Fine-tuning BERT for Multi-task Learning

Stanford CS224N Default Project

Yutai Luo
Stanford Center for Professional Development
yutailuo@stanford.edu

Abstract

BERT (Bidirectional Encoder Representations from Transformers) has demonstrated outstanding performance across a wide range of language understanding tasks. When applying BERT to downstream tasks, fine-tuning is a commonly used approach to further improve model performance. Traditionally, a separate model is fine-tuned for each task. However, this approach leads to a substantial increase in the number of parameters and training time. In this paper, we explored learning approaches that utilize a single shared BERT model for multiple tasks. We focus on three key NLP tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. Our techniques included data preprocessing, reducing multi-task gradient interference, task sampling, and task-specific projected attention layers. We found that the Gradient Vaccine technique delivered the best results, achieving an overall accuracy of 0.787 on the test dataset, with 0.532 on the SST dataset, 0.893 on the Quora dataset, and 0.875 on the STS dataset, which significantly outperformed the baseline model trained with naive multi-task learning.

1 Key Information to include

- Mentor: **Neil Nie**
- External Collaborators: **No**
- Sharing project: **No**

2 Introduction

BERT (Bidirectional Encoder Representations from Transformers) has demonstrated outstanding performance across a wide range of language understanding tasks. However, adapting a single pre-trained BERT model to multiple different downstream tasks remains an active research area. Traditional methods involve fine-tuning a distinct model for each task, leading to a significant increase in parameters and training time. In this paper, we explored learning approaches that utilize a single shared BERT model for multiple NLP tasks. In contrast to some multi-task learning approaches where a universal model applied to different tasks, we use an alternative architecture where tasks share the majority of their parameters but also have a minor subset of task-specific parameters for customized adaptation. This flexible method can also handle scenarios where input and output spaces differ across multiple tasks.

Multi-task learning often encounters the issue of gradient interference, where gradients from different tasks can counteract each other. To mitigate this issue, we employed techniques like Gradient Surgery Yu et al. (2020) and Gradient Vaccine Wang et al. (2020), which help to manage and align gradients more effectively. We also explored projected gradient layers Stickland and Murray (2019) to introduce task-specific parameters, ensuring that each task receives tailored adaptation. Another challenge arises from varying dataset sizes across tasks. Because the Quora dataset is significantly larger than others, we observed the conventional round-robin task sampling resulted in overfitting for smaller datasets. To address this challenge, we experimented with various task sampling strategies, including annealed sampling, which adjusts the sampling probabilities based on dataset sizes.

Among these efforts, the Gradient Vaccine technique yielded the best results, achieving an overall accuracy of 0.787 on the test dataset, with 0.532 on the SST dataset, 0.893 on the Quora dataset, and 0.875 on the STS dataset, significantly outperforming the baseline model trained with naive multi-task learning.

The following sections are organized as follows: Section 3 reviews related work, and Section 4 describes model architecture and our methodology. Section 6 gives detailed experiment result, followed by concrete analysis in Section 7. The last two sections conclude our findings and discuss ethical concerns with regards to our work.

3 Related Work

Gradient Projection Yu et al. (2020) introduced a technique called Gradient Surgery (PCGrad), which projects a task's gradient onto the normal plane of the gradient of any other task with a conflicting gradient. However, Wang et al. (2020) discovered that PCGrad is ineffective when dealing with positive gradient similarities. To address this, they proposed a new method called Gradient Vaccine, which offers improved gradient projection in such scenarios.

Task-specific Parameters Stickland and Murray (2019) introduced Projected Attention Layers (PALs) as an innovative approach to enhance BERT's multi-task learning capabilities. PALs add small and task-specific layers to improve the model's ability to adapt to various tasks. This method allows the model to better handle the unique requirements of each task without significantly increasing its size.

Multi-task Scheduling A basic method for training a model on multiple tasks involves selecting a batch of training examples from each task in a fixed sequence, known as round-robin sampling. However, this method can be ineffective if the tasks have different numbers of training examples. A more sophisticated approach, as used by Stickland and Murray (2019), involves sampling training data based on dataset sizes, allocating more training budget to larger datasets. This approach helps balance the training process, ensuring that each dataset is appropriately represented.

