Attention Is All You Need [Implementation & Demo]

Aniket Pandey
Department of Computer Science
George Mason University
apandey7@gmu.edu

Adel Alkhamisy
Department of Computer Science
George Mason University
aalkhami@gmu.edu

1 Introduction

1.1 Task / Research Question Description

In an encoder-decoder setup, large recurrent or convolutional neural networks provide the foundation of the most widely used sequence transduction models. The top-performing models additionally use an attention mechanism to link the encoder and decoder. Vaswani et al. suggest the a transformer model, a new straightforward network architecture that completely eschews convolution and recurrence in favor of attention methods.

In (Vaswani et al., 2017), the model is tested on two Machine Translation task namely English-to-German and English-to-French and acheive a BLEU score of 28.4 and 41.8 respectively.

1.2 Motivation

We are reproducing (Vaswani et al., 2017) because the Transformer algorithm introduced in the paper is foundational to our research interests. Transformers are the cutting-edge in zero-shot, pattern-learning computational systems and provide a workable way to combine different modalities into a single algorithm. It is crucial that we can duplicate their results in order to verify that our code works as intended.

1.3 Proposed Approach

Our approach to reimplementing the paper is very simple and straight forward. We will first start by getting the WMT 2014 datatset and creating an efficient pipeline to read the data in our project. Following the pipeline step we will start building the model. The paper has very clear and concise description of their model. The first is encoder and decoder stack. The decoder introduces a third sub-layer which performs multi-head attention over the output of the encoder stack. In addition to attention sub-layers, each of the layers in our encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically.

The authors use learned embeddings to convert the input tokens and output tokens to vectors of dimension d_{model}. They also add positional encodings to the input embeddings at the bottoms of the encoder and decoder stacks. By following the above steps we will be successfully able to implement the transformer model.

Next step is getting the resources for performing the training. Luckily for us we have GMU ORC which gives us access to a maximum of 10 GPU on a single click. However for our case we will not require these many GPUs as there are no computations in the model design that call for using the entire training batch. We can now divide the training batches into smaller batches to calculate gradients. We are able to generate gradients that are computationally similar to those produced with higher batch sizes because these gradients can be stored over numerous sub-batches. The time it takes to train a single model will take longer as a result of this strategy but that shouldn't affect the fundamental calculations in any way.

The optimizer hyper-parameter and regularization details is specified in paper. So, our next step will be to train the model and see if we get the BLEU score close to what is presented in the paper.

We also plan to present a demo of the transformer model to the entire class.

1.4 Likely challenges and mitigations

The main challenge while working on this problem statement is obtaining the WMT 2014 dataset and creating a efficient pipeline that will encode the data using byte-pair encoding and feeds the data to a model for training and inference. We will need to completely understand the encoder

decoder block and implement it alongside multi-headed attention mechanism that makes the transformer model. On top of this we need to build an infrastructure that will combine all of the previous steps together for testing.

2 Experiments

2.1 Datasets

To test the correct implementation of our transformer model we will be first using WMT 2014 English-to-German and English-to-French task. We would also like to contrast our work with WMT 2014 English-to-Hindi task.

2.2 Baselines

As a baseline we will compare our model with the BLEU score obtained by (Vaswani et al., 2017). The transformer model introduced in paper when tested with machine translation task produces a BLEU score of 28.4 and 41.8 on English-to-German and English-to-French task respectively.

2.3 Timeline

Week	Goal
Week 1	Understanding the paper and creating a high-level structure of the code base.
Week 2	Pipeline to efficiently read data into the project.
Week 3	Implementation of the Transformer model.
Week 4	Continuing with the implementation of the model.
Week 5	Testing our implementation on translation task.
Week 6	Ablation study.
Week 7	Creating a demo to present before the class and Report.

Table 1: Timeline.

We will start with creating pipeline to efficiently read dataset into project. Next step will be to implement the transformer code base, model architecture and training jobs as described in paper. Creating job on `https://ondemand.orc.gmu.edu` for multiple GPUs. Once the above setup and implementation is complete we will validate and test our implementation for the translation task. We will utilize the productivity of each of our team member and all work will be done in collaboration. Additionally, a GitHub repository for the same will be created and step by step progress can be tracked using the git commits. To complete the work successfully we will also employ the practice of pair-programming and every week distribute work among ourselves and collaborate among us if a member needs help.

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.