An Empirical Study on the Transferability of Transformer Modules in Parameter-Efficient Fine-Tuning

AkbarTajari, M.; Rajaee, S.; Pilehvar, M.T.

DOI
10.18653/v1/2022.emnlp-main.726

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing

An Empirical Study on the Transferability of Transformer Modules in Parameter-Efficient Fine-Tuning

Mohammad AkbarTajari[*1], **Sara Rajaee**[*2], and **Mohammad Taher Pilehvar**[3]
[1] Sharif University of Technology, Iran
[2] University of Amsterdam, Netherlands
[3] Tehran Institute for Advanced Studies, Khatam University, Iran
m.akbarTajari@gmail.com s.rajaee@uva.nl mp792@cam.ac.uk

Abstract

Parameter-efficient fine-tuning approaches have recently garnered a lot of attention. Having considerably lower number of trainable weights, these methods can bring about scalability and computational effectiveness. In this paper, we look for optimal sub-networks and investigate the capability of different transformer modules in transferring knowledge from a pre-trained model to a downstream task. Our empirical results suggest that *every* transformer module in BERT can act as a *winning ticket*: fine-tuning each specific module while keeping the rest of the network frozen can lead to comparable performance to the full fine-tuning. Among different modules, LayerNorms exhibit the best capacity for knowledge transfer with limited trainable weights, to the extent that, with only 0.003% of all parameters in the layer-wise analysis, they show acceptable performance on various target tasks. On the reasons behind their effectiveness, we argue that their notable performance could be attributed to their high-magnitude weights compared to that of the other modules in the pre-trained BERT. The code for this paper is freely available at https://github.com/m-tajari/transformer-transferability.

1 Introduction

Fine-tuning is widely used as a procedure to employ the knowledge learned during pre-training of language models for specific tasks (Howard and Ruder, 2018; Peters et al., 2019; Merchant et al., 2020; Zhou and Srikumar, 2022). However, fine-tuning can be a computationally expensive process, given that it usually involves updating all the parameters in transformer-based models which are often massive in size. Parameter-efficient fine-tuning methods try to ameliorate this by reducing the number of updatable parameters during fine-tuning.

Adapters (Houlsby et al., 2019; Pfeiffer et al., 2020; Wang et al., 2021; Rücklé et al., 2021; Karimi Mahabadi et al., 2021; Hu et al., 2021) try to circumvent this issue by inserting light-weight modules in the transformer blocks, tuning of which usually results in comparable performance to the full fine-tuning (while the number of updatable parameters is significantly lower). Nevertheless, introducing new parameters to an already-large model can be considered a drawback. Another category of parameter-efficient fine-tuning methods is based on the *Lottery Ticket Hypothesis* (Prasanna et al., 2020), where the goal is to find a small subset of parameters that can compete with the full fine-tuning setting. Various subsets of network parameters have been suggested as the *winning ticket*, including the connections with high magnitudes (Han et al., 2015), identity mappings (Lin et al., 2020), and dominant dimensions (Guo et al., 2021).

In this paper, we study the ability of different modules of a transformer block in knowledge transfer. Our experiments provide a more comprehensive analysis than the existing work, which usually suggests specific modules as the *winning ticket*, such as the bias terms (Ben Zaken et al., 2022). Through module-wise fine-tuning, we check if the *winning ticket* is a property that can be associated only with some specific modules in the transformer block. Our results suggest that *all* individual modules possess this property to some extent. Among these, LayerNorms prove to be highly reliable for knowledge transfer: fine-tuning only $37k$ LayerNorm weights (out of $110M$ parameters in BERT-base) is often on par with full fine-tuning on various downstream tasks. Extending this analysis, we show that tuning even only one LayerNorm can yield comparable performance and that the middle layers are the best in terms of transferability. We also investigate the reasons behind the effectiveness of LayerNorm

[*]The authors contributed equally to this work.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 10617 - 10625
December 7-11, 2022

tuning. Our experiments suggest that this could be due to the relatively high-magnitude weights in these modules. In fact, we show that tuning just a tiny fraction of high-magnitude dimensions (usually referred to as *outliers*) can lead to competitive performance on various tasks.

