Evolution of Representations in Cross-Lingual Fine-tuning

Aman Hussain
12667447
aman.hussain@student.uva.nl

Emil Dudev
12767492
emil.dudev@student.uva.nl

Abstract

Cross-lingual fine-tuning has been widely used to bridge the gap between high-resource & low-resource languages. In this paper, we study the evolution of the learned representations during cross-lingual fine-tuning. We fine-tune a pre-trained multi-lingual BERT on a small Dutch corpus. A BERT model, pre-trained on Dutch exclusively, is used as a comparative baseline. We show that our transferred multi-lingual BERT learns a different representation subspace than the native model. Additionally, we explore the loss in multi-lingual capacity during fine-tuning.

1 Introduction

The field of natural language processing has been riding high on the modern deep learning techniques. However, deep learning suffers from a *data dependence* problem. Consequently, progress in the field becomes dependent on the availability of language resources. Inevitably, this has led to a divide between high-resource languages and low-resource languages.

Transfer learning has been prescribed as a solution. It relaxes the independent and identically distributed (i.i.d.) requirement between training and test data. The idea is to first learn some general language representation using supervised or self-supervised methods and then *transfer* it to task-specific downstream systems. Word embeddings in Mikolov et al. (2013) first popularized this approach. Pre-trained language models in Peters et al. (2018); Howard and Ruder (2018); Devlin et al. (2019) have led the recent efforts in this front. A cross-lingual extension of pre-trained language modeling could be a way to bridge this gap between high and low resource languages.

Multilingual language models such as mBERT in Devlin et al. (2019) and XLM in Lample and Conneau (2019) have shown that it is possible to transfer cross-lingual information. They can be trained on the supervised data of multiple languages and made to perform zero-shot learning on a new language (Artetxe and Schwenk, 2018). Several studies (Wu and Dredze, 2019; Pires et al., 2019) have been conducted to understand why multi-lingual transfer learning works.

In this paper, we study how the pre-trained representations evolve during cross-lingual fine-tuning [1]. We fine-tune a pre-trained model with multi-lingual vocabulary - *transfer* model - to learn the *target* language Dutch. In the process, we investigate several phenomena such as the loss of multi-lingual capacity and differences of the *transferred* representations from that of a *native* model. This will lead us to several insights - some surprising, others expected - into the cross-lingual fine-tuning process. The key findings are listed below:

- Our *transferred* model learns a different represenation of the target language compared to the *native* model (Section 4.3).
- We observe a catastrophic loss in multi-lingual capacity of the *transferred* model (Section 4.1). We also show that there can be balance that preserves the multi-lingual capacity to some extent (Section 5.1).

2 Related Work

Transfer learning in NLP is still maturing. Attempts to fine-tune text classifiers in the past have either been unsuccessful or have required intricate training procedures, uncommon for transfer learning. In their recent work, Howard and Ruder (2018) propose several novel transfer-learning techniques for NLP tasks. However, their ULMFiT's recommended discriminative fine-tuning and slated triangular rates were evaluated on a 3-layer LSTM language model. We adapt ULMFiT's gradual unfreezing technique to train our transformer models rather than LSTM-based models.

Similarly, Eisenschlos et al. (2019) build upon previous research and adapt a fine-tuning approach

[1] fine-tuning and transfer learning are used interchangeably

employing sub-word tokenisation and cross-lingual bootstrapping. Our *transfer* model architecture already includes the sub-word tokenization method.

When analysing a model's inner working, it is common to look at the parameters' weights. Voita et al. (2019) examine the per-token representations of several layers in a transformer model. They answer pertinent questions, such as 'What does a layer represent?'. We are inspired to use a similar approach in the cross-lingual transfer learning context.

Recently K et al. (2020) conducted a comprehensive study of BERT's cross-lingual linguistic properties. They simplified the task by working with Bilingual BERT and intorducing a synthetic language. Moreover, they also looked into the model's architecture and how it influences the results of various NLP tasks when subjected to cross-lingual transfer.

