Transformer Adapter Capstone Project

Mandeep Baines (msb@), Shruti Bhosale (shru@), Siddharth Goyal (sidgoyal@)

Motivation

- What problem did this project try to solve?
  - To achieve a SoTA result on a down-stream NLP task, a pre-trained Transformer language model is fine-tuned to the task. With standard fine-tuning, the pre-trained weights and a new top-layer are co-trained resulting a whole new set of weights for each new task. For large models which can be 100s of MB in size, storing and sharing a complete set of weights for each task can be a challenge for multi-task applications.
  - One possible solution is to train a single multi-task model. However, training a single multi-task model is challenging to train and requires simultaneous access to all tasks during training. Creating a single multi-task model by sequential fine-tuning runs into the problem of catastrophic forgetting (McCloskey and Cohen, 1989). Recently adapters (Rebuffi et al., 2017; Houlsby et al., 2019) have appeared as a parameter-efficient alternative to fine-tuning. On the GLUE benchmark, adapters almost match (Houlsby et al., 2019) the performance of fully fine-tuned BERT, but uses only 3% task-specific parameters, while fine-tuning uses 100% task-specific parameters.
- Who cares? If you are successful, what difference will it make?
  - Adapter-based tuning requires training much fewer parameters while attaining similar performance to fine-tuning. With full fine-tuning, avoiding over-fitting of the adaptation corpus can be challenging. This is much less of an issue with adapters. For multi-task learning, adapter training permits sequentially training while standard training does not. Finally, adapters are modular, composable (Pfeiffer et al., 2020) and easily share-able. In the future, an updated adapter (order of MB) could be deployed to update a task-specific model in the field instead of an entirely new model checkpoint (order of GB)

Problem

- In Gururangan et al. (2020), the authors show that domain and task adaptation of the RoBERTa language model (LM) prior to task fine-tuning gives better results than task fine-tuning alone. The improvement is more prominent the lesser the overlap between task domain and the original corpus used to pre-train the language model. We would like to understand if it possible to achieve similar results with adapter-based methods.

- Other interesting problems:
  - Create adapters for a transformer not yet adapted (for which fine-tuned modes already exist) and compare adapter performance to fine-tuning.
  - Create an adapter for a new task (for which fine-tuned model already exists) and compare adapter performance
  - Experiment with different adapter architectures and configurations for which there is already a published result and achieve a new state of the art results in size or performance.
  - In the ablation study in Houlsby et al. (2019), the authors show that adapters at lower layers have less impact than adapters at higher layers. Experiment with smaller adapters at lower layers and large adapters at higher layers in order to achieve a SoTA results with less parameters.

Approaches

- How is the problem approached today, and what are the limits of current practice?
  - In Gururangan et al. (2020), the author's use LM fine-tuning to adapt a pre-trained RoBERTa model to a domain and then a task. Fine-tuning requires generating a new set of weights and must be done serially.
- What are the baselines? Is there any potential approaches students can start with, without major change from the baselines?
  - Students can use the adapter-transformers framework (Pfeiffer et al., 2020) to build and train domain and task specific adapters to compare with the previously published results (Gururangan et al., 2020) obtained through standard fine-tuning.
  - The task and domain adapters can be trained sequentially or in-parallel. The adapters can be composed or stacked.

Metrics

- What are the common datasets and benchmarks?
  - What's the availability of the datasets? Are they open-sourced? Is there any restriction regarding how students should use them?
    - Students can use the same datasets as the original paper or potentially the kaggle arXiv dataset.
- What is the state-of-the-art approach, if applicable?
  - Model fine-tuning.

Scope

- What are possible directions that students could explore? E.g. In terms of modeling, data, efficient computation.
  - students could include an ablation study to show the impact of adapter configuration and size on performance on the task
  - students could explore all of the domains explored in the paper or just the CS domain where domain and task adaptation were most beneficial
  - students could reproduce the results from the paper as a baseline or save time and take the results as presented
  - students could omit domain adaption or task adaption
- How much work in each direction would justify a good grade?
  - a team of 3 should be able to reproduce the main results of the paper related to Task-Adaptive Pretraining. The results related to Augmenting Training Data for Task-Adaptive Pretraining can be omitted.
- What is the ideal size for a team tackling this project?
  - 2 to 4 people

Resources

- https://course.fast.ai/#using-a-gpu
- https://adapterhub.ml/
- https://huggingface.co/transformers/

References

1. Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, Noah A. Smith, "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks", In ACL 2020.
2. 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.
3. McCloskey, M. and Cohen, N. J., "Catastrophic interference in connectionist networks: The sequential learning problem", In Psychology of learning and motivation. 1989.
4. Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, Iryna Gurevych, "AdapterHub: A Framework for Adapting Transformers", arXiv preprint.
5. Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, Iryna Gurevych, "AdapterFusion: Non-Destructive Task Composition for Transfer Learning", arXiv preprint.
6. Sylvestre-Alvise Rebuffi, Hakan Bilen, Andrea Vedaldi, "Learning multiple visual domains with residual adapters", In NeurIPS 2017.