Iran University of Science and Technology
Deep Learning (Spring 2020)

The Convexity of BERT: From Cause to Solution

Hosein Mohebbi*
Department of Computer Engineering
Iran University of Science and Technology
hosein_mohebbi@comp.iust.ac.ir

S. M. Ali Modarressi*
Department of Computer Engineering
Iran University of Science and Technology
m_modarressi@comp.iust.ac.ir

Abstract

Many research has been done on probing and analyzing how BERT works and what are its characteristics. A similar feature within the layerwise results is a convexity in performance, which means the last layers in BERT are showing degradation in probing results. We investigate this fact remains in fine-tuned models by conducting probing tasks. The main goal of this work is to demonstrate the main cause of this point, which is BERT's MLM pre-training, using a novel MLM head probing method. In addition, we try out some proposed methods and discuss on what cases BERT could work more efficiently.

1 Introduction

Since two years ago, Transformer-based Pre-trained Language Models (PLMs) shown state-of-the-art and competitive performances on a variety of natural language processing (NLP) tasks. Using a self-attention architecture causes a higher level of parallelization than recurrent neural networks (RNNs) and also a larger span for context modelling.

One of the best-known models, which considers the input's context in a bidirectional approach, is BERT (Devlin et al., 2018) which is pre-trained using masked language modelling (MLM) and next sentence prediction (NSP). By fine-tuning for various NLP tasks such as sequence classification or word/token classification, BERT shows state-of-the-art performances.

Recent studies use different methods for analyzing these models, which one of the most used techniques is using "probing tasks" that are linguistic probes capable of capturing or analyzing the model-under-test from a certain linguistic point of view. For example, Hewitt and Manning (2019) were able to compose a probe capable of extracting syntax knowledge from the output representations.

One similar result shown within these research findings is having a convexity in performance within the encoder layers. Probes usually show their best performance at some layers before the final layer and decrease afterwards. We use SentEval (Conneau and Kiela, 2018) probing tasks on fine-tuned BERT models to see whether this convexity remains or not, and after that, we proceed to our main goal which is finding the primary source of this fact in BERT. Also, we try to test out some proposed methods to see that if we can handle this convexity or not.

*Equal Contribution

(LateX template borrowed from NeurIPS 2019 and COLING 2020)

2 Datasets

2.1 GLUE Benchmark

The General Language Understanding Evaluation (**GLUE**) benchmark consists of nine datasets for natural language understanding evaluation Wang et al. (2018).

MNLI In the Multi-Genre Natural Language Inference dataset, each example is a pair of sentences for a three-class–entailment, neutral or contradiction- inference classification task Williams et al. (2017).

QQP The Quora Question Pairs is a paraphrasing task for classifying whether a question pair are semantically the same or not Chen et al. (2018).

QNLI Question Natural Language Inference is a binary classification adaptation Wang et al. (2018) from the SQUAD question answering task Rajpurkar et al. (2016). Each example contains a question/answer pair which in case the answer is correct; the example is positive. Otherwise, the example is negative.

SST-2 Stanford Sentiment Treebank dataset includes examples of movie reviews labeled based on their sentiment Socher et al. (2013).

CoLA Corpus of Linguistic Acceptability is a task to determine whether a sentence is grammatically acceptable or not Warstadt et al. (2019).

STS-B Semantic Textual Similarity Benchmark consists of sentence pairs each annotated by their semantic similarity with ranging from 1 to 5 Cer et al. (2017).

MRPC Microsoft Research Paraphrase Corpus is a paraphrasing binary classification task in which every sentence pair labeled by their semantic equivalency Dolan and Brockett (2005).

RTE Recognizing Textual Entailment is another inference task which specifying a pair of sentences decides whether the second sentence is an entailment regarding the first one or not Bentivogli et al. (2009). In contrast with MNLI, training examples are much fewer.

WNLI Winograd Natural Language Inference is a similar task to RTE with a fewer amount of training examples Levesque et al. (2012).

