Multitask Learning for BERT Model

Stanford CS224N Default Project

Shuojia Fu
Department of Civil and Environmental Engineering
Stanford University
shuojia@stanford.edu

Chunwei Chan
Stanford University
cweic9@stanford.edu

Abstract

Multitask learning has broad applications in large language models, where many tasks exhibits similarities, thereby improving overall model effectiveness. However, multitask learning can have difficult convergence as different tasks may have different converging speed and might have conflicting learning gradients. Therefore, in this paper, we integrated different techniques to improve model performance including gradient surgery, different weights of tasks in the objective, extend dataset and include cosine similarity as objective function. Our goal is to evaluate the effects of each method on the model performance. We found that the integration of PCGrad, extending the dataset, setting different weights for different tasks cannot help improve model performance. In contrast, application of cosine similarity for semantic analysis, model structure modification and hyper parameter finetuning could improve the model performance.

1 Key Information to include

- Mentor:Arvind Venkat Mahankali

2 Introduction

BERT, a bidirectional encoder representation from transformer, is a revolutionary model in the Large Language Model field that enables the machine to undertsand human language effectively. Its bidirectionality and multi-head attention structures enable the model to capture the semantic relationships between words in each text, leading to a deeper understanding on the language and a more accurate generation of text embedding. The versatility of the BERT model enables its application across a variety of language tasks, including question-answering systems, information retrieval, text summarization, sentiment analysis, paraphrase detection, and assessing semantic textual similarity. The multifaceted applicability of BERT necessitates its training through multitask learning, and such a training ensure that the model can achieve good performance across different tasks simultaneously.

However, multitask learning presents challenges, as tasks often converge at different rate, making it difficult for the overall model to reach optimal performance. In addition, the risk of bias towards certrain tasks due to the variation in data size or the different configuration of objective function can worse the model's performance in multitasks. Previous research has explored various strategies to enhance multitask learning for specific downstram tasks, and these strategies include employing multi-head attention regimes, training the model across multiple modules, and expanding training dataset. Driven by the challenges of achieving convergence in multitask learning and importance of multitask learning for BERT model specifically, this paper will investigate the multitask learning of BERT for three downstream tasks: sentiment analysis, paraphrase detection, and semantic similarity analysis. We aim to identify approaches that can boost the model's overall performance and investigate the effects of each approaches. This study is important because it identifies the key problems for multitask convergence for BERT and proposes some niche techniques to solve them. Our findings will provide valuable insights for subsequent research efforts that leverage BERT, offering strategies to enhance its performance across a variety of tasks.

3 Related Work

Multi-tasks learning can make more efficient usage of data and computation. It enables the model to discover shared structure and parameters across different tasks, therefore achieving better efficiency and performance than solving task individually. Theoretically, multi-task model can be resolved by employing a multi-head/ multi-output model. However, in practice, multi-task optimization is hard. Prior work described some potential causes for the difficult convergence, including varying learning speed of different tasks(Kendall et al., 2018), choices of wrong model structure(Kendall et al., 2018), conflicting gradients(Yu et al., 2020).

Prior studies have explored different methods to improve multitask performance in various field. For instance, expansion of dataset has helped multitask learning in drug discovery(Sosnina et al., 2023). In addition, various architectural solutions have been proposed to multitask learning, including PathNet(Fernando et al., 2017), progressive neural network(Rusu et al., 2016).

Previous studies have focus on improving performance of Language model development and BERT on different tasks. For instance, Jiang proposed regularized optimization for fine-tuning balance the learning from downstream tasks and pre-trained data(Jiang et al., 2019). They commented that due to the limited data size from downstream tasks compared to the pre-trained model, the fine-tuned model tend to overfit on downstream data(Jiang et al., 2019). Therefore, they proposed to include smoothness-inducing regularization and Bregman proximal point optimization for regulaized optimization(Jiang et al., 2019). Using cosine similarity as loss function for analyzing semantic textual similarity(Reimers and Gurevych, 2019). Extension of dataset has also been found to improve BERT's performance in different tasks(Jiang et al., 2019).

