Adapters as a Pre-training Tool for Enhanced Performance
Adapt Or Die : CS 7643

Anthony Menninger, Jacob G Ressler-Craig, Liudmila Tuzova
Georgia Institute of Technology
amenninger3@gatech.edu, jgrc3@gatech.edu, ltuzova3@gatech.edu

Abstract

Add abstract

1. Introduction/Background/Motivation

JACOB

2. Approach

Our project had three main phases: reproduce initial pre-training results, replicate the results using an adapter architecture and explore potential novel benefits in using adapters.

2.1. Reproducing Results

Our first goal was to reproduce results from Gururangan 2020 [1] for Domain Adaptive Pre-Training (DAPT), Task Adaptive Pre-Training (TAPT) for at least one domain. This would allow us to create a nuts and bolts understanding of the pre-training process and ensure we could make it work.

2.1.1 DAPT Training

Our first unexpected challenge related to data. Our proposal research had shown that this paper had an online repository [2] that held both the code and data used for the paper. While the Task data was available as expected, the Domain data, which was much larger, was not. We decided to focus on the Amazon Review data because we were able to identify what we think is a similar source. Gururangan 2020 identifies He, McAuley 2016 [14] as the Domain data source, which led to Ni, Li, McAuley [15] paper, which in turn has a website with Amazon reviews.

Gururangan 2020 describes 24.76 million reviews in their dataset. McAuley's site contained several versions, none of which matched this. We chose a filtered set (5-core) of the 2018 dataset, as we thought it was most likely closest to the original. It contained 37 different categories with a total of 75 million reviews. We sampled each category proportionally to get to 25 million and shuffled then shuffled the data.

A key challenge we had anticipated was the computational load of running LLM models. The roberta-base model we used contains 124 million parameters. While relatively small by today's standards, this is still quite large when performing training. Gururangan 2020 used a large batch size for DAPT pre-training, which required the use of gradient accumulation. While this allows training to fit into memory, smaller batch sizes creates longer the run times. The pre-training performs Masked Language Modeling, which does not use labels, so we did have one option. The context window for the roberta-base model is 512 tokens and the average review was 87 tokens long. Each training batch was mostly empty, but used the same amount of memory for computing gradients for backprop. We could put almost 6 reviews on average in each row, increasing training efficiency by almost a factor of 6, while still exposing the model to the same number of reviews. We used this for all DAPT pre-training. We were not able to use this enhancement for classification, because a label is attached for each review preventing mingling of different reviews.

One other improvement we were able to use on the initial reproduction was the use of PyTorch compilation. Since version 2.0, PyTorch allows for compilation of the computation graph, which in our case almost doubled performance, even with the overhead associated with the compilation. This brings us to our second unexpected challenge, which was the effective age of the Gururangan 2020 codebase. The versions of tools needed to run the allennlp tool set [16] are quite old and we were not able to install them on the Google Colab platform [12]. One of our team members had access to a local environment with some computational resources where she was able to make adjustments to allow for installation, but it was tedious. This highlighted the importance of using a robust set of platform tools that will be kept current and working.

At this point we worked in parallel, with one team member reproducing results using code from the paper and an-other reproducing the papers results using the HuggingFace transformers platform [10] to reproduce the results in Google Colab. The huggingface platform provides several powerful tools for Deep Learning.

Stored datasets and our trained models on the hugging face platform - add appendix?

We used the roberta-base HuggingFace module

2.1.2 Classification

F1 score Dataset imbalance

2.1.3 TAPT Training

2.2. Replicating using Adapters

2.3. Adapter Exploration

3. Experiments and Results

LYUDMILA

4. Work Division

ADD PARAGRAPH. Table is added and needs to be filled in. This all counts after the 6 page limit.

5. Miscellaneous Information

References

[1] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, Noah A. Smith, “Don’t Stop Pre-training: Adapt Language Models to Domains and Tasks“, In ACL 2020. 1

[2] https://github.com/allenai/dont-stop-pretraining 1

[3] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, Sylvain Gelly, “Parameter-Efficient Transfer Learning for NLP”, In ICML 2019.

[4] McCloskey, M. and Cohen, N. J., “Catastrophic interference in connectionist networks: The sequential learning problem”, In Psychology of learning and motivation. 1989.

[5] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, Iryna Gurevych, “ AdapterHub: A Framework for Adapting Transformers”, arXiv preprint.

[6] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rü cklé, Kyunghyun Cho, Iryna Gurevych, “ AdapterFusion: Non-Destructive Task Composition for Transfer Learning”, arXiv preprint.

[7] Sylvestre-Alvise Rebuffi, Hakan Bilen, Andrea Vedaldi, “Learning multiple visual domains with residual adapters”, In NeurIPS 2017.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding.

[9] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pre-training approach. arXiv:1907.11692.

[10] HuggingFace roberta-base (https://huggingface.co/FacebookAI/roberta-base) roberta-large (https://huggingface.co/FacebookAI/roberta-large) 2

[11] https://adapterhub.ml/

[12] Google Colab 1

Student Name	Contributed Aspects	Details
Anthony Menninger	TBD	TBD
Jacob G Ressler-Craig	TBD	TBD
Liudmila Tuzova	TBD	TBD

Table 1. Contributions of team members.

[13] Vaswani, Ashish, et al. ”Attention is all you need.” *Advances in neural information processing systems* 30 (2017).

[14] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. 1

[15] Jianmo Ni, Jiacheng Li, Julian McAuley Empirical Methods in Natural Language Processing (EMNLP), 2019 Justifying recommendations using distantly-labeled reviews and fined-grained aspects 1

[16] Matt Gardner, Joel Grus , Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, Luke S. Zettlemoyer 2017 AllenNLP: A Deep Semantic Natural Language Processing Platform 1