2 Winning Modules

According to the Lottery Ticket Hypothesis, there are small sub-networks whose performance is comparable to the over-parameterized model on different tasks (Frankle and Carbin, 2019). Several studies have been carried out to identify sub-networks across the model that can provide the best transferability (Gale et al., 2019; Evci et al., 2020; Lee et al., 2021; Guo et al., 2021; Hu et al., 2021). Nonetheless, finding the winning sub-network usually requires extra computation, which is costly in terms of time and memory. In this section, we take another look at the transformer block of BERT and focus on the ability of its different modules to transfer knowledge to various downstream tasks. More specifically, we aim to find the winning module among the different modules in the transformer-based architecture of the pre-trained BERT.

2.1 Experimental Setup

Datasets. We fine-tune our models on the GLUE benchmark (Wang et al., 2018). We leave out the WNLI (the Winograd Schema Challenge) task (Levesque et al., 2012)), given that BERT's performance on this benchmark is not much better than a random classifier. Instead, we test the models on the Corpus of Linguistic Acceptability (Warstadt et al., 2019, CoLA), the Stanford Sentiment Treebank (Socher et al., 2013, SST-2), the Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005, MRPC), the Semantic Textual Similarity (Cer et al., 2017, STS-B), the Quora Question Pairs (Wang et al., 2018, QQP), the Multi-Genre Natural Language Inference Corpus (Williams et al., 2018, MNLI), the Stanford Question Answering Dataset (Rajpurkar et al., 2016, QNLI), and the Recognizing Textual Entailment (Dagan et al., 2005, RTE). All the reported results are obtained on the corresponding development sets.

Models. We opt for bert-base-uncased, implemented by the HuggingFace library in TensorFlow (Wolf et al., 2020; Abadi et al., 2015). The maximum sequence length is set to 128. Except for the fully fine-tuned model (Full-FT), where we train the models for five epochs, the number of epochs is chosen based on the size of the tasks: 10 epochs for SST-2, QQP, MNLI, and QNLI and 20 epochs otherwise. We use the Adam optimizer with an epsilon set to 1e-6, a warmup ratio of 10%, and a batch size of 16. The only hyperparameter tuning we do is on choosing the learning rate from {1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2} to draw a fair comparison with previous work. We report the average and standard deviation of the results of three models trained with different random seeds. All the models are trained on four NVIDIA Tesla V100S-32G GPUs.

Module Settings. To find out the potential of transformer modules in transfer learning, we pick similar modules across all layers and fine-tune them while keeping the rest of the network frozen. The aim of this setup is to broaden our insights on the distribution of knowledge across the model and the adaptability of different modules to target tasks.

In every transformer block, we check for the role played by the *Multi-head* attention, *Feedforward* layer, and *LayerNorms* in knowledge transfer. Since every transformer block has two LayerNorms (attention and feedforward), we also consider fine-tuning them separately ($LayerNorms_A$ and $LayerNorms_F$). We also compare against the replicated results of BitFit (Ben Zaken et al., 2022), in which consistent bias terms across the transformer blocks are employed for fine-tuning. To verify if consistency in selecting parameters matters, we also show the results of fine-tuning only a small randomly selected subset of all the parameters with the same size as *LayerNorms* (*Random*). In the experiments, the full fine-tuning (*Full-FT*) and *Frozen* modes are considered as the upper and lower bounds, respectively.

2.2 Results

Table 1 shows our experimental results on eight tasks from the GLUE benchmark.[1] For each setting, we also report the corresponding ratio of updatable parameters (compared to full fine-tuning).

[1] As for BitFit, we were unable to carry out a full hyperparameter tuning on three tasks due to the large dataset size and computational constraints. Instead, we report results as given in the original paper, which is around 5% better than the best results we obtained for these settings.