4 Approach

4.1 Main Method

4.2 PCGrad

We wrote and adopted PCGrad (projecting conflicting gradients) to deconflict gradients during optimization. The procedure can be described as follow: first, the model determines whether g_i conflicts with g_j by computing the cosine similarity between the two gradients; second, if the gradients between two tasks are in conflict, representing by the negative cosine similarity, we project the gradient of each task to the normal plane of the other task's gradient, which helps remove the conflicting part of the gradients. The following equation has been applied:

$$g_i^{\text{PC}} = g_i^{\text{PC}} - \frac{g_i^{\text{PC}} \cdot g_j}{\|g_j\|^2} g_j \tag{1}$$

Finally, we apply PCGrad across all the three tasks.

4.3 Different weights for tasks in optimization objective

We developed a baseline model utilizing a balanced multi-objective approach across three tasks. However, the model underperformed on the semantic analysis task with the STS development dataset compared to the other two tasks. We hypothesized that this can due to the much smaller dataset of STS compared to the other two tasks. To solve such a issue, we amplified the weight of the STS task's loss function within the overall objective, aiming to mitigate the adverse effects of the disproportionate dataset sizes.

4.4 Extended Dataset

4.4.1 Automated Data Augmentation Using NLP Aug

In order to enhance the diversity and volume of our datasets for sentiment analysis and semantic textual analysis, we employed NLP Aug(Ma, 2019), an open-source library designed for augmenting natural language processing (NLP) datasets. This tool facilitated the automatic generation of new sentences from our existing datasets—Stanford Sentiment Treebank (SST) for sentiment analysis and SemEval for semantic textual analysis—thereby addressing the challenge of limited data size and variety.

4.4.2 NLP Aug Configuration

NLP Aug operates by taking an input sentence and producing a new sentence with alterations based on specified settings, including the percentage of the sentence to be replaced and the choice of augmentation database (WordNet for synonym replacement and PPDB for paraphrasing). Our configuration was as follows:

Percentage of Replacement: We determined an optimal percentage of each sentence to be altered, aiming to strike a balance between introducing sufficient diversity and maintaining the original meaning and sentiment of the sentences. Database Selection: For sentiment analysis augmentations using the SST dataset, we primarily utilized WordNet to ensure that synonyms matched the original sentiment. For semantic textual analysis with the SemEval dataset, PPDB was employed to generate paraphrases that preserved the original semantic relationships.

The augmentation process was straightforward yet effective:

Each sentence from the SST and SemEval datasets was passed through NLP Aug, specifying the desired replacement percentage and database. NLP Aug then automatically generated a new sentence, incorporating synonyms or paraphrases as per our configuration. This approach enabled us to efficiently expand the SST dataset from 8k to 14k samples and the SemEval dataset from 6k to 16k samples, thereby enriching our training data without the need for manual sentence analysis or validation.

4.5 Cosine Similarity

Cosine similarity is chosen for measuring sentence similarity in our study due to its effectiveness in capturing the semantic orientation of sentences in vector space. Unlike other metrics, cosine similarity evaluates the cosine of the angle between two vectors, making it inherently sensitive to semantic nuances while being robust against variations in sentence length. This attribute is crucial for assessing the semantic similarity between sentences accurately.

Given the natural range of cosine similarity is between -1 and 1, and our sentence similarity labels are originally on a 0-5 scale, we scale these labels to a 0-1 range. This scaling ensures compatibility between the model's output and human-judged similarities, enabling a more straightforward evaluation of model performance. By dividing the original similarity labels by 5, we align the highest similarity label (5) to 1 and maintain 0 as the lowest score, facilitating a direct and meaningful comparison.

