Lifelong Language Learning with Adapter based Transformers

Abstract

Continual Learning is important for real-world natural language processing applications, where computational systems are required to interact with continuous streams of tasks and language over time. When forced to adapt to new tasks and inputs, language models experience catastrophic forgetting. The current generative replay-based algorithms are not scalable to many tasks, and their performance may degrade from a change in the task order. In this paper, we propose a model based on network growth - a pre-trained Transformer with Adapter modules for each task - that sequentially learns new NLP tasks in various domains and prevents catastrophic forgetting without retraining the model from scratch. We train and maintain light weight adapter modules sequentially for each task. Without increasing network growth by more than 15% and avoiding replay and task order bias, the current design allows us to increase average task accuracy by 1.3% over the baseline models.

Keywords: Continual Learning; Lifelong Language Learning; Adapter Transformer; LAMOL

1. Introduction

Humans can continuously accumulate, develop, and transfer knowledge and skills throughout their lifetimes, giving rise to lifelong learning principles. Continuous learning is crucial in real-world natural language processing applications, where computer systems must interact with ongoing streams of data and language across time. Isolated learning is currently the most used paradigm in machine learning. In isolated learning, the model experiences catastrophic forgetting or interference due to non-stationary data distribution that biases the model, making it unable to remember the information it has previously learned on a stream of tasks joined to be trained sequentially.

In comparison, continual learning focuses on the ability of the model to learn continuously and adaptively over time, which allows it to learn new information without forgetting the past knowledge. In this work, we focus on lifelong language learning (LLL) on a continuous stream of NLP tasks. The performance of LLL is typically viewed as having an upper bound provided by multi-task learning. There is still a performance gap between other frameworks for LLL Sun et al. (2019); Huang et al. (2021); Kanwatchara et al. (2021) and multi-task learning. In these previous works, continual learning is maintained through generative replay which limits its application to large number of tasks and the model performance changes as the task order changes Sun et al. (2019).

We enhance the current LLL strategies by proposing a novel approach - Adapter based Transformer - a dynamic architecture based on network growth. Particularly, we propose to use task-specific adapters for each task, which equips the framework to learn new tasks while retaining information about older tasks and thus, avoid catastrophic forgetting.

Our main contributions are :

© 2022 .

- We present Adapter based Transformers, our framework is space efficient. Due to the adapters being lightweight, very little extra memory is utilized.
- Number of tasks to be trained need not be known in advance, we can always learn new tasks by adding new adapter modules.
- The performance of our framework is independent of the order of the tasks.

2. Background and Related Work

2.1. Continual Learning

Among existing continual learning approaches for LLL, replay-based methods Sun et al. (2019) and regularization-based methods Huang et al. (2021) have been widely applied to NLP tasks to enable large pre-trained models to acquire knowledge from streams of textual data without forgetting the already learned knowledge. LAMOL Sun et al. (2019) is a data-based LLL approach that simultaneously learns to solve a new task, while generating pseudo samples for previous tasks to train alongside the new task. A single model is used here, and no extra generator is used. Rational LAMOL Kanwatchara et al. (2021) is an enhancement on LAMOL. This framework applies freezing to critical components, that are identified by rationales, which are part of input texts that best explain the prediction or class labels, in transformer based language models, to maintain previously learned knowledge while being trained on a new task. This is done as pseudo-sample generation may not be sufficient to prevent catastrophic forgetting. In Information Disentanglement based Regularization Huang et al. (2021), the framework focuses on how to generalize models to new tasks, rather than just focusing on preserving knowledge from previous tasks. It uses a multi layer encoder for a given sentence, that outputs hidden representations which contain generic as well as task specific information, and two disentanglement networks to extract the generic and specific representations. While training on new tasks, the model regularizes both these representations to different extents, to better remember previous knowledge as well as transfer to new tasks.

2.2. Adapters

In a large pre-trained model with parameters Θ, adapters Houlsby et al. (2019) are neural modules with a limited number of newly added parameters Φ. While keeping Θ constant, the parameters Φ are learned on a target task; as a result, Φ learn to encode the task-specific representations in the pre-trained model's intermediate layers. When compared to the amount of parameters in a pre-trained model, the number of parameters in adapters is parameter-efficient, comprising only 1% to 3% of those parameters. Due to their modularity and small size, they accelerate training iterations and are shareable and composable Pfeiffer et al. (2020). Different adapter architectures are tested in Houlsby et al. (2019), and the results empirically demonstrate the effectiveness of a two-layer feed-forward neural network with a bottleneck as an adapter. A schematic representation of the adapter module is provided in 1.

3. Continual Learning with Adapter based Transformers

We formalize our problem as a lifelong learning problem on a set of NLP tasks $\{T_1, \ldots, T_N\}$ taken sequentially, where the number of tasks may be unknown. We propose Adapter based Transformers, a framework that contains adapter modules specific to each task. Our framework sequentially learns new NLP tasks from different domains and avoids catastrophic forgetting by storing relevant information of older tasks in respective task-specific adapters. We used a pre-trained GPT-2 model Radford et al. (2018) as the language model. Whenever a new task comes, we add two new adapters specific to the task, to each layer of the transformer. The task-specific adapter is then trained, keeping the weights of the underlying language model and previous task specific adapters frozen. Therefore, the model remembers previous tasks perfectly, while simultaneously being able to learn the new task. Since adapters have smaller number of parameters as compared to the original network, the model size growth will be very small as the number of tasks grows.