Model	%Param.	CoLA	SST-2	MRPC	STS-B	QQP	MNLI$_m$	MNLI$_{mm}$	QNLI	RTE	*Avg.*
Full-FT	100.0%	54.7±0.7	**93.1±0.1**	90.9±1.0	89.0±0.5	**87.3±0.1**	**83.4±0.2**	**83.7±0.4**	**91.5±0.1**	71.7±1.2	82.8
Feed-Forward	51.76%	56.2±1.5	92.7±0.1	91.1±0.5	88.9±0.6	87.1±0.1	81.9±0.3	82.7±0.2	90.8±0.0	71.1±0.8	82.5
Multi-Head	25.89%	**59.0±1.3**	92.9±0.3	**91.2±0.3**	**89.1±0.5**	86.7±0.2	83.2±0.2	83.6±0.3	91.2±0.1	72.0±0.2	**83.2**
LayerNorms	0.03%	52.8±2.0	92.2±0.4	91.0±0.2	88.4±0.1	81.5±0.1	79.3±0.2	80.7±0.3	89.4±0.2	70.2±0.4	80.6
LayerNorms$_A$	0.02%	50.8±0.5	91.8±0.2	90.6±0.5	88.3±0.2	80.9±0.1	76.3±0.2	75.9±0.3	88.8±0.1	70.5±0.9	79.3
LayerNorms$_F$	0.02%	52.9±1.9	91.7±0.2	91.0±0.2	88.4±0.1	81.0±0.1	77.0±0.1	77.2±0.3	88.3±0.0	69.2±0.7	79.6
BitFit	0.12%	53.6±1.9	89.6±0.2	90.5±0.2	88.2±0.0	81.8±0.3	†81.4±0.2	†82.2±0.2	†90.2±0.2	**72.8±0.2**	81.1
Random	0.03%	35.5±3.0	86.4±0.3	85.1±0.5	84.3±0.0	74.8±0.2	64.5±0.2	66.1±0.3	76.4±0.0	60.8±0.5	70.4
Frozen	0.00%	35.2±1.6	81.5±0.3	80.7±0.1	68.3±0.1	60.1±0.2	44.0±0.3	45.1±0.2	68.8±0.1	58.6±0.5	60.25

Table 1: The performance of BERT on the GLUE benchmark with different fine-tuning strategies. We report Matthew's correlation for CoLA, F1 score for MRPC and QQP, Spearman's correlation for STS-B, and accuracy for the rest. **LayerNorms$_A$** (**LayerNorms$_F$**) stands for the scenario in which only the LayerNorms of **A**ttention (**F**eedforward) modules are set to be trainable. The best and the second-best results are highlighted for each task. † Results from Ben Zaken et al. (2022).

As can be observed, individual modules of BERT can be considered as *winning tickets* because they can achieve comparable performance to the *Full-FT* setting, despite involving significantly smaller numbers of trainable parameters. In particular, LayerNorms prove to have a high potential in transferability and adaptability to various downstream tasks with a very limited set of trainable parameters (0.034%). The performance is mostly preserved even when only one of the two LayerNorms is set to be trainable, reducing the number of effective parameters to 0.017% of that in the full fine-tuning. Moreover, our results also reveal that selecting consistent weights (similar modules across layers) has a key role in fine-tuning quality, given that the random subset of a comparable number of parameters does not lead to the same performance levels.

2.3 Token-level Classification

In addition to the sentence-level tasks of the GLUE benchmark, we also conduct experiments on two different token-level datasets to broaden our insights on the capacity of individual modules: Penn Treebank Part-of-speech tagging (Marcus et al., 1993) and CoNLL-2003 Named Entity Recognition (Tjong Kim Sang and De Meulder, 2003). For part-of-speech tagging, we use the subset of the Wall Street Journal (WSJ) portion of PTB which is freely available in the Natural Language Toolkit (Bird et al., 2009, NLTK). In this experiment, we adhere to the convention of using the cased version of BERT, given the case-sensitive nature of these token-level tasks.

Model	%Param.	PTB	CoNLL
Full-FT	100.0%	**98.5±0.1**	**94.2±0.1 / 98.6±0.0**
BitFit	0.12%	98.1±0.1	86.9±0.1 / 96.9±0.0
LayerNorms	0.03%	98.4±0.1	92.4±0.1 / 98.1±0.0
Random	0.03%	97.6±0.1	82.7±0.1 / 96.1±0.0
Frozen	0.00%	96.2±0.0	78.1±0.0 / 94.8±0.0

Table 2: The performance of BERT on Penn Treebank (PTB) and CoNLL-2003 (CoNLL) datasets with five different fine-tuning strategies. We report accuracy for PTB and F1 score (macro/micro) for CoNLL.

Table 2 summarizes the results. Similarly to what is observed on the sentence-level tasks, LeyerNorms can attain competitive performance on the two token-level tasks, despite involving just a small fraction of all the model parameters. Moreover, in comparison with the equal number of randomly selected weights, they demonstrate remarkably better performance.