4 Approach

4.1 Model Architecture

To introduce task specific parameters, we extend the BERT model by adding projected attention layers (PALs) in parallel with self-attention layers. Prior to this extension, each BERT layer can be represented as:

$$BL(h) = LN(h + SA(h)) \tag{1}$$

where $SA(\cdot)$ represents a self-attention sublayer:

$$SA(h) = FFN(LN(h + MH(h))) \tag{2}$$

FFN is a standard feed-forward network with $2dd_{ff}$ parameters, so there are total $d^2 + 2dd_{ff}$ parameters from a BERT layer. With $d = d_m$ and $d_{ff} = 4d_m$, a BERT layer has $12d_m^2$ parameters. The expression of a BERT layer combined with PALs is as follows:

$$BL_{PAL}(h) = LN(h + SA(h) + TS(h)) \tag{3}$$

The TS for a PAL is given as:

$$TS(h) = V^D MH(V^E h) \tag{4}$$

where V^E is a $d_s * d_m$ encoder matrix, V^D is a $d_m * d_s$ decoder matrix with $d_s < d_m$. Considering there are 3 NLP tasks, we require 3 PALs for each BERT layer, introducing $3 * (2d_m d_s + 12 * 3d_s^2)$

parameters. Additionally, we add a pooling layer applied to the final hidden state of the [CLS] token for each task, which adds $3 * d_m^2$ parameters. Therefore, we need $3 * d_m^2 + 6d_m d_s + 108d_s^2$ extra parameters. In our model, with $d_s = 204$ and $d_m = 768$, this amounts to approximately 72M parameters.

The overall architecture of our model is shown in Figure 1:

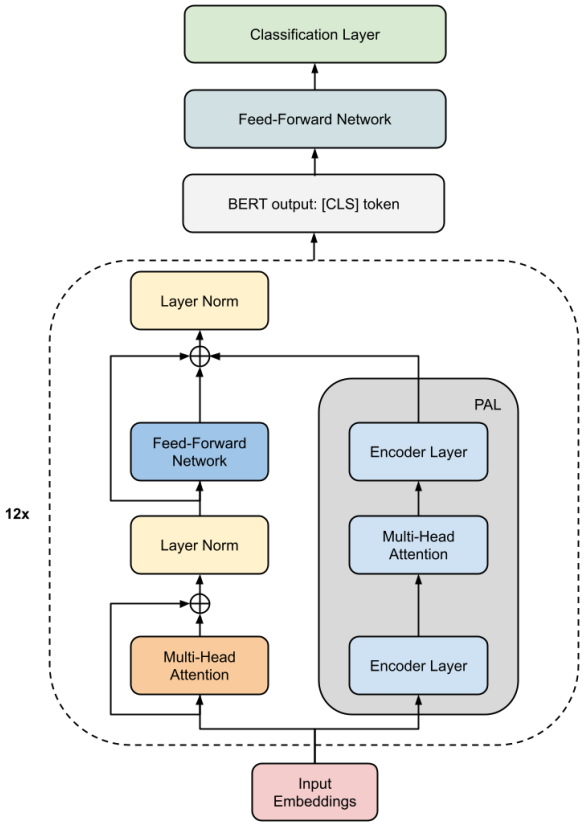


Figure 1: The model architecture

In the following sections, we delve deeper into the model architecture and key methods.

4.2 Data Preprocessing

For each training example in the paraphrase detection and semantic textual similarity datasets, there are two sentences involved. To preprocess these datasets, we concatenate the two sentences of each sample into a single sequence, separated by a [SEP] token. The tokenizer then processes this concatenated sequence by adding a [CLS] token at the beginning and appending padding tokens at the end if needed, ensuring that all sequences in a batch have the same length. This method maintains consistency with the input format used during the BERT model's pre-training, facilitating effective learning for both paraphrase detection and semantic textual similarity tasks.