2.2 SentEval

The SentEval toolkit Conneau and Kiela (2018) based on the probing dataset created by Conneau et al. (2018). This toolkit includes a set of probing tasks that are cast in the form of single-sentence classification tasks to measure what linguistic properties are captured in sentence representations. These probing tasks are categorized into Surface, Syntactic and Semantic information.

The Surface category includes a task **(SentLen)** to predict the length of the sentences among six classes which are length bins following Adi et al. (2016) and another task Word Content **(WC)** to predict which of the 1k target words exists in the given input sentence.

In the Syntactic category, the tree depth **(TreeDepth)** task asks for the maximum depth of the constituency parse tree of the sentence with ranging from 5 to 12, the Top Constituent **(TopConst)** task to identify the sequence of constituents in the syntax tree among 20 classes and the Bigram Shift **(BShift)** task examine sensitivity to sentences that their two random adjacent words have been inverted.

The Semantic category encompasses tasks for the unveiling of the semantic properties captured by sentence representations such as past present **(Tense)** check for the ability to classify the tense of the main-clause verb of the sentence among three classes, the subject number **(SubjNum)** and the object number **(ObjNum)** tasks to identify the number of the subject and direct object respectively in the main clause(singular or plural), the Semantic Odd Man Out **(SOMO)** tests for the sensitivity to random replacement of a word with another word of the same part of speech, and coordination inversion **(CoordInv)** to detect whether the order of the clauses is inverted in sentences with two coordinate clauses.

Each task contains 100k examples in the training set and 10k examples in validation and test sets that are balanced among its target classes.

2.3 SQuAD

The Stanford Question Answering Dataset (SQuAD v1.1) Rajpurkar et al. (2016) is a reading comprehension dataset, consisting of question/answer pairs posed by crowd-workers on 536 Wikipedia articles.

3 Related work/Background

Tenney et al. (2019a) experimented based on the "edge probing" approach of Tenney et al. (2019b) to measure how the different layers of the BERT network can resolve syntactic and semantic structure within a sentence. They used two complementary measurements: scaler mixing weights and cumulative scoring, and observed that a consistent ordering emerges: POS tags captured earliest, followed by constituents, dependencies, semantic roles, and coreference. That is, the basic syntactic information appears earlier in the model, while high-level semantic information appears at higher layers. Moreover, they observed that in general, syntactic information is more localizable, with weights related to syntactic tasks tending to be concentrated on a few layers while information related to semantic tasks is generally spread across the entire network. This finding is consistent with observations by Jawahar et al. (2019) with regrads to semantic probes, which only one SentEval semantic probe in their study topped at the last layer, and the others peaked around the middle and then considerably deteriorated by the final layers.

According to study that conducted by Ethayarajh (2019), in BERT, words in the same sentence are more dissimilar to one another in upper layers. As word representations in a sentence become more context-specific in upper layers, they drift away from one another, except the last layer. Liu et al. (2019) also employed a set of sixteen diverse English probing tasks and concluded that the middle layers of Transformers are overall the best-performing and the most transferable across tasks rather than exhibit a monotonic increase.

To overcome these convexity of BERT performance regards to the layers, several studies explored the possibilities of improving the fine-tuning of BERT. Su and Cheng (2019) aimed to find high-quality attention output layers and then extract information from aspects in all layers by using a weighted representation of them through Squeeze and Excitation approach introduced by Hu et al. (2018).

4 Proposed methods

4.1 Probing Fine-tuned BERT Models

In order to expose the effect of fine-tuning of BERT on probing tasks, in contrast to the probing setup introduced by Jawahar et al. (2019), which evaluates each layer of pre-trained BERT, we specifically focused on revealing the probing task performance of each BERT layer after fine-tuning. To do so, we first fine-tuned BERT(Base, Uncased) on each of nine different GLUE tasks, then evaluated their BERT layers separately through ten probing datasets created by Conneau et al. (2018). All fine-tuning has been done with recommended hyperparameters by Devlin et al. (2018), with batch size {16, 32}, learning rate (Adam){5e-5, 3e-5, 2e-5} and max epochs of 5. In each fine-tuning, we chose a model that has the most validation performance score while being higher than the accuracy performance score at the same time. To alleviate instability in fine-tuning, we repeated fine-tuning BERT(Base, uncased) and training and evaluating probing classifiers three times per each GLUE task.

