU RAM capacity. We trained Electra Transformer for 30K steps and Electra Reformer for 10K steps on ~13 GB P100 GPU using a batch size of 128 for both. By comparison, Google's Electra-Small and Electra-Large are trained for 1M and 400K steps respectively as per the paper. The Electra Reformer model was trained for fewer steps than its competing benchmark as it was slower to train due to the extra dimensions and trainable parameters. However, the total training time and memory requirements for the two was similar with the major difference being that Electra Reformer was 8 times larger than the other as a result of the locality-sensitive-hashing and reversible residual layers in the Reformer based generator and discriminator implementation.

2.2. Fine-Tuning

For Fine Tuning, the GLUE dataset [7] was downloaded. We fine tuned our models on the MRPC (Microsoft Research Paraphrase Corpus) sub-task of the GLUE benchmark. The MRPC captures a paraphrase/semantic equivalence relationship. So MRPC sub-task is like a binary classification task which categorizes each sentence pair as either being semantically equivalent or not. To perform this task, the pretrained Electra Transformer and Electra Reformer models were loaded from the saved 30K and 10K checkpoint step respectively through the AutoConfig.from_pretrained Transformer function with the number of labels set as 2. This adds 2 neurons to the classification head of the discriminator architecture. We experimented with different hyperparameters during the fine tuning stage and finally settled on a learning rate of 5e-06 with no weight decay or warm up steps needed. The choice of optimizer (AdamW) and learning rate scheduler (linear) was the same as the pre-training stage.

3. Results

We used the small Electra Transformer model that was trained for 30,000 steps as the baseline to compare against our Electra Reformer model which was trained for 10,000 steps. We do not compare the two models during the pre-training phase. However, for the fine-tuning MRPC Glue task, we use the F1 score and binary accuracy as evaluation metrics. Figure 1 shows our fine tuned results against the Transformer Electra baseline along with some other benchmarks on the leaderboard just for reference. The other models shown here have been trained on significantly larger datasets and additional steps than that of ours.

Based on our experiments, we see that the Electra Reformer model when trained for only a third of the total steps as the Electra Transformer model comes in significantly close to the performance of the latter for the MRPC fine tuning task with the difference in F1 score between the two at only 0.85%.

Model Name	Accuracy	F1 Score
Vega v1 (1st place on leaderboard currently)	92.6	94.5
Google Electra-Large + Standard Tricks (400K steps)	90.7	93.1
Electra Transformer (30K steps)	**71.5**	**82.0**
Electra Reformer (10K steps)	**69.9**	**81.3**
Single Task BiLSTM + ELMo + Attn	68.8	80.2
Single Task BiLSTM + ELMo	69.0	80.8

Figure 1. MRPC task leaderboard scores evaluated based on binary accuracy and F1 score.

4. Analysis

The notable achievement of this experiment comes through the use of fewer steps and less data needed to achieve comparable results while maintaining similar GPU RAM usage. This feat is accomplished by means of scaling up the Electra transformer neural network by 8 times through the use of a highly efficient Reformer architecture. We attribute this comparable performance of Electra Reformer to the presence of extra dimensions in its encoder feedforward representations and attention weight matrices along with the addition of extra attention heads. Scaling up the language model is made possible through advancements made in the Reformer paper that solve the problems of attention and memory allocation through locality-sensitive-hashing to reduce the complexity of attending over long sequences, and reversible residual layers to more efficiently use the memory available.

5. Conclusion

In this paper, we are able to mitigate the memory and computational bottlenecks of transformer based language models by repurposing the Reformer encoder architecture as underlying foundational blocks for the Electra pre-training framework. Through our experiments, we demonstrated that the Electra Reformer architecture can be scaled up to match the performance of the Electra Transformer architecture through significantly fewer training iterations. The results of our experiments are an important step towards the development of more efficient pre-training approaches with limited amounts of data to train upon. Future tasks that deal with few shot learning techniques could employ the Electra Reformer pre-training strategy that combines the power of the two to yield the best possible downstream performance. Future work related to this study could involve complete training by way of more training data and iterations along with an extensive experimentation of downstream tasks to evaluate the performance of the model.

6. Acknowledgments

We would like to thank professor David Demeter for his valuable inputs and guidance, without which this paper would not have been possible.

7. References

1. Clark, K., Luong, M., Le, Q.V., & Manning, C.D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. ArXiv, abs/2003.10555.
2. Gokaslan, A., & Cohen, V. (2019). OpenWebText Corpus.
3. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv, abs/1907.11692.
4. Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., & Le, Q.V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. NeurIPS.
5. Kitaev, N., Kaiser, L., & Levskaya, A. (2020). Reformer: The Efficient Transformer. ArXiv, abs/2001.04451.
6. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. CoRR, 2017. URL http://arxiv.org/abs/1706.03762.
7. Warstadt, A., Singh, A., & Bowman, S. R. (2018). Neural Network Acceptability Judgments. arXiv preprint arXiv:1805. 12471.
8. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding.
9. Kitaev, N., Kaiser, Ł., & Levskaya, A. (2020). Reformer: The Efficient Transformer. Ανακτήθηκε από http://arxiv.org/abs/2001.04451.