3 Methodology

In this paper, we use the BERT architecture for our studies. Our *transfer* model is the pre-trained multilingual BERT from the HuggingFace library (Wolf et al., 2019). Our *native* model is pre-trained Dutch BERT (BERTje) by de Vries et al. (2019). In this way, we can compare the precise effect of the training regime on identical model architectures. Our corpus is obtained from Dutch Wikipedia articles and split into training, validation and test datasets.

3.1 Multi-lingual capacity during fine-tuning

We start by studying the evolution of BERT's multilingual capacity during fine-tuning. This will tell us if a fine-tuned multi-lingual model forgets other languages. We state the null hypothesis as follows: the perplexity of our transfer model on a multi-lingual corpus stays the same throughout fine-tuning. Since multilingual BERT was evaluated on the XNLI corpus (Conneau et al., 2018), we select the validation set of XNLI to measure the multilingual perplexity. The perplexity on our Dutch validation set is the control baseline to compare against. To keep the comparison fair, we reduced the number of sentences in the XNLI corpus to match our Dutch dataset.

3.2 Transfer learning techniques

We follow HuggingFace's recommendation for transfer-learning by choosing linear learning rate scheduling and the AdamW optimizer (Loshchilov and Hutter, 2019). However, we experiment with the layer-wise partial training approach. Only a few layers of the network are trained while the rest are frozen. For our purposes, we train the last $1, 3, 6, 12$ layers of BERT. The initial embedding layer of multi-lingual BERT is never trained.

We also experiment with a gradual layer freezing method. We unfreeze 4 layers consecutively from the end at every epoch for three epochs. The embedding layer is always frozen and the last pooler layer for classification is always trained. ULMFiT (Howard and Ruder, 2018) and MultiFit (Eisenschlos et al., 2019) use gradual layer unfreezing in their setup. We hypothesize that the gradual layer unfreezing technique might lead to the best results in terms of perplexity on the test set.

3.3 Attention comparison with native model

The last point we want to study is how do the inferred attention scores of a fine-tuned model compare to a native one. We hypothesize that after convergence, the fine-tuned model shall reach attention values close to the ones of the native model.

To determine how much does the fine-tuned model differ from the native one, we shall look at the mean Euclidean distance of the computed activations per layer. The choice of the L2 norm of the differences is dictated by the unbounded value range of the transformer heads, while the decision to look at the distances of each layer stems from our choice of a fine-tuning procedure (layer-based unfreezing of the weights). Moreover, the L2 norm shall be averaged across the input batch size and the sequence length. As the activation values are dependant on the input and are expected to have a relatively large range, we limit our experiment to only a single input batch of 16 sentences. Should we take the mean distance of thousands of input samples, we would expect the computed value to be 'averaged-out' and not representative of potential model divergence.

4 Experiments and Results

4.1 Multi-lingual capacity during fine-tuning

Table 1 shows the change in perplexity at the start of fine-tuning a multi-lingual BERT model. The perplexity on our Dutch validation set starts higher than the multi-lingual perplexity. This is the expected behaviour from any multi-lingual pretrained model. However, the perplexity on the multi-lingual set quickly blows-up and the Dutch

Fine-tuning steps	Dutch Perplexity	Multi-lingual Perplexity
#1	11.54 (±1.28)	06.08 (±0.33)
#501	07.76 (±0.86)	10.39 (±0.69)
#1501	05.56 (±0.56)	22.45 (±1.60)

Table 1: Change in multi-lingual perplexity during fine-tuning. The aggregated mean and s.d. are measured across three different runs.

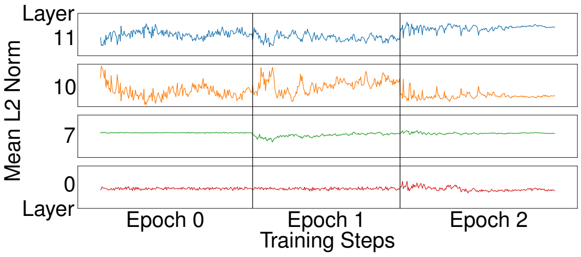

Figure 1: Layer activation comparison for a select group of layers. The y-axis spans a range of about 3.8. For these results, gradual unfreezing was used (Section 3.2. Data on all layers is available in Figure A.1)

perplexity plummets. We reject our null hypothesis based on this evidence. Our transfer model loses its multi-lingual capacity to focus on learning Dutch exclusively.