4.2 Results

Figure 1 shows the differences in probe score of fine-tuned and pre-trained of each layer of BERT(Base, uncased). Green color means positive delta that indicates improving probe score during fine-tuning. We observed that some probing tasks such as BShift, Tense, and SOMO are correlated with CoLA in GLUE tasks. Fine-tuning in GLUE tasks leads to a decrease in the performance score of the SentLen probe except for question-based tasks like QQP and QNLI. This may explain the impact of the question marks in these tasks that help the model to learn the length of the sentence better. Overall, depending on the GLUE tasks, probe scores may increase or decrease. On the other hand, even by fine-tuning a BERT model, some characteristics will remain the same. In Figure 2, Instead of having a monotonical increase in the probing performance, nearly every

fine-tuned and pre-trained probing, we can observe a small but considerable amount of decline in the resulted performance. According to figure 2 in almost every probing task (except for Tree Depth and ObjNum), the performance heatmap shows a slight decrease after before reaching the final layer (layer 12).

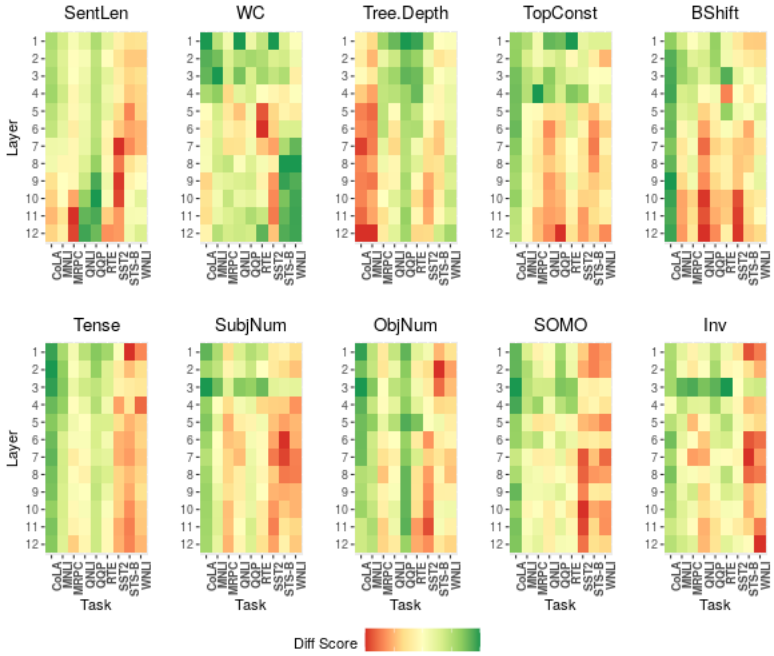

Figure 1: The difference probe scores between fine-tuned and pre-trained model among different GLUE tasks