4.3 Gradient Projection

One major challenge in multi-task learning is gradient interference, where gradients from different tasks can counteract each other. Yu et al. (2020) identified the tragic triad that prevents efficient multi-task learning: conflicting gradients, dominating gradients, and high curvature. To address this problem, they introduced a technique called gradient surgery (PCGrad), which projects a task's gradient onto the normal plane of any conflicting gradient from another task. Specifically, if $\mathbf{g}_i$ and $\mathbf{g}_j$ are conflicting gradients, PCGrad will projects $\mathbf{g}_i$ onto the normal plane of $\mathbf{g}_j$ as:

$$\mathbf{g}_i' = \mathbf{g}_i - \frac{\mathbf{g}_i \cdot \mathbf{g}_j}{\|\mathbf{g}_j\|^2} \mathbf{g}_j \tag{5}$$

However, the PCGrad algorithm assumes that the gradient cosine similarity between any two tasks should be zero after projection. Moreover, it is ineffective for positive gradient similarities. To address these shortcomings, Wang et al. (2020) developed a new algorithm called Gradient Vaccine, which offers a more robust solution to the issue of gradient interference in multi-task learning. For example, suppose the cosine similarity of $\mathbf{g}_i$ and $\mathbf{g}_j$ is $\cos(\theta) = \phi_{ij}$ and we have some similarity goal of $\cos(\theta') = \phi_{ij}^T > \phi_{ij}$, the magnitude and direction of $\mathbf{g}_i$ will be altered as:

$$\mathbf{g}_i' = \mathbf{g}_i + \frac{\|\mathbf{g}_i\| \left(\phi_{ij}^T \sqrt{1-\phi_{ij}^2} - \phi_{ij}\sqrt{1-(\phi_{ij}^T)^2}\right)}{\|\mathbf{g}_j\| \sqrt{1-(\phi_{ij}^T)^2}} \cdot \mathbf{g}_j \tag{6}$$

where ϕ_{ij}^T represents the similarity objective of gradients $\mathbf{g}_i$ and $\mathbf{g}_j$ and can be set as the moving average of the cosine similarity of gradients $\mathbf{g}_i$ and $\mathbf{g}_j$.

4.4 Task Scheduling

A straightforward method for training a model on multiple tasks is round-robin sampling. The process begins with a batch from the sentiment analysis dataset, followed by a batch from the paraphrase detection dataset, and then a batch from the semantic textual similarity dataset. After this, it loops back to process another batch from the sentiment analysis dataset. This cycle continues until finish. However, this approach falls short when tasks have varying numbers of training examples. In our case, the Quora dataset has 283003 train examples, significantly more than the SST and STS datasets. Consequently, using round-robin sampling results in less training for the Quora dataset and overfitting for the SST and STS datasets. A more advanced approach is to sample training data based on dataset sizes, formulated as follows:

$$p_i \propto N_i^{\alpha} \tag{7}$$

We use $\alpha = 0.5$ in our experiments, and this method is called square root sampling. Stickland and Murray (2019) observed that it was beneficial to train on tasks more equally towards the end of training. To achieve this, they developed the annealed sampling method, where α is adjusted with each epoch e:

$$\alpha = 1 - 0.8\frac{e-1}{E-1} \tag{8}$$

We evaluated both of these scheduling mechanisms in our experiments, and the results are presented in Section 5.

4.5 Baseline

Our baseline model is trained using a simple multi-task learning approach, utilizing round-robin sampling and excluding projected attention layers and gradient projection for conflicting gradients.

5 Experiments

5.1 Data

The dataset used for sentiment analysis is the Stanford Sentiment Treebank Socher et al. (2013) (SST) dataset, with 8544 train examples, 1101 dev examples and 2210 test examples. For paraphrase detection, we used the Quora dataset with 283003 train examples, 40429 dev examples and 80858 test examples. Finally, the dataset used for semantic textual analysis is the SemEval dataset, with 6040 train examples, 863 dev examples and 1725 test examples.

5.2 Evaluation method

To evaluate model's performance on sentiment analysis and paraphrase detection, we used accuracy scores. For the semantic text similarity task, the Pearson correlation coefficient is used as the evaluation metric.