4.2 Transfer learning techniques

Table 2 compares the different transfer learning techniques in terms of the perplexity on our Dutch test set. For comparison, it also shows the test perplexity of our native model (BERTje). We can see that training all the layers of our transfer model at once led to the best test perplexity (for details refer to Section A.3). Going against our hypothesis, the gradual layer unfreezing procedure is the second-best.

4.3 Attention comparison with native model

In Figure 1 we present how the Euclidean distances of the hidden states of selected layers of the fine-tuned model differed from the ones of the native Dutch BERT. It is evident that as layers are unfrozen, their attention outputs slowly converge to a stable state. We interpret this as a sign that our fine-tuned Dutch model converges to a local weight optimum, which is different than the one of the native Dutch model. These results effectively disprove our hypothesis that a fine-tuned model shall align itself with a native one.

What is interesting, though, is that for some layers the distance between the models' activations stabilises at a relatively high value (layer 11), while for others, it stabilises at a low value (layer 10). This could mean that the linguistic features learned by the separate layers can be encoded in many different ways (or that fine-tuning causes different features to be learned). The exploration of this phenomena is left for future work.

The above results were obtained during the gradual unfreezing approach to fine-tuning (in groups of 4), but the results are not limited to only this scenario. We obtained similar results when training all layers at once (without modifying the embedding layer), and therefore draw the same conclusion (see Figure A.3).

One possible explanation for the divergence of the two models (and the rejection of our hypothesis) is that our fine-tuned model had overfit on our training data, whereas the native model is capable of generalisation. We look into this possibility in the next section.

5 Discussion

5.1 Multi-lingual capacity during fine-tuning

Some degree of performance loss on the original task is expected from transfer learning. However, our experiments show that fine-tuning a multi-lingual BERT leads to a substantial loss in multi-lingual capacity. On examining this trend at a more granular level, we find that there exists an optimum point which balances both capacities. In Figure 2, we can see the change in perplexity at every 50 steps of training. We recorded the measurements for three runs at different random seeds. At around 200th step of every run, both perplexity curves cross each other. This seems to be the spot which optimizes for both the multi-lingual and Dutch perplexity. Yet this might be a dataset-specific phenomenon and we should keep in mind that such a balancing act is not always required in the real world. Using cross-lingual transfer learning often implies that the user cares about performance on the target language at the expense of the base languages.

We must also point out the hidden assumption of our experiment. We assume that perplexity is the perfect measure of multi-lingual capacity or understanding of the Dutch language. It would be better to estimate multi-lingual capacity by evaluating on the cross-lingual sentence classification task the XNLI dataset is designed for. Given the compute

Models	BERT					BERTje
Fine-tuning	layer 0-12	layer 6-12	layer 9-12	layer 11-12	gradual unfreezing	pre-trained
Test Perplexity	**7.342**	9.045	14.159	16.110	8.950	23.547

Table 2: Test perplexity of different transfer learning techniques. Layer M-N indicate that only layers M to N were trained. BERTje was not fine-tuned at all. Pre-trained weights were used to yield a comparative baseline.

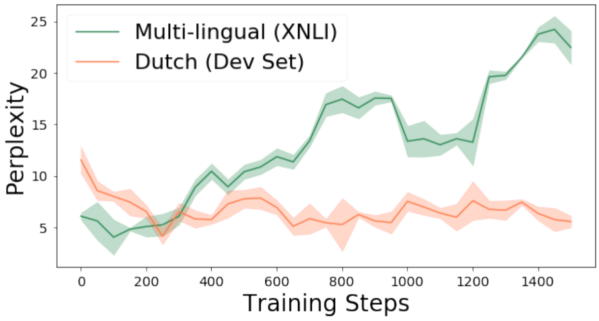

Figure 2: Change in multi-lingual and Dutch perplexity during fine-tuning at every 50 steps of fine-tuning. The error bands are composed using three different runs.