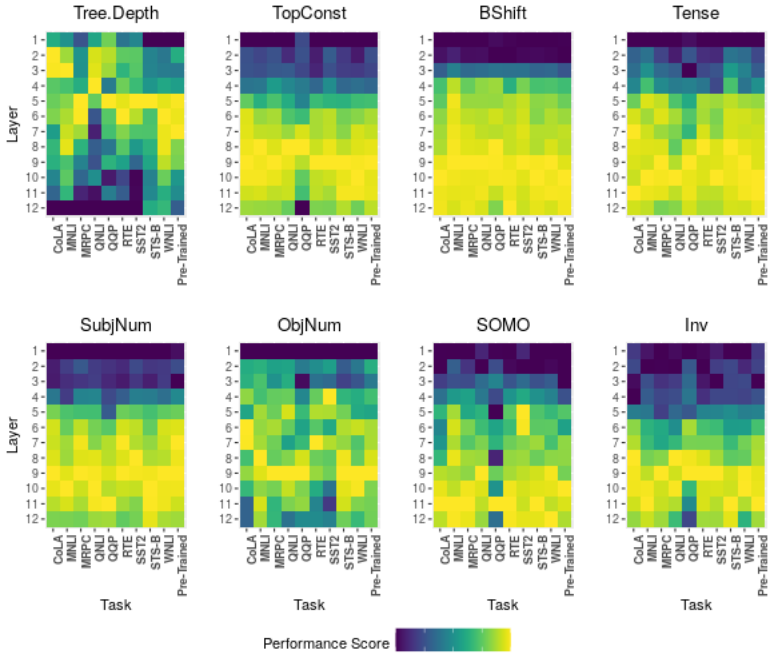

Figure 2: The probe scores of fine-tuned models on different GLUE tasks

4.3 MLM pre-Training Effect

As we have shown and discussed in the previous section, probing tasks shows some level of concavity in the probing performance results. This means BERT's final layer does not show the highest performance. In prior studies, which provide layer-based probing results show similar characteristics. Using measures of contextuality Ethayarajh Ethayarajh (2019) applied a contextuality analysis on each layer of BERT, GPT-2 Radford et al. (2019), and ELMO Peters et al. (2018a), which shows the penultimate layer has the extreme value in all of its self-defined contextual metrics. Liu et al., Liu et al. (2019) defined sixteen different probing tasks in which the results also express the same phenomenon in both BERT-Base and BERT-Large pre-trained models. Other layerwise probing studies such as Tenney et al. (2019a); Peters et al. (2018b) can provide further indications of such an argument.

Our next research question arises from all the above evidence: Why does BERT not show a monotonic performance in probing tasks and, in contrast, has a concave performance result? To discuss the reason, we take a step back to how BERT's contextual representations are trained.

4.3.1 An interpretation of BERT's MLM pre-training objective

One of the main pre-training objectives in BERT is the Masked Language Model (MLM)Devlin et al. (2018). This task objective is to predict randomly masked tokens from the input using an MLM Head over the final output word representations.

As we have shown the MLM Head structure in figure 3, the head consists of a Transformation layer, matrix multiplication with the input embeddings weights, a bias addition, and a final softmax layer. After the output representations are transformed using a fully connected dense layer and a layer normalization layer, the following output is multiplied with the input embedding weights in a matrix multiplication layer and added with a bias vector with the same size of the vocabulary size in the model and subsequently will be passed to softmax function to produce the probability of the predicting mask token.

Since the MLM head is only using a single word output representations to infer its original word, the final representation cannot be fully contextualized. In other words, the final output word representations should be non-linearly similar to the initial input word embedding weights (referred to layer 0 in some literature), which are a form of word-based embeddings. The non-linear similarity is transformed using the transform layer. Another view of this situation is similar to an Autoencoder architecture Rumelhart et al. (1986), except that the bottleneck type here is not a matter of hidden layer size. Since the last layer's task is to transform highly contextualized embeddings to a lower level of contextualization to feed the MLM head, so the bottleneck will be based on the level of contextualization.

We show that the similarity to the initial embeddings reduces as we move towards the final layer, except for the final layer itself, which implies a higher similarity to a non-contextualized embedding than its 2 or 3 previous layers.

4.3.2 MLM Head Probing Setup

To compute the similarity between each layer word representations with the initial embeddings, we use an individual MLM head for each layer to demonstrate that if a layer could be transformed using a single transform layer in the head to the low-level initial embeddings, so it is non-linearly similar to a low-level embedding. We use the same head implementation in BERT's MLM pre-training task. Since the head uses the input embeddings weights and we want to observe the pre-trained model characteristics, we freeze the pre-trained model, so the only trainable weights in the head are the dense and bias layers with the trainable parameters in the LayerNorm layer.

Since the lower layers are closer to the initial embeddings, if we train these heads with unmasked data, the MLM head will only learn exact similarities without considering transformation. And on the other hand, if training is applied with masking token, i.e. [MASK], these layers cannot predict the original word since the layers are not contextualized enough to generate an embedding similar to the original one.