4.6 Modified Model Structure

Our modified BERT model architecture introduces tailored approaches for each of the three tasks—Sentiment Analysis (SST), Paraphrase Detection (Para), and Semantic Textual Similarity (STS)—to optimize performance across multitask learning. Here's a brief overview of the key modifications:

4.6.1 Sentiment Analysis (SST):

We maintain a straightforward linear projection from the BERT pooled output to the task-specific output size, optimizing it with a cross-entropy loss function. This simplicity reflects our finding that additional complexity, such as dropout or GELU activation, offered negligible benefits for our model's performance in our SST data and tasks.

4.6.2 Paraphrase Detection (Para):

For this task, we leverage a concatenated output that combines the pooled representations of two input sentences along with their absolute difference. This is then projected down to a single dimension: BERTHIDDENSIZE * 3. This architecture aims to capture not only the individual sentence embeddings but also the relationship between them, which is crucial for identifying paraphrases. The model is optimized using a binary cross-entropy loss with logits, focusing on distinguishing paraphrases from non-paraphrases effectively.

4.6.3 Semantic Textual Similarity (STS):

Acknowledging the ordinal nature of sentiments on a scale from 0 to 5, our approach departs from conventional classification techniques. We innovatively combine max and mean pooling strategies to capture a richer representation of sentences, followed by a linear layer to project these enriched features. The model is optimized using the mean squared error (MSE) loss to approximate sentiment intensity, recognizing the continuity in sentiment scores and treating sentiment analysis as a regression task rather than discrete classification.

Throughout our experiments, we tested the incorporation of dropout and GELU activation layers to introduce regularization and non-linearity. However, these additions did not significantly impact our results, suggesting that the primary BERT architecture, combined with our task-specific modifications, was sufficient for our multitask learning objectives.

By adjusting our model's architecture to the specific demands of each task, we aimed to leverage the inherent strengths of BERT while addressing the challenges of multitask learning. These modifications reflect our iterative process of refining the model structure to enhance performance across a diverse set of NLP tasks.

4.7 Hyperparameter Finetuning

In our quest to optimize the multitask BERT model, we meticulously experimented with several key hyperparameters, including learning rate, epochs, batch size, and dropout rate. Each parameter was varied within a specified range to observe its effect on the model's performance and training efficiency. Below, we summarize our findings and conclusions drawn from these experiments:

4.7.1 Learning Rate:

We tested learning rates ranging from 1e-3 to 1e-6. Through evaluation on the development set, we observed that a learning rate of 1e-5 consistently yielded the best balance between convergence speed and performance stability across all tasks.

4.7.2 Epochs:

Our experimentation with epochs spanned from 10 to 20. While extending training to 20 epochs led to marginal performance improvements, it also resulted in a doubling of training time. Analysis of the learning curves suggests that optimal results occur between 10 to 12 epochs, providing a practical trade-off between model accuracy and computational efficiency.

4.7.3 Batch Size:

Considering batch sizes of 4, 8, 16, and 32, we found that while a smaller batch size of 4 achieved the highest accuracy, it significantly increased training time. Given the constraints of GPU memory, we often utilized batch sizes of 16 or 32 for rapid experimentation. However, for producing our final outputs, we settled on a batch size of 8 as it offered an optimal compromise between training efficiency and GPU memory utilization.

4.7.4 Dropout Rate:

Variations in dropout rate (0, 0.1, 0.2, 0.3) appeared to have minimal impact on the overall model performance. This suggests that, within the tested range, the model is relatively robust to changes in regularization strength.

Our experiments highlight the intertwined nature of hyperparameter settings with task-specific and dataset-specific characteristics. The optimal configuration for one task may not translate directly to another, underscoring the necessity of task-specific tuning to achieve the best possible model performance. The accompanying charts for epochs and batch sizes illustrate the trade-offs between training time and model accuracy, providing a visual guide for selecting hyperparameters in future experiments

4.8 Baseline Model

We will have two baseline models. The first baseline model is the base BERT model from part one, with 12 Encoder Transformer layers. The baseline model is only trained with objective on sentimental analysis task, not the other two. The second baseline model is the base BERT model, trained with evenly-weighted multi-objectives of three tasks. We then compare the modified model and the two baseline models' performance on the three downstream tasks.