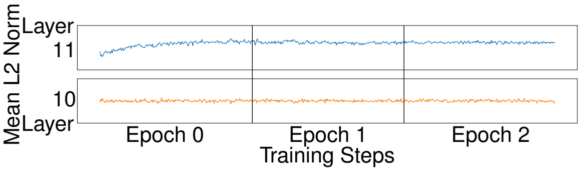

Figure 3: Layer activation comparison for the last 2 layers of a model trained with only 1 unfrozen layer (Section 3.2. The y-axis spans a range of about 3.8. Data on all layers is available in Figure A.2)

resources and time constraints, it was infeasible for us to do so during the training of a BERT-sized architecture.

5.2 Transfer learning techniques

Contrary to some transfer learning literature, fine-tuning the entire transfer model at once turned out to be the best technique for us. However, we cannot provide any theoretical justification for this behaviour. To the best of our knowledge, a theory of transfer learning has not been proposed to explain such phenomenon yet. The conventional wisdom is to empirically find out the technique that works best for the task-specific setup.

5.3 Attention comparison with native model

In Section 4.3 we hint at the possibility of our model overfitting. To determine whether this is the case, we conducted the same experiment having unfrozen only a single layer; results are shown in Figure 3.

We observe that the stable state of the last layer is reached early in the training process, without even having completed the first epoch. Moreover, the calculated activations are of sentences, which have not been seen by our trained model, making overfitting unlikely. This, along with the clear divergence of the fine-tuned model in comparison to the native one even in this scenario, lead us to believe that our hypothesis on convergence is in fact wrong. Lastly, we believe this is evidence that after being subjected to transfer-learning, a BERT model would stabilise to a feature representation which is different from the one obtained by a natively trained model. In future work, it remains to be seen how many such representations exist, or rather what are the common properties of such models and what ‘shortcuts’ can be employed when training, evaluating, or analysing them.

One shortcoming of our gradual-unfreezing technique is that we operate on groups of 4 layers at a time. This can have an impact both on the convergence time and the actual learned weights. Our decision to do so was dictated largely of practical reasons. We deemed in-feasible to conduct an over-a-day long transfer-learning experiment, even more so, as we would need to conduct several such.

6 Conclusion

We set out to explore BERT’s behaviour during and after transfer learning, to confirm simple assumptions. However, we discovered a catastrophic loss in multi-lingual capacity and found that BERT need not benefit from sophisticated fine-tuning methods. Furthermore, we hint at the existence of several good representation sub-spaces, which raises more questions regarding the models’ common properties. Future work can take a closer look at this finding and employ a probing classifier to gain more insight.

References

Mikel Artetxe and Holger Schwenk. 2018. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *CoRR*, abs/1812.10464.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Julian Eisenschlos, Sebastian Ruder, Piotr Czapla, Marcin Kadras, Sylvain Gugger, and Jeremy Howard. 2019. MultiFiT: Efficient multi-lingual language model fine-tuning. 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 5702–5707, Hong Kong, China. Association for Computational Linguistics.

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.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. 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 4396–4406, Hong Kong, China. Association for Computational Linguistics.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*.

Thomas Wolf, L Debut, V Sanh, J Chaumond, C Delangue, A Moi, P Cistac, T Rault, R Louf, M Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv, abs/1910.03771*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness 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 833–844, Hong Kong, China. Association for Computational Linguistics.

A Extra material

This appendix section contains explanatory notes, which did not make it in the main paper. We include details in relation to the dataset being used, the training procedure, as well as some discreptancies in the results we obtained.

A.1 Dataset size

Our training Dutch dataset consists of 80,000 sentences, whereas our validation and test sets each have 10,000 sentences. The XNLI set used also has 10,000 sentences randomly picked from the original XNLI validation set.

A.2 Multi-lingual capacity during fine-tuning

In this experiment, we fine-tune the entire pre-trained multi-lingual BERT for 3 epochs on our Dutch train set with a batch size of 8. At every 50 steps, we measure the perplexity on the Dutch validation set and on the XNLI validation set with twice (16) the batch size. We use a learning rate of 5×10^{-5} with linear scheduling as prescribed by the HuggingFace library. The trend was clear from the very beginning and hence only those results are shown in the paper.