To handle this problem, we used an embedding dropout mechanism introduced by Zhou et al. (2019). This method partially masks the input word embeddings which makes this probing task easier for the

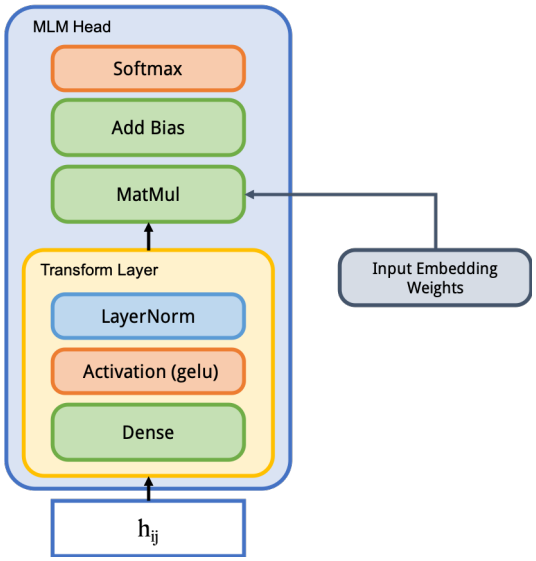

Figure 3: Architecture of an MLM head. This head computes the similarity between the output representation and the input embeddings with an extra feed forward layer

lower-level layers but still due to the existing dropout, the outputted representation is not entirely similar to the initial embedding and obliges the transform layer in the MLM head to learn more non-linear similarities.

This dropout mechanism applies during training on the SQUAD (Rajpurkar et al., 2016) passages as a text corpus. The passages in this dataset are gathered from Wikipedia articles which is the text corpus used for pre-training BERT (Devlin et al., 2018). Using this masking technique, we train an MLM head for each layer. For evaluation, tokens will not be masked since we want to measure the similarity between the input tokens and MLM Head outputs using the accuracy of the predictions.

In training these heads, two hyperparameters do matter the most: Masking Ratio and Dropout Ratio. The first one determines the probability of token being masked. If the masking ratio drops below the optimum point, fewer tokens will be masked at each epoch, which means slower convergence, and a high masking rate will cause low performance in token predictions. We've set the masking ratio equal to 0.15, as same as it is in the pre-training configuration. The dropout ratio is the boundary which controls how much do we want the model to know about the input tokens. Zero dropout means no masking and full dropout (Dropout Ratio = 1), means masking the whole embedding, much alike using '[MASK]' token. We train several tests with different masking ratios and dropout ratios.

The dropout ratio is the boundary which controls how much do we want the model to know about the input tokens, zero dropout means no masking and full dropout (Dropout Ratio = 1), means masking the whole embedding, much alike using '[MASK]' token.

4.3.3 Results

After training the MLM heads with the optimal ratios, the resulted Figure 4 shows a drop in prediction performance at the penultimate layer and a recovery for the last. Lower prediction performance implies the fact that the 11th layer representations have the lowest similarity to the input static embeddings which conclude to higher contextualism. Another point is that the last layer's representations show a smaller amount of contextualism by the fact that it has a higher similarity to the input. This probing result shows that the concept of convexity is based on the MLM Pre-training.

4.4 Proposed Methods to utilize Maximum Contextualism

Now, as we have seen convexity appears in the penultimate layer, we proposed three implementations in an effort to handle this fact during fine-tuning.

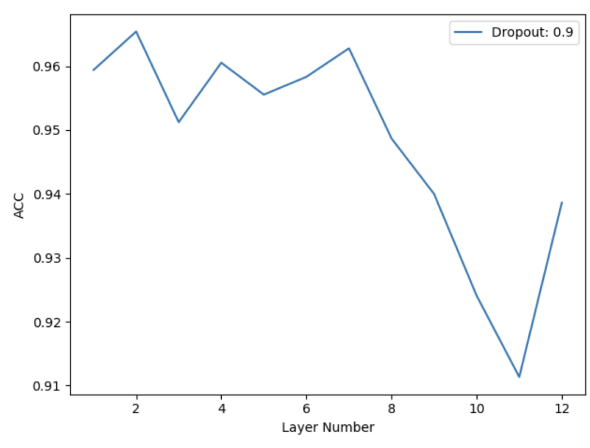

Figure 4: MLM head probing results for each layer shows degradation in performance towards the top layer. Except the top layer which shows higher similarity to the initial static embeddings