2.4 Single Norms Tuning

Previous studies have reported that different layers do not contribute equally to the ultimate performance in transfer learning (Zhou and Srikumar, 2021; Rogers et al., 2020; Kovaleva et al., 2019; Mehrafarin et al., 2022). We are interested in studying the extent to which individual LayerNorms in different transformer blocks are adaptable to downstream tasks. To this end, we perform a layer-wise analysis in which the only trainable parameters are the two LayerNorms in each block and the final classifier. Therefore, the total number of fine-tuning parameters is less than 5K

Task	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	Full-FT
CoLA	44.0±1.5	43.3±2.8	46.8±2.4	47.6±1.7	46.1±1.9	47.0±2.1	47.0±3.4	**48.1±0.9**	47.1±2.5	45.7±1.5	42.3±1.9	36.9±0.2	**54.7±0.7**
MRPC	84.7±0.5	85.9±0.8	84.2±0.3	85.9±0.6	**89.0±1.1**	88.8±0.4	87.5±0.7	86.1±0.2	86.6±0.2	86.4±0.7	85.2±0.5	82.6±0.1	**90.9±1.0**
STS-B	85.1±0.3	85.7±0.1	86.1±0.1	86.1±0.2	86.7±0.3	87.1±0.1	86.9±0.1	**87.2±0.1**	86.7±0.1	86.6±0.1	86.5±0.1	83.5±0.2	**89.0±0.5**
RTE	61.5±1.7	65.3±0.6	63.9±1.5	64.1±0.6	65.2±0.6	64.1±1.5	67.3±0.3	**67.5±1.0**	63.5±0.8	65.8±0.3	67.0±0.2	60.5±1.4	**71.7±1.2**

Table 3: The performance of layer-wise fine-tuning of LayerNorms on the selected downstream tasks for BERT. The LayerNorms in the middle layers tend to have the highest transferability.

(3,072 and 1,538 for LayerNorms and the classifier, respectively)[2], which is about 0.003% of all the parameters. Due to our limited computational resources, we restrict our experiments to CoLA, MRPC, STS-B, and RTE.

Table 3 presents the results for the layer-wise analysis. According to fine-tuning results, tuning a single LayerNorm may be sufficient to achieve performance comparable to fine-tuning all LayerNorms. Furthermore, the middle-layer LayerNorms exhibit the best results across all layers, which can be attributed to the high transferability of the middle layers in BERT, corroborating previous findings on the concentration of task-specific features in these subsets of the network (Liu et al., 2019).

3 Analysis

In the previous section, we have shown that different modules of a transformer block can play as the *winning tickets*, since they all have the potential for transferring knowledge to the selected downstream tasks. Among different modules, LayerNorms have proven to be the most reliable in fine-tuning. In this section, we search for the reasons behind the effectiveness of these modules. To this end, we focus on the magnitude of every weight and how they change during full fine-tuning across all layers.

As a first step, in Figure 1, we visualize the distribution of weights for different BERT modules on RTE and STS-B (more tasks can be found in the Appendix). In general, the distribution of weights is similar across Feed-Forward and Multi-Head modules. Nevertheless, LayerNorms tend to have a bimodal distribution, with one of the modes having significantly higher magnitudes. The pattern is consistent across LayerNorms$_A$ and LayerNorms$_F$. We hypothesize that these high-magnitude weights are the reason behind the effectiveness of LayerNorms and, in what follows, check our hypothesis by restricting our experiments to only high-magnitude dimensions of LayerNorms.

[2]For STS-B, the number of classifier parameters is 769.

3.1 Outlier Tuning

Outliers are high-magnitude weights in LayerNorms appearing early in the pre-training process (Kovaleva et al., 2021). Transformer-based models perform significantly worse on downstream tasks when their outliers are disabled after the fine-tuning process (Kovaleva et al., 2021).

In this experiment, we choose outliers as the set of n weights whose values are farthest from the mean. Except for the outliers, all the parameters are frozen during fine-tuning. It should be considered that the specific dimensions where the outliers appear may not necessarily be the same across different layers.

Table 4 presents the performance of fine-tuned BERT in two different settings and for four different values of n: 4, 16, 64, 256. We also report the results for the corresponding sets of n randomly selected weights. As can be observed, outliers tuning leads to competitive performance on most target tasks, despite using less than 0.0056% of all the model parameters. Interestingly, tuning in the extremely constrained setting of $n = 4$ still outperforms the frozen model, sometimes by significant margins (e.g., on STS-B). Setting n to higher values gives the model more capacity, bringing about higher performance.

Overall, we can conclude that the high-magnitude weights in LayerNorms play an important role in the effectiveness of these modules in parameter-efficient fine-tuning.