5 Experiments

Data We used the default dataset defined in the default course project description, including the CFIMDB dataset (binary), Stanford Sentiment Treebank dataset (multi class), Quora dataset and SemEval STS Benchmark Dataset. Media review, Stanford Sentiment Treebank dataset are used for sentimental analysis. Quora dataset is a binary label and is used for paraphrase detection. SemEval STS Benchmark dataset is with paired sentence with scores ranging from 0 to 5 to describe the similarity between sentences. It is used for semantic textual similarity.

We also extend the dataset by applying NLP Aug to SST and SemEval, to enhance the performance for sentiment analysis and semantic textual analysis.

Evaluation method We evaluated our model based on accuracy for paraphrase detection (as it is a binary classification), and Pearson correlation of the true similarity value and the predicted one for the semantic textual similarity. We will use accuracy for sentimental analysis (accuracy for both binary labels and multi-classes labels)

5.1 Experimental details

Our experimental setup was conducted on a local NVIDIA RTX 4090 GPU due to constraints with cloud resources. Key aspects of our methodology included:

5.1.1 Model Configuration:

A multitask BERT model focusing on sentiment analysis, paraphrase detection, and semantic textual similarity, incorporating strategies like gradient surgery and task-specific loss weighting. Sentiment analysis is trained with cross-entropy loss, paraphrase detection is trained with binary cross entropy loss, and semantic textual similarity is trained with MSE loss.

5.1.2 Hyperparameter Tuning:

We experimented with learning rates from 1e-3 to 1e-6, batch sizes between 4 and 32, and epochs ranging from 10 to 20. The optimal configuration identified was a learning rate of 1e-5, a batch size of 8, and around 10 to 12 epochs.

5.1.3 Training Duration:

The comprehensive training required approximately 8 hours, balancing performance with computational limits of our hardware.

5.2 Results

Comparative analysis of our BERT model's performance under different conditions revealed enlightening outcomes:

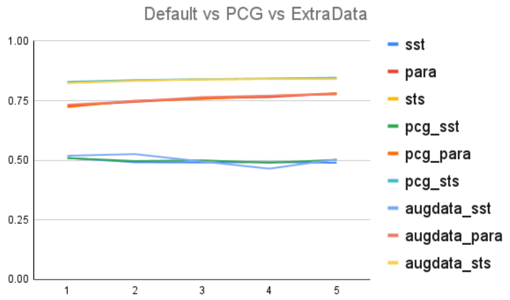

Figure 1: Baseline vs PCG vs ExtraData

5.2.1 Baseline with multi-objectives vs. PCGrad vs. Extended Data:

Surprisingly, the addition of PCGrad and augmented data did not significantly alter model performance compared to the baseline with multi-objectives. Further investigation into PCGrad's ineffectiveness revealed that conflicts in gradients, dot product is negative, among the three tasks were rare (less than 5% of the time), and even when present, the conflicts were minimal. This suggests that gradient conflict was not a significant obstacle for our multitask learning scenario.

5.2.2 Impact of Augmented Data:

Augmented data did not yield the expected performance improvements. Detailed analysis suggested that word replacements might inadvertently alter sentence meanings, rendering some augmented data labels inaccurate. Furthermore, the diversity introduced by NLP Aug may not have been sufficiently represented in the development dataset to positively impact model evaluation.

5.2.3 Debugging and Refinements:

Through debugging and reviewing model performance across different setups, it became clear that strategic adjustments in model architecture and task-specific optimizations were more crucial for improving outcomes.

5.2.4 Final Model Performance:

Despite the minimal impact of PCGrad and augmented data, our final model showcased improved accuracy in sentiment analysis and paraphrase detection tasks, attributed to refined architectural decisions and hyperparameter tuning. However, the semantic textual similarity task presented a challenging frontier, with modest gains indicating potential areas for future exploration.