4.4.1 Fine-Tuning + MLM Task

We used the sense that if the last layer is going to have a decontextualization role, let's use another layer for that role. As shown in Figure 5, an extra encoder layer is added to the fine-tuned model. This encoder will be trained alongside with the classifier over the [CLS] pooler.

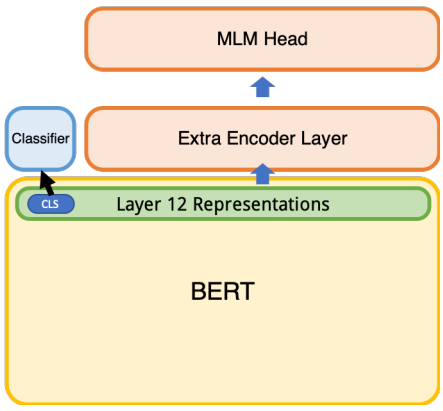

Figure 5: Fine-tuning the downstream task and MLM task simultaneously using an extra encoder and a classifier over BERT

To fine-tune with this method, we manipulate the training data. In each batch, we randomly select some samples and randomly select some tokens for the dropout masking. A new input called mask-id refers to the tokens which are going to be masked by the model. Either the value could be an ignore label which is ignored by the loss function, or the token id itself, which means the token should be masked and the label is the token id. Other samples that are not having masked tokens are used for the main classification task. By using this mixture of samples with two type of tasks, both encoder and classifier will be trained. Theoretically, in this case, the new encoder will be the decontextualizing layer, and the penultimate layer should be the layer with the most contextualism. At inference time, there is no need for the encoder layer so that it can be removed after training.

4.4.2 Copying Layers

Another concept is to copy one layer weights with high-level interpretation and initialize the layers above as shown in Figure 6 (a). This idea comes from that if a layer can build a higher level of contextualism, so it may be replicated to do the same job for one or two other layers instead of using

Model	CoLA	MRPC	STS-B	RTE	Average
BERT(base, uncased)	58.73	85.29	87.37	**72**	**75.84**
Multi-objective fine-tuning	59.83	83.33	**89.86**	70.31	75.83
Multi-objective + extra Layer fine-tuning	**60.6**	84.8	89.59	66.43	75.35
Coping last layers fine-tuning	58.8	**86.76**	88.33	63.18	74.26

Table 1: Results on GLUE dev sets. CoLA is evaluated using Matthew's Correlation. STS-B is evaluated using Pearson's correlation coefficient. MRPC and RTE are evaluated using accuracy.

Model	CoLA	MRPC	STS-B	RTE	Average
BERT(base, cased)	**58.71**	84.64	**87.9**	**71.48**	**75.68**
Dropping last two layers fine-tuning	55.27	**85.05**	85.84	67.15	73.32

Table 2: Results on GLUE dev sets. CoLA is evaluated using Matthew's Correlation. STS-B is evaluated using Pearson's correlation coefficient. MRPC and RTE are evaluated using accuracy.

the decontextualizing ones. We tried with different layers as source layers, and the most effective layer was the 10th layer.

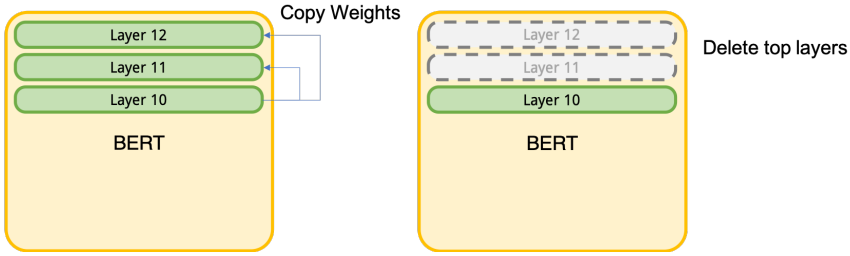

Figure 6: (a): Copying weights from the optimal layer and reinitialize the layers above, (b): Dropping top layers

4.4.3 Dropping Layers

Similar to the copying method, instead of copying layers to the above ones, we just drop a number of top layers (Figure 6 (b)). Here we assume that the top layers responsible for decontextualization should only be present during pre-training.

