Review

How to Use Transformers for Transfer Learning?

Maryam Ebrahimzadeh * and Hamidreza Asadi

Posted Date: 31 May 2023

doi: 10.20944/preprints202305.2185.v1

Keywords: optics; photonics; light, lasers; journal manuscripts; LaTeX template

Article

How to Use Transformers for Transfer Learning?

Maryam Ebrahimzadeh and Hamidreza Asadi

Azad University Najafabad Branch

Abstract: Transformers are increasing replacing older generation of deep neural networks due to their success in a wide range of application. The dominant approach of using transformers is to pre-train them on a large training dataset and then fine-tune them on a downstream task. However, as transformers becoming larger, the fine-tuning approach is become an infeasible approach for transfer learning. In this short survey, we list a few recent methods that makes using transformers based on transfer learning more efficient.

Keywords: optics; photonics; light; lasers; journal manuscripts; LaTeX template

1. Introduction

Transformer neural networks are becoming increasingly popular in a wide range of applications in neural language processing [4,11,20], computer vision [1,3,22], and video processing [6,12,13]. The power of transformers stems from the attention mechanism, which allows the model to focus on relevant parts of the input sequence when processing each position. Additionally, it enables transformers to process the input sequence in parallel and hence in a short time. This parallelism allows transformers to take full advantage of GPUs and TPUs, leading to faster training and inference times.

Transformers have been used in pre-training and transfer learning approaches [2]. These models are initially trained on a large training dataset and then are fine-tuned for specific downstream tasks with smaller labeled datasets. This approach has significantly improved the performance of machine learning using less labeled data, leading to state-of-the-art performance on a wide range of tasks, including machine translation, question-answering, sentiment analysis, and named entity recognition.

Despite being very effective, using fine-tuning for transfer learning is becoming infeasible when the base model is a transformer. The major reason is that when we fine-tune a model, its generalization power is lost because it becomes specialized on the downstream task. As a result, a new version of the transformer should be stored per task. However, as the transformers becoming larger, storing them is becoming an infeasible solution. Our goal in this short survey is review some works that try to make transfer learning more efficient when the base model is a transformers.

2. Adapters

Adapters are a technique used in transformer models to enhance their flexibility and enable efficient parameter sharing [16]. They allow for easy and modular model extension without requiring significant changes to the original transformer architecture. Adapters are small, task-specific modules that are inserted between the layers of a transformer model. Each adapter consists of a bottleneck layer (typically a feed-forward neural network) followed by a projection layer. The bottleneck layer reduces the dimensionality of the input, while the projection layer maps it back to the original dimension.

The key idea behind adapters is parameter sharing [18]. Instead of creating separate task-specific models or fine-tuning the entire transformer for each new task, adapters enable reusing most of the pre-trained transformer's parameters. The adapter parameters are relatively small compared to the entire model, which allows for efficient parameter sharing across multiple tasks. Adapters are designed to be easily plugged into different layers of the transformer. They can be inserted between any pair of transformer layers, enabling different layers to have different adapter modules. This flexibility allows

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 May 2023 doi:10.20944/preprints202305.2185.v1

for fine-grained control over which layers and positions in the transformer benefit from task-specific adaptation.

The use of adapters improves the scalability and efficiency of transformer models [19]. Instead of adding task-specific layers or fine-tuning the entire model, which can be computationally expensive and memory-intensive, adapters provide a lightweight solution. The number of adapter parameters is much smaller than the transformer's parameters, making it more efficient to add adapters for multiple tasks. Additionally, adapters are particularly useful for transfer learning and multi-task learning scenarios because fine-tuning can be done only them which preserves the generalization power of the base transformer. Adapters enable the base model to generalize to new tasks and facilitate knowledge transfer across related tasks. We review three strategies to train adapter weights that are more efficient even compared to adapter fine-tuning.

3. Adapter Fusion

Adapter fusion is an extension of the adapter technique for transformers that enables the sharing and combination of information across different tasks or domains[15]. It allows for the integration of multiple adapters into a single model to leverage the knowledge learned from various sources.

Multiple adapters: In adapter fusion, multiple adapters are added to a transformer model, each representing a different task or domain. These adapters can be designed and trained independently for specific tasks or datasets. Each adapter captures task-specific or domain-specific information. Adapter fusion facilitates the sharing of information across different adapters. Instead of treating each adapter as a separate module, fusion mechanisms enable the adapters to exchange information and combine their knowledge during computation.

Various fusion mechanisms can be used to combine the information from multiple adapters. Some common fusion approaches include weighted averaging, gating mechanisms, and attention-based fusion. These mechanisms allow the adapters to influence each other's computations, enabling the model to benefit from the knowledge captured in different adapters.

Adapter fusion is particularly useful for transfer learning and domain adaptation scenarios [21]. By fusing adapters from different pre-trained models or domains, the model can leverage the knowledge captured by each adapter and effectively transfer it to new tasks or domains. The fusion process helps integrate and combine the task-specific information learned from different sources. Adapter fusion has been shown to improve the overall performance and robustness of transformer models. By combining information from multiple adapters, the model can capture a broader range of task-specific or domain-specific patterns. This integration helps in tasks that may benefit from knowledge from multiple sources or in scenarios with limited labeled data.