5.3 Experimental details

Our models are fine-tuned from the $\text{BERT}_{\text{BASE}}$ model using the provided pre-trained weights, and they are all trained with 10 epochs, a learning rate of $1e^{-5}$ and batch sizes varying from 16 to 128.

5.4 Results

Our first experiment compares the strategies used to address gradient conflicts with the baseline model. We tested gradient surgery and gradient vaccine, and results are shown as follows:

	Overall	SST	Paraphrase	STS
Baseline	0.748	0.508	0.861	0.750
PCGrad	0.772	0.501	**0.895**	0.841
GradVac	**0.781**	**0.527**	0.893	**0.848**

Table 1: Dev set performance comparing baseline, PCGrad and GradVac

Based on the Table 1, the gradient vaccine method achieves the best dev set performance among 3 models, significantly better than the baseline.

Next, we compare the baseline model with the addition of projected attention layers and various task scheduling mechanisms.

	Overall	SST	Paraphrase	STS
Baseline	0.748	0.508	0.861	0.750
PALs + Round Robin	0.770	0.512	0.889	0.82
PALs + Square Root Sampling	0.771	0.522	0.881	**0.821**
PALs + Annealed Sampling	**0.776**	**0.524**	**0.896**	0.816

Table 2: Dev set performance comparing baseline with PALs and various task scheduling approaches

The results in Table 2 show that PALs with annealed sampling provide the best performance on the dev set, significantly outperforming the baseline. Among all the approaches, GradVac achieves the highest performance on the dev set, with an overall score of 0.781.

	Overall	SST	Paraphrase	STS
PALs + Annealed Sampling	0.785	0.522	**0.895**	0.873
GradVac	**0.787**	**0.532**	0.893	**0.875**

Table 3: Test set performance comparing GradVac and PALs with Annealed Sampling

Lastly, the above table presents the model performance on the test set, comparing GradVac with PALs using annealed sampling. GradVac continues to perform better.

6 Analysis

In this section, we delve deeper into the experiment results and conduct qualitative analysis.

6.1 Gradient Vaccine

According to the experimental results, the gradient vaccine outperforms all other approaches. Figure 2 shows the number of gradient projections per epoch using GradVac during training. The first epoch has the highest number of projections, which resolves most of the conflicting gradients. GradVac

plays a little role in later epochs; we believe this is because gradients from different tasks converge towards a similar direction after the first epoch. To increase the number of projections in later epochs, we could tighten the gradient similarity objective by setting a lower ϕ_{ij}^T.

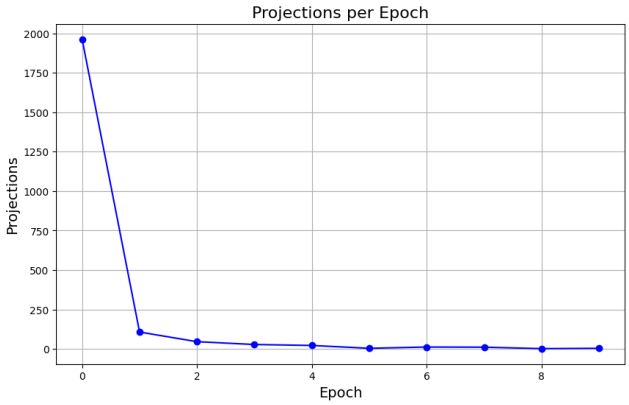
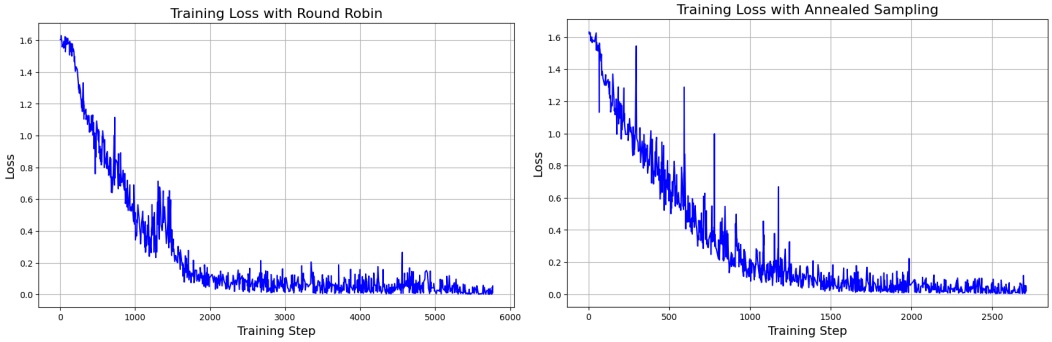