A.3 Transfer learning techniques

For this experiment we run 5 training sessions where a different layer unfreezing scheme is followed. What remains same is the batch size of 8, learning rate of 5×10^{-5} with linear scheduling over 3 epochs. Early stopping is followed based on the best validation accuracy at every 100 steps. The first run fine-tunes all 12 layers, the second run fine-tunes only layers from 6 to 12, the third run fine-tunes only layers from 9 to 12, the fourth run fine-tunes only layers from 11 to 12 and the fifth run gradually unfreezes the all the layers as the training progresses. In the end we test all the five fine-tuned models on the Dutch test set. We also test the pre-trained BERTje on our Dutch test set to have a comparative baseline.

A.4 Attention values

We would like to mention a discreptancy in the results we obtained. The results we present of activation comparison of our fine-tuned BERT model to a native Dutch one show slight variability for all frozen layers. The results were obtained during training without applying backpropagation, and although a batch from our validation set was used (not present in the training set), the dropout sublayers were not disabled. However, we believe that this fault on our side does not invalidate the conclusions we draw from the obtained results, as the fluctuation is minimal and we interpret the trend at a larger scale. We also opted against smoothing the lines, as the observed variability may help readers draw other conclusions (perhaps unrelated to our research points).

The careful reader might also notice that we do not include any y-axis ticks or grid marking. We deemed this to be unimportant, as we report relative differences. However, we do make sure that the scale amongst the layers is preserved (otherwise the small fluctuations of the frozen layers would have been displayed as a very large variance). The y-axis spans a range of about 3.8, which in all figures is no more than 14% of the absolute value (observed with the highly fluctuating last layer).

In our paper we discussed the obtained average L2 norm of a select group of layers during two fine-tuning approaches. Namely, we looked at the data obtainied from gradual unfreezing, and having only a single unfrozen layer. The complete result set is available in Figures A.1 and A.2. We also add the results we obtained when having unfrozen all layers (Figure A.3), and having unfrozen only half of the layers (Figure A.4).

A.5 Environment setup

As we used Google Colab[2] for our experiments, we ran into issues with it being unable to efficiently read and store large amounts of files. Specifically, even though we used mounted a Google Drive partition, Colab required all accessed files to be copied on disk[3]. Since the disk of the virtual machines is restricted (to about 30-40 GiB), this made the generation and saving of many files impossible. All our output files total at about 2 TiB. To deal with this, we invested a significant amount of time in developing a custom file syncrhonisation scheme, which was able to deal with Colab's limitation.

[2] https://colab.research.google.com/

[3] https://github.com/googlecolab/colabtools/issues/960

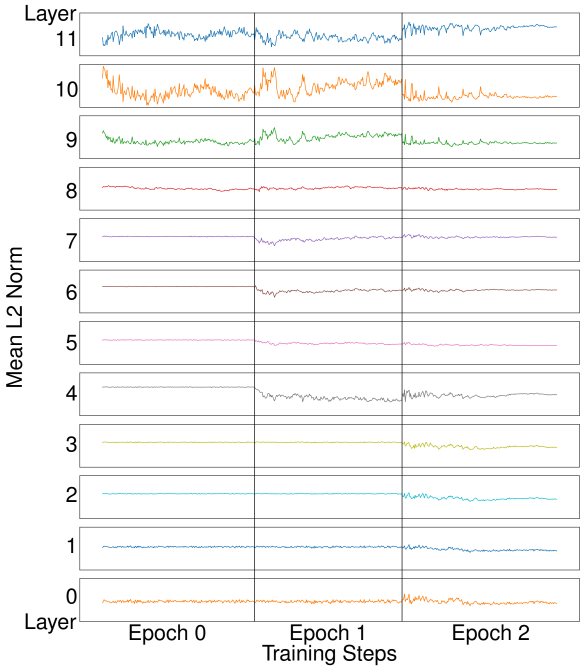

Figure A.1: Layer activation comparison during a gradual-unfreezing fine-tuning approach.