Adapter fusion is also compatible with the fine-tuning process. The adapters can be trained independently and then fused during fine-tuning or adapted together as a unified model. The flexibility of adapter fusion allows for easy extension and adaptation of transformer models to new tasks, domains, or combinations of tasks. To sum up, adapter fusion enhances the transferability, flexibility, and overall capabilities of transformer models in multi-task learning, domain adaptation, and other complex scenarios.

4. Adapter Distillation

Knowledge distillation[7] has been used extensively to transfer knowledge from a large, complex model (referred to as the "teacher" model) to a smaller, more efficient model (referred to as the "student" model). The goal is to train the student model to mimic the behavior and predictions of the teacher model. The process of knowledge distillation typically involves two steps. First, the teacher model is typically a large, well-performing model, which is trained on a large dataset. Second, the student model is trained to mimic the behavior of the teacher model. intuition behind knowledge distillation is that the teacher model has learned valuable insights and generalizations from its training on a large dataset. By using the teacher's predictions as additional information during the training of the student model,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 May 2023 doi:10.20944/preprints202305.2185.v1

the student can benefit from this knowledge and potentially achieve better performance compared to training it from scratch. Knowledge distillation can be extended to transfer knowledge between different models or across different tasks, not necessarily restricted to teacher-student scenarios.

Knowledge distillation can be used in the context of adapter training. To this end, specialized neural network modules introduce new parameters to the neural network [5,9,14]. The specialized neural network also incorporates a distillation mechanism to merge the newly acquired information with the previously learned knowledge. Importantly, this consolidation of information is achieved within a fixed parameter space, minimizing the required storage. This approach enables users to control memory consumption while maintaining state-of-the-art performance when dealing with sequences of multiple tasks. This fine-grained memory control is particularly valuable in industrial applications. In contrast, an alternative approach that allows the model to grow in size with the number of tasks would necessitate hardware changes to accommodate the expanding memory requirements. Such a scenario would pose challenges, including heightened risk of system instability and the need to make conservative hardware decisions.

5. Adapter Generation

Although adapter fusion is helpful but as the number of tasks grow and we need to store a set of adapters for each task, the size of adapters will become larger and add storage cost. Adapter generation for transformers refers to the process of creating and integrating adapters into a pre-trained transformer model. As a result, a model is trained to learn proper adapters for a downstream task which is far smaller compared to the side of adapter layers. In other words, rather than storing the original adapter layers, we learn how to store the knowledge for generating them in a compressed way.

A practical approach for adapter generation is to extend hypernetowrks in the context of transformers. A hypernetwork is a type of neural network that is designed to generate the weights or parameters for another neural network [8]. In other words, it is a network that produces the weights of another network, often referred to as the "target network" or "main network". The main idea behind a hypernetwork is to use one neural network, the hypernetwork, to generate the weights or parameters of another neural network, rather than directly learning those weights through traditional methods like backpropagation[10,17]. This approach has several potential advantages. First, a hypernetwork can be used to dynamically generate weights for different instances of the target network. This allows for more flexibility and adaptability, as the hypernetwork can generate different sets of weights depending on the input or context. A hypernetwork can be used to reduce the number of adapters that need to be learned in the transformer. This can be particularly useful in scenarios where the transformer network needs to be deployed on devices with limited computational resources or memory.

6. Conclusion

We survey recent advances on transfer learning when the base model is a transformer. Due to popularity of transformers, we foresee that transformers will continue to become more popular and as a result research on this direction is necessary to benefit from the full potential of transformers.

References

1. Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.
2. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

3. Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 589–598, 2021.
4. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
5. Beyza Ermis, Giovanni Zappella, Martin Wistuba, Aditya Rawal, and Cedric Archambeau. Memory efficient continual learning with transformers. *Advances in Neural Information Processing Systems*, 35:10629–10642, 2022.
6. Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253, 2019.
7. Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.
8. David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.
9. Gousia Habib, Tausifa Jan Saleem, and Brejesh Lall. Knowledge distillation in vision transformers: A critical review. *arXiv preprint arXiv:2302.02108*, 2023.
10. Xisen Jin, Bill Yuchen Lin, Mohammad Rostami, and Xiang Ren. Learn continually, generalize rapidly: Lifelong knowledge accumulation for few-shot learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 714–729, 2021.
11. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
12. Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.
13. Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3163–3172, 2021.
14. Mohammadmahdi Nouriborji, Omid Rohanian, Samaneh Kouchaki, and David A Clifton. Minialbert: Model distillation via parameter-efficient recursive transformers. *arXiv preprint arXiv:2210.06425*, 2022.
15. Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020.
16. Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*, 2020.
17. Yi Tay, Zhe Zhao, Dara Bahri, Don Metzler, and Da-Cheng Juan. Hypergrid transformers: Towards a single model for multiple tasks. 2021.
18. Justin K Terry, Nathaniel Grammel, Ananth Hari, Luis Santos, and Benjamin Black. Revisiting parameter sharing in multi-agent deep reinforcement learning. *arXiv preprint arXiv:2005.13625*, 2020.
19. Bethan Thomas, Samuel Kessler, and Salah Karout. Efficient adapter transfer of self-supervised speech models for automatic speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7102–7106. IEEE, 2022.
20. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
21. Rongsheng Zhang, Yinhe Zheng, Xiaoxi Mao, and Minlie Huang. Unsupervised domain adaptation with adapter. *arXiv preprint arXiv:2111.00667*, 2021.
22. Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.