4 Conclusions

In this work, we study the efficiency of different modules in the transformer block of BERT to transfer knowledge from the pre-trained model to various downstream tasks. Our experimental results demonstrate that, contrary to what was sug-

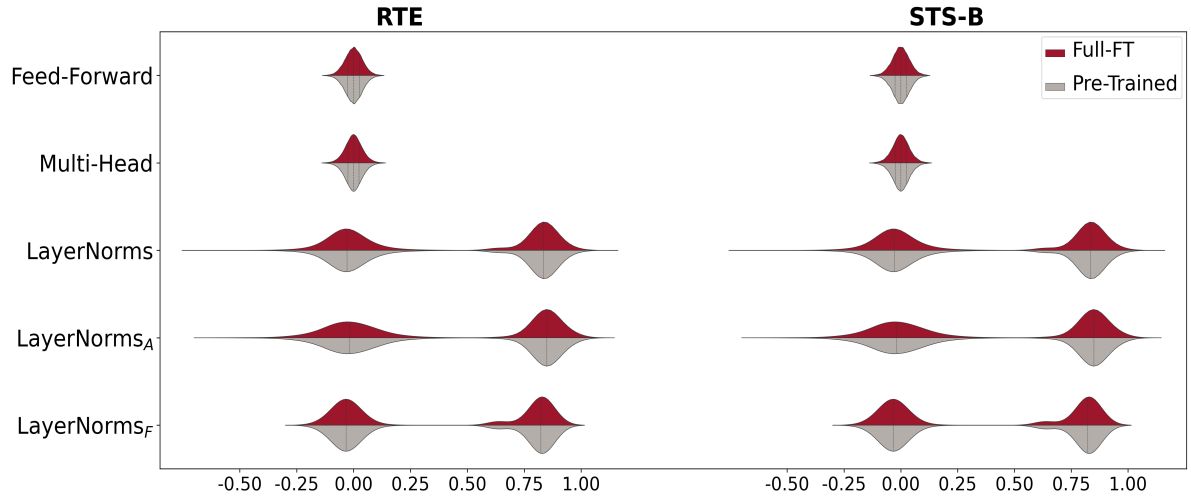

Figure 1: The empirical distribution of a random subset of weights in different modules of BERT. For better visualization, we have discarded outliers. The weights of LayerNorms appear to have a bimodal distribution with significantly higher overall averages and standard deviations.

n	%Param.	CoLA		MRPC		STS-B		RTE	
		Outlier	Random	Outlier	Random	Outlier	Random	Outlier	Random
256	0.0056%	55.9±0.5	54.3±1.6	90.3±0.7	89.4±0.8	88.1±0.1	87.9±0.2	68.8±0.4	67.8±1.2
64	0.0014%	51.6±0.6	48.3±2.8	87.5±0.7	88.1±0.7	86.5±0.2	86.5±0.2	63.5±0.3	64.8±1.5
16	0.0004%	47.0±1.4	40.8±4.5	85.3±0.3	85.0±0.2	85.3±0.1	84.7±0.1	64.1±0.8	63.1±1.1
4	0.0001%	39.5±0.8	36.7±4.9	83.5±0.1	82.6±0.2	83.8±0.2	80.0±0.2	61.4±0.5	59.3±0.7

Table 4: The performance of the fine-tuned BERT with n trainable parameters in every LayerNorm module on four different target tasks. Selecting the n parameters from the outliers leads to better performance in most cases, compared to the random selection. For $n = 256$, the results of outlier tuning are comparable with the Full-FT scenario.

gested by previous work, every module can be a *winning ticket*, achieving comparable performance to the full fine-tuning scenario. Among all modules, LayerNorms prove to be the most reliable for transferability with a limited number of trainable weights, such that tuning them in only one layer can be sufficient for attaining performance on a par with that of the full fine-tuning. We find that the weights in these modules have notably high magnitudes compared to other modules, which could be the reason for their effectiveness. We examine this hypothesis through outlier tuning (tuning only the n weights in a LayerNorm whose values are farthest from the mean), limiting the number of tunable parameters to a significantly small fraction.

Our results pave the way for better parameter-efficient fine-tuning of large language models without the need for costly algorithms to determine the optimum sub-network or introduce additional parameters for knowledge transfer.

5 Acknowledgment

We thank the anonymous reviewers for the constructive comments and suggestions that helped improve the paper. Sara Rajaee is funded in part by the Netherlands Organization for Scientific Research (NWO) under project number VI.C.192.080.