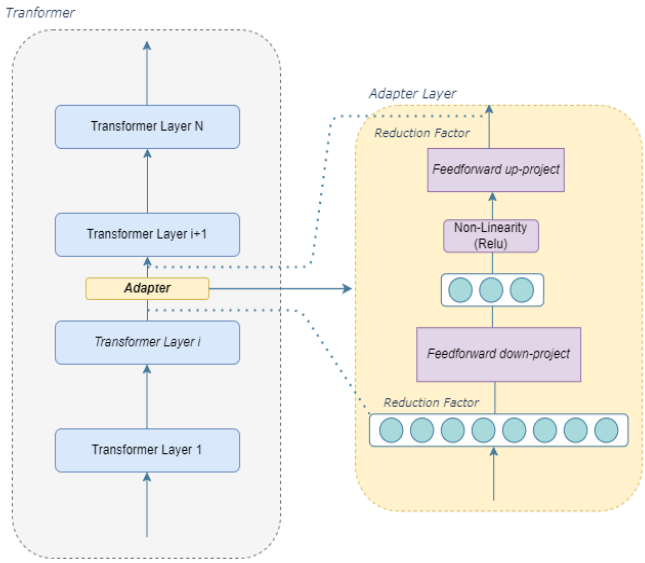

Figure 1: Adapter Model

We denote Θ as the parameters of pre-trained GPT-2 , and Φ_i as the parameters corresponding to task-specific Adapter A_i associated with Task T_i.

For each new task T_i, the parameters of the pre-trained transformer Θ as well as older task-specific adapters $A_1, \ldots, A_{i-1}$are kept frozen, and the adapter corresponding to the new task is trained. The parameters Φ_i of Adapter A_i associated with task T_i are randomly initialized, and A_i is then trained on the new task T_i. Freezing of the pre-trained transformer parameters and older task specific parameters enables our framework to retain knowledge of older tasks while simultaneously learning new tasks. The task specific adapter parameters are learnt by optimizing the loss function $\mathcal{L}_i()$ corresponding to the task T_i, for instance the cross-entropy loss for many NLP tasks.

$$\Phi_i^* = \underset{\Phi_i}{\operatorname{argmin}} \quad \mathcal{L}_i(\Theta, \Phi_i) \tag{1}$$

Please note that there is no language model (LM) loss to be used to train the proposed adapter transformer based CL model unlike prior replay based CL models such as LAMOL. This is because we don't have to generate pseudo samples from the previous tasks to train the transformer to remember previous tasks.

Dataset	Train	Test	Metric
SST	6920	1821	EM
SRL	6414	2201	nF1
WOZ	2536	1646	dsEM

Table 1: Details of dataset size and metrics. nF1 - normalized F1 score; EM - exact match between texts; dsEM - turn-based dialogue state exact match

Model	Train Time
Adapter-Transformer	2650 secs
LAMOL	2919 secs

Table 2: Training time taken for our proposed Adapter-Transformer method and baseline LAMOL

4. Experimental Setup

4.1. Dataset Details

We conducted experiments on the following datasets to evaluate our proposed framework. Table 1 contains a summary of the datasets, dataset sizes, and metrics.
Stanford Sentiment Treebank(SST) Socher et al. (2013) - which is a sentiment analysis task consisting of sentiments (binary - positive or negative) corresponding to a movie review.
Semantic Role Labeling (SRL) He et al. (2015) - which is a role labeling task where Wikipedia domain of QA-SRL 1.0 is used, and the task is to assign labels that assigns semantic meaning to a phrase or sentence.
Goal-oriented dialogue (WOZ) Mrkšić et al. (2017) - which is a reservation task from English Wizard of Oz restaurant, and it comes with predefined information that will assist an agent to make a reservation for the customer.

4.2. Data Formatting

Following the process used in decaNLP McCann et al. (2018) and LAMOL Sun et al. (2019), samples from the datasets we use are framed into a SQuAD-like scheme, which consists of context, question, and answer. Special tokens are also added: ANS is inserted between question and answer. As the context and question are known during inference, decoding starts after inputting ANS. EOS is the last token of every example. Decoding stops when EOS is encountered.