4.4.4 Results

We fine-tune all three proposed methods on four small GLUE tasks in an effort to find optimal hyperparameters more efficiently. The results in Table 1 and 2 shows the suggested method results and the BERT-base scores as a baseline. We could infer that some tasks do benefit from the applied methods while others do not. This task dependency is solved with some novel techniques which are discussed in the following discussion section.

5 Experimental Setting

All of our implementations are in TensorFlow using the HuggingFace Transformers library. Due to GPU memory and computation limit, we use BERT-base models as initial pre-trained models, and all wordpiece tokenizations are padded or trimmed to 128 maximum tokens length. Similar to the fine-tuning configuration specified by Devlin et al. (2018) we use Adam optimizer (Kingma and

Ba, 2014) with a learning rate which is selected from 2e-5, 3e-5, 5e-5. For fine-tuning, we used Google Colab tool because it provides a higher amount of GPU memory for free. In the SentEval probing, the auxiliary classifiers are the only trainable element, and the MLM heads are also not so memory-expensive at training, so we ran these probings on a workstation equipped with a 1070Ti GTX GPU.

6 Discussion

In this work, we used SentEval probing tasks on fine-tuned BERT models to show that the performance convexity of layers in BERT abides even after fine-tuning. Then, we proposed a new diagnostic MLM head with a dropout embedding approach to illustrate why BERT shows a convex performance in different tasks. Finally, we proposed some new methods of fine-tuning to examine whether we can handle this convexity. Our results show that adding MLM objective with its own special layer to the downstream task objective or replaced De-contextualized layers with contextualized layers or totally removing them does not mitigate the convexity of performance during the fine-tuning. Recently, Zhou et al. (2020); Liu et al. (2020) introduced novel adaptive inference methods by attaching couples an internal-classifier to each layer of a PLM to dynamically stop inference. To do so, they implicitly overcame over BERT's performance convexity as well as inference latency.

References

Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., and Goldberg, Y. (2016). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Bentivogli, L., Clark, P., Dagan, I., and Giampiccolo, D. (2009). The fifth pascal recognizing textual entailment challenge. In *TAC*.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Chen, Z., Zhang, H., Zhang, X., and Zhao, L. (2018). Quora question pairs.

Conneau, A. and Kiela, D. (2018). Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*.

Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word 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 4129–4138.

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jawahar, G., Sagot, B., and Seddah, D. (2019). What does bert learn about the structure of language?

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Levesque, H., Davis, E., and Morgenstern, L. (2012). The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M., and Smith, N. A. (2019). Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Deng, H., and Ju, Q. (2020). Fastbert: a self-distilling bert with adaptive inference time. *arXiv preprint arXiv:2004.02178*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018a). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Peters, M. E., Neumann, M., Zettlemoyer, L., and Yih, W.-t. (2018b). Dissecting contextual word embeddings: Architecture and representation. *arXiv preprint arXiv:1808.08949*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (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.

Su, T.-C. and Cheng, H.-C. (2019). Sesamebert: Attention for anywhere. *arXiv preprint arXiv:1910.03176*.

Tenney, I., Das, D., and Pavlick, E. (2019a). Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S., Das, D., and Pavlick, E. (2019b). What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Warstadt, A., Singh, A., and Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Williams, A., Nangia, N., and Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Zhou, W., Ge, T., Xu, K., Wei, F., and Zhou, M. (2019). Bert-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373.

Zhou, W., Xu, C., Ge, T., McAuley, J., Xu, K., and Wei, F. (2020). Bert loses patience: Fast and robust inference with early exit. *arXiv preprint arXiv:2006.04152*.