This journey through experimental adjustments and outcome analyses underscores the nuanced nature of multitask learning with BERT. It highlights the importance of targeted architectural and hyperparameter optimizations over generic solutions like gradient surgery or data augmentation.

SST accuracy	Paraphrase accuracy	STS correlation	overall
0.511	0.861	0.838	0.764

Table 1: Results: Evaluations on test sets for Finalized Model

6 Analysis

6.1 Ineffective strategies

We found that the integration of PCGrad, data extension, different weights of tasks in the multi-objective don't improve the model performance compared to the baseline model trained with evenly weighted multi-objective. The limited effectiveness of PCGrad indicates that conflicting gradient is not a barrier in our study (as negative dot product appears less than 5% of the time). This is reasonable, given that the three tasks share substantial similarities and don't have conflicting objectives. The extended dataset with the application of NLP Aug doesn't help the model performance and we deduce that this can be due to the reason we only mimic and paraphrase the existing dataset, but not introduce new data. Such a process can be seen as a "copy" of the existing dataset, and its effects can be limited. In addition, closely examining the NLP augmentation, we discovered that certain imprecise word substitutions change the meaning of the sentences. These alterations adversely affect the dataset, leading to unintended and negative consequences. The constrained effectiveness of different weights of task in the final objective indicate that unbalanced data size is not the problem in training the multitask model and each task is relatively independent on each other. Such a hypothesis has furthered been proved as we train the model on each task individually and independently, and the individual model achieve similar performance as the multitask modeling.

6.2 Effective strategies

In contrast, we found that model structure modification, hyper parameter tuning, and cosine similarity for STS set help the model performance. To be specific, for the model structure, we found that the integration of pooling can effectively improve the model performance. Max pooling and mean pooling can be found to improve the model performance, especially in STS dataset. Such a effect is expected because the pooling layer helps to extract more key information and reduce noise in the dataset. For STS data, CLS is not enough since the information may be too general and aggregated. Therefore, more information is required and the pooling layers help extract more information. Cosine similarity measures the similarity between two input and is applied to semantic analysis. The implementation of cosine similarity calculation has been demonstrated to enhance model performance significantly. This approach allows for the capture of semantic nuances, enabling a more accurate differentiation of semantic meanings across various scales. Finally, we try different hyper parameters and found the combination that yields the best result on the dev set. The result shows that hyper parameter tuning can improve the model performance.

6.3 Further research direction

Upon analyzing the model's performance on the development set, we found that the majority of incorrectly labeled examples are either not represented or lack similar counterparts in the training set. Therefore, we suggest further studies to include more diverse data set into the training to help improve the model performance.

7 Conclusion

In our study, we found that model structure modification, particularly the integration of max and mean pooling for STS task, integration of cosine similarity for STS, and hyperparameter tuning can help improve BERT's performance in multitask learning in sentiment analysis, paraphrase detection, and semantic analysis. In contrast, data augmentation through NLP Aug, PCGrad, and different weighing for various tasks cannot improve the model performance. We therefore suggest that for multitask learning for BERT topic, model architecture is an important part to focus on and the specific structure should be depend on the specific task (for instance, for our semantic analysis task, we introduce max and mean pool layers, and introduce cosine similarity calculation).

An limitation of our study is that for data augmentation, we only apply NLP Aug to the existing dataset. But such application can introduce errors to the existing dataset. In addition, it doesn't introduce new samples to the dataset. Therefore, we propose further studies to introduce different and diverse dataset into the training set and further test the effects of data augmentation in multitask learning.

References

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. 2017. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2019. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Ekaterina A Sosnina, Sergey Sosnin, and Maxim V Fedorov. 2023. Improvement of multi-task learning by data enrichment: application for drug discovery. *Journal of Computer-Aided Molecular Design*, 37(4):183–200.

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.