5. Results

We compare the performance of Adapter-Transformers with the LAMOL Sun et al. (2019) as the baseline. The training and inference of both the models are done on the NVIDIA Tesla V100 Graphics cards of 32 Gbs. Both the models are trained sequentially on the train samples from the three data sets SST, SRL and WOZ and their performance is obtained on the test samples of these data sets after completing the sequential training. The results are provided in Table 3. The proposed transformer adapter model sees a 1.3% increase in average task accuracy as compared to baseline, as can be seen in Table 3. In another experiment we conducted, as we added another task (Amazon Review d'Autume et al. (2019) along with SST, SRL, WOZ), our framework performed better than LAMOL on 2 out of 4 tasks (WOZ and amazon) and the EM score was 62.41 on the amazon review

Dataset	Metrics	Adapter-Transformer	LAMOL
SST	EM	90.88	90.94
SRL	nF1	67.96	68.38
WOZ	dsEM	88.54	85.75

Table 3: Details of averaged metric scores for our proposed Adapter-Transformer method and baseline LAMOL

Dataset	Task Order (SRL,SST, WOZ)	Task Order (SRL,WOZ, SST)
SST	90.6	90.8
SRL	67.3	67.7
WOZ	88.3	88.1

Table 4: Details of average metric scores for different task ordering

dataset while LAMOL was 52.50. This shows that as we add more tasks the CL capability of the proposed approach becomes better. Task ordering doesn't have much effect on model performance which can be seen in Table 4.

As our proposed model is light-weight and avoids replay, we observe in Table 2 that it has faster training time compared to baseline, with less than a 15% increase in network growth, as can be seen in Table 5.

6. Parameter Growth Study

Table 5 indicates the total number of parameters in our proposed model. It has the trainable adapter parameters as well the parameters of the baseline LM that are kept constant. For a particular task, we train 11,822,592 adapter parameters which leads to around 13.9% growth in the network. In *"Layerwise + Fix"*, *Fix* refers to the embedding that is only needed once, the word embedding layer parameters (3,860,448) and positional embedding layer parameters (768,432); *Layerwise* for GPT-2 refers to the number of parameters in each of the 12 decoder layers (attention and feed forward) which amounts to 7,087,104 per layer, and for the Adapter GPT-2 model it amounts to 8,072,320 per layer (*attention + feed forward + adapter parameters*)

	Adapter-Transformer	LAMOL	Increase	Increase%
`1 layer`	8,072,320	7,087,104	985,216	13.90
12 layers—	96,867,840	85,045,248	11,822,592	13.90
`Total(Layerwise+Fix)`	136,254,720	124,432,128	11,822,592	9.50

Table 5: Summary of Parameter increase in proposed model vs baseline

7. Conclusion

We propose Adapter-Transformers, a framework for LLL, that can efficiently learn information of new tasks without forgetting older tasks. With the current design, we can improve average task accuracy by 1.3% over LAMOL without increasing the network growth by more than 15% and avoiding replay and task order bias.

References

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 1177, pages 13132–13141. Curran Associates Inc., Red Hook, NY, USA, December 2019.

Luheng He, Mike Lewis, and Luke Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1076. URL https://aclanthology.org/D15-1076.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. Technical Report arXiv:1902.00751, arXiv, June 2019. URL http://arxiv.org/abs/1902.00751. arXiv:1902.00751 [cs, stat] type: article.

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. Continual Learning for Text Classification with Information Disentanglement Based Regularization. Technical Report arXiv:2104.05489, arXiv, June 2021. URL http://arxiv.org/abs/2104.05489. arXiv:2104.05489 [cs] type: article.

Kasidis Kanwatchara, Thanapapas Horsuwan, Piyawat Lertvittayakumjorn, Boonserm Kijsirikul, and Peerapon Vateekul. Rational LAMOL: A Rationale-based Lifelong Learning Framework. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2942–2953, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.229. URL https://aclanthology.org/2021.acl-long.229.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. Technical Report arXiv:1711.05101, arXiv, January 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs, math] type: article.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The Natural Language Decathlon: Multitask Learning as Question Answering. Technical Report arXiv:1806.08730, arXiv, June 2018. URL http://arxiv.org/abs/1806.08730. arXiv:1806.08730 [cs, stat] type: article.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1163. URL https://aclanthology.org/P17-1163.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting

transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online, 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.7.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. LAMOL: LAnguage MOdeling for Lifelong Language Learning. Technical Report arXiv:1909.03329, arXiv, December 2019. URL http://arxiv.org/abs/1909.03329. arXiv:1909.03329 [cs] type: article.

Appendix A. Hyper-parameter Setting

Baseline LAMOL: We wanted to compare the best version of LAMOL Sun et al. (2019) with our model; hence LAMOL's best-performing hyper-parameters were used. We have set k = 20 in top-k sampling and $\lambda = 0.2$ for the weight of the LM loss. The learning rate is set up as $6.25e^{-5}$ and is scheduled to be linear with a warmup. Each task is trained for five epochs with loss as a summation of Question Answering (QA) and Language model (LM) loss and the optimizer is AdamW Loshchilov and Hutter (2019).

Adapter Transformer: To have a fair comparison with LAMOL, we used almost similar hyper-parameters. We have set $k = 20$ in top-k sampling and there is no LM loss to be used in the adapter transformer. The learning rate is set up as $6.25e^{-5}$ for all the tasks and is scheduled to be linear with a warmup. Each task is trained for 12 epochs with loss as only Question Answering (QA) loss and the optimizer is AdamW.