6 Limitations

We were subject to the constraints of computational resources; as a consequence, we reported results only for bert-base and chose the four smallest tasks of the GLUE benchmark in the tuning single norms (Section 2.4) as well as outliers tuning (Section 3.1). Obviously, the more trainable parameters a model has, the more accurate its results will be. Since our Outlier Tuning technique fine-tunes just a tiny portion of parameters, less than 0.006% of the model weights, there is an upper bound on its learning capability.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2015. Tensorflow: Large-scale machine learning on heterogeneous systems.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. 2020. Rigging the lottery: Making all tickets winners. In *Proceedings of Machine Learning and Systems 2020*, pages 471–481.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Trevor Gale, Erich Elsen, and Sara Hooker. 2019. The state of sparsity in deep neural networks. In *International Conference on Learning Representations*.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. 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 4884–4896, Online. Association for Computational Linguistics.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Edward Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, volume 34, pages 1022–1035. Curran Associates, Inc.

Olga Kovaleva, Saurabh Kulshreshtha, Anna Rogers, and Anna Rumshisky. 2021. BERT busters: Outlier dimensions that disrupt transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3392–3405, Online. Association for Computational Linguistics.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Jaeho Lee, Sejun Park, Sangwoo Mo, Sungsoo Ahn, and Jinwoo Shin. 2021. Layer-adaptive sparsity for the magnitude-based pruning. In *International Conference on Learning Representations*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Zi Lin, Jeremiah Liu, Zi Yang, Nan Hua, and Dan Roth. 2020. Pruning redundant mappings in transformer models via spectral-normalized identity prior. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 719–730, Online. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Houman Mehrafarin, Sara Rajaee, and Mohammad Taher Pilehvar. 2022. On the importance of data size in probing fine-tuned models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 228–238, Dublin, Ireland. Association for Computational Linguistics.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT Plays the Lottery, All Tickets Are Winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3208–3229, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. 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. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame,

Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yichu Zhou and Vivek Srikumar. 2021. DirectProbe: Studying representations without classifiers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5070–5083, Online. Association for Computational Linguistics.

Yichu Zhou and Vivek Srikumar. 2022. A closer look at how fine-tuning changes BERT. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1046–1061, Dublin, Ireland. Association for Computational Linguistics.

A Weight Distribution

Figure 2 demonstrates the distribution of weights for different BERT modules on MRPC and CoLA.

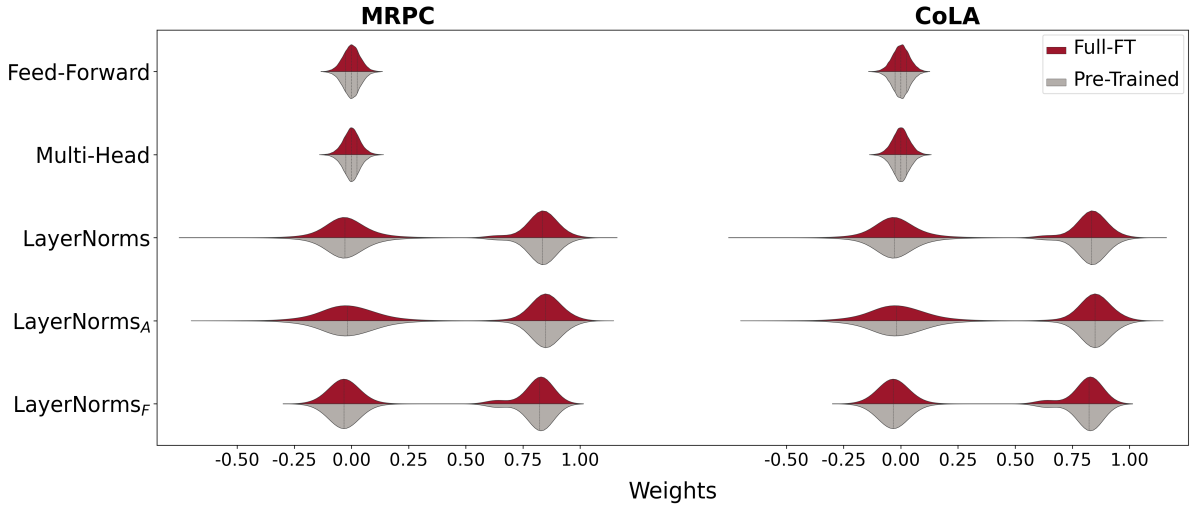

Figure 2: The empirical distribution of a random subset of weights in different modules of BERT on MRPC and CoLA.