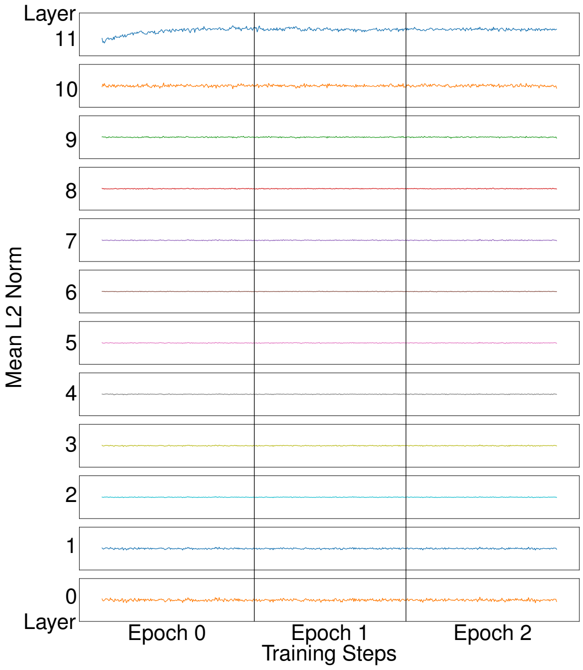

Figure A.2: Layer activation comparison when fine-tuning with only the last layer being unfrozen

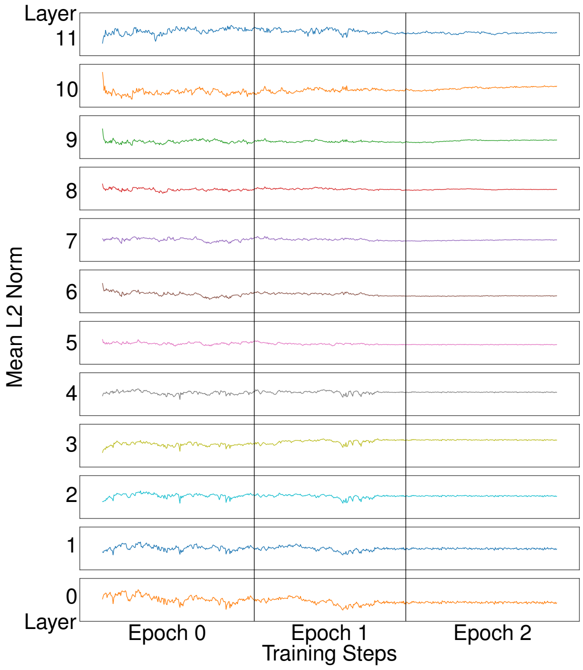

Figure A.3: Layer activation comparison when fine-tuning with all layers being unfrozen.

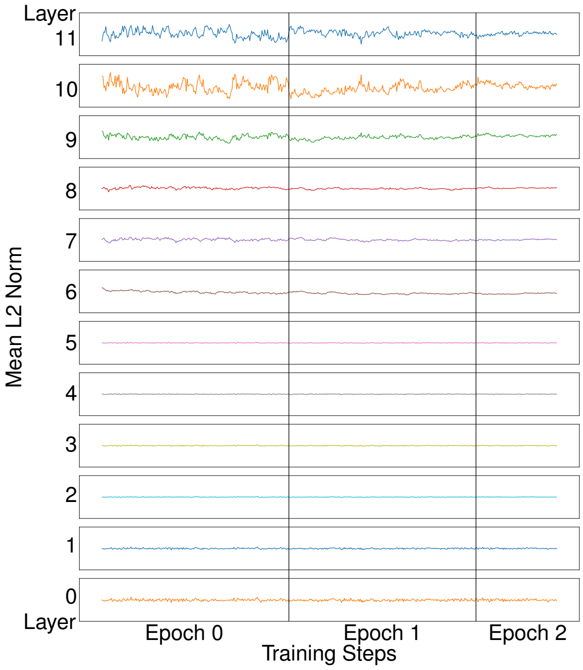

Figure A.4: Layer activation comparison when fine-tuning with only the last 6 layers being unfrozen