Figure 2: The number of GradVac projections per Epoch

6.2 Overfitting

Due to the varying sizes of different datasets, employing round-robin scheduling results in overfitting for the smaller datasets, as illustrated in the Figure 3a. In contrast, using annealed sampling allocates a smaller training budget to the smaller datasets and a larger training budget to the larger dataset, thereby reducing the risk of overfitting as shown in the Figure 3b.



(a) SST training loss of PALs with round robin (b) SST training loss of PALs with annealed sampling

Figure 3: SST training loss with different scheduling strategies

6.3 STS Performance

When training PALs with annealed sampling, STS performance on the dev set almost consistently declines during training as shown in Figure 4. This decline is a major reason why PALs do not outperform GradVac. We also experimented with adding a sigmoid function after the linear transformation and rescaling the output to [0, 5], but the results remained similar. Our model currently processes the concatenation of two sentences and applies transformations to the final hidden state of the single output [CLS] token. However, for semantic similarity tasks, it might be more effective to process two sentences separately and use cosine similarity loss on the hidden states of the two [CLS] outputs.

7 Conclusion

In this paper, we explored various learning approaches that leverage a single shared BERT model for multiple tasks. Our best model achieved a score of 0.787 on the test set, with GradVac delivering

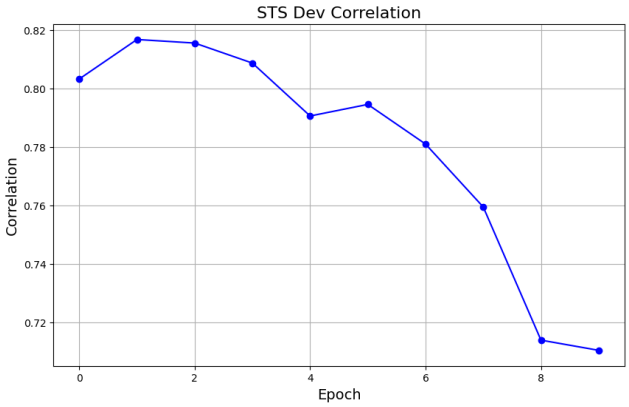


Figure 4: STS correlation on the dev set per Epoch

the best overall performance. However, GradVac exhibited overfitting for smaller datasets due to its use of round-robin task scheduling. We subsequently experimented with PALs and more advanced scheduling strategies to mitigate overfitting. Advanced task scheduling methods, such as annealed sampling, allocate more training resources to larger datasets like Quora, resulting in better performance on paraphrase detection. However, PALs fell short in the semantic textual similarity task. Future work could enhance PALs performance on this task by employing cosine similarity and sentence embeddings.

8 Ethics Statement

One ethical concern is bias. If the dataset lacks diversity and representation from different demographics, locations, and opinions, it may lead to biased predictions. For instance, a sentiment analysis model trained mainly on the data from a specific region might not generalize well to other contexts. To mitigate biases, it is crucial to ensure that training data is inclusive, which can help reduce bias and improve the model's generalization abilities. Another ethical challenge is misuse and manipulation. For example, language models can be misused for malicious purposes, such as spreading misinformation and manipulating public opinion. By fine-tuning the model with additional data using techniques like Reinforcement Learning from Human Feedback Ouyang et al. (2022) (RLHF), it is possible to mitigate these risks. RLHF can help a model avoid dangerous behaviors or decline dangerous requests.

References

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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.

Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In *International Conference on Machine Learning*, pages 5986–5995. PMLR.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. 2020. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *arXiv preprint arXiv:2010.05874*.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836.