Multitask BERT Model with Regularized Optimization and Gradient Surgery

Stanford CS224N Custom Project

Peiru Jenny Xu
Institute for Computational and Mathematical Engineering
Stanford University
peiruxu@stanford.edu

Abstract

This paper proposes a multitask BERT model with customized fine-tuning for three different NLP tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. The min-BERT architecture with multi-head attention serves as the backbone for the multitask BERT model, which is adjusted with in-domain finetuning, gradient surgery and regularized optimization strategies to perform multitask learning. Experiments show that the multitask BERT model with the proposed extensions demonstrates advanced performance in handling multiple NLP downstream tasks.

1 Introduction

Natural language processing (NLP) algorithms have been actively studied in recent decades aiming to enable computers to understand, interpret, and generate human language with meaningful and contextually relevant content. With the elevation in computational capability such as enhanced GPUs and TPUs with higher bandwidths, large language models (LLMs) provide a revolutionized approach to achieve outperforming performance in broad applications, and wide uses of pre-trained LLM models such as BERT have shown significant improvement in performing various NLP tasks, ranging from sentiment analysis to question answering.

Given the board landscape of NLP downstream tasks, researchers have been leveraging the powerful BERT architecture, which has demonstrated state-of-the-art performances in many popular NLP tasks, to handle multiple tasks simultaneously. This paper proposes a multitask BERT model with customized fine-tuning for three different NLP tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. These three tasks involve identifying, categorizing, and analyzing textual data. which are quite challenging given the complexity of human language and the subtlety of semantic meanings. This paper discusses how the base BERT architecture can be extended to perform multitask learning, and explores different strategies for multitask learning, such as gradient surgery and regularized optimization, for improving performance on multitask metrics.

2 Related Work

This section discusses relevant research studies of BERT model and multitask learning, as well as novel approaches and extensions that mitigate the challenges in multitask finetuning.

2.1 BERT Finetuning for Sentiment Classification

The authors of Munikar et al. (2019) use the pretrained BERT model and finetune for sentiment classification. The proposed architecture consists of text pre-processing, pretrained BERT model as embedding, a dropout regularization and softmax classifier layer. Despite its simplicity with only

dropout layer and softmax classifier, the proposed architecture outperforms baseline models in text sentiment classifications.

2.2 BERT Model for Multitask Learning

As BERT model demonstrates advanced performance through finetuning for single specific task, researchers have turned the focus to leveraging BERT for multiple NLP tasks simultaneously. The authors of Liu et al. (2019) propose a multitask learning framework utilizing BERT. The lower layers of the multitask learning framework are shared across all tasks, and the top layers are specific for different natural language understanding (NLU) tasks. By training a single model on multiple natural language understanding tasks simultaneously, the proposed multitask model can learn more generalized and robust representations of language, reaching new state-of-the-art results on multiple natural language understanding tasks.

2.3 Gradient Surgery for Multitask Learning

In multitask finetuning, the gradients of each tasks are computed, which may result in conflicting gradient directions and lead to suboptimal performance on the tasks. To deal with the competing gradients, the authors of Yu et al. (2020) propose the gradient surgery method that projects the gradient of one task onto the normal plane of another when the two gradients are in opposite directions. This approach therefore prevents the competing gradient component of one specific task from being applied to the network of a different task, and demonstrates substantial gains in multitask learning performance.

2.4 Finetuning with Regularized Optimization

Aggressive finetuning for downstream tasks may cause overfitting and failure to generalize to unseen data or capture robust language representations. The authors of Jiang et al. (2020) propose the SMART finetuning framework to mitigate this issue. The SMART finetuning framework consists of two parts, the smoothness-inducing regularization and Bregman proximal point optimization. Smoothness-inducing adversarial regularization adds a smoothness-inducing regularizer to the original loss function of the target task, which reduces overfitting and improves generalizability to other data. Then, during the loss optimization, the Bregman proximal point optimization is utilized to prevent the model from aggressive updating. The SMART architecture achieves better generalization performances with robust and efficient finetuning.

3 Approach

This section elaborates the main approach for this project, including a brief overview of the base BERT architecture, the downstream classification model, and the extensions to the multitask BERT model. This section also discusses the baseline model that is compared with in the experiments.

3.1 Base BERT Architecture

The base BERT model described in Devlin et al. (2019) consists of 12 encoder transformer layers, where each transformer layer consists of multi-head attention, an additive and normalization layer with residual connection, a feed-forward layer, and finally an additive and normalization layer with a residual connection. The architecture of the transformer encoder layer is outlined in Figure 1, originally from Vaswani et al. (2017).

The key component of transformer layer and BERT model is the multi-head self attention. Multi-head self attention consists of a scale dot-product attention applied across multiple heads (Vaswani et al., 2017). Multi-head self attention enables the model to simultaneously attend to information from different representation subspaces across different positions. The details about scaled dot-product attention and multi-head attention are illustrated in Figure 2, originally from Vaswani et al. (2017).

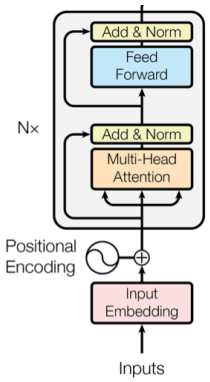

Figure 1: Transformer Encoder Layer.

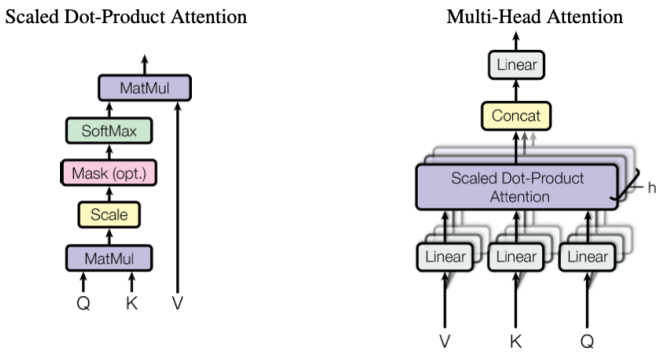

Figure 2: Multi-Head Self Attention.

3.2 Downstream Classification

The multitask BERT model is built on top of the base BERT model implementation, with downstream layers connected to handle different tasks. The model with downstream classification architecture is illustrated in Figure 3

For sentiment analysis, the pooler output from the BERT model is passed through a feed-forward neural network with ReLU activation and dropout, with last layer as a linear layer to the number of sentiment classes. The final output is the product probability logits. Cross-entropy loss is used during training, and during evaluation, the out logits are passed through softmax function.

For paraphrase detection and text similarity tasks, the pair of sentences is concatenated into a single sentence with `[SEP]` token added, and then passed into the BERT to obtain the pooler output, which is further passed through a feed-forward neural network with ReLU activation and dropout to produce a single logit. For paraphrase detection, the loss criterion used during training is cross-entropy loss, and during evaluation, the output logit is passed through sigmoid function. For text similarity task, the loss criterion is mean square loss.

For all three tasks, the number of hidden layers in the dense feed-forward neural network is set as hyperparameter. Different numbers of hidden layers are experimented as described later in the experimental details. Adam optimizer is used for efficient stochastic optimization.

In the base multitask BERT setup, the model is trained on all of the data for three tasks in a sequential manner. In sequential training, the model loads the training batch of one task, finishes training on that task completely before moving on to the next task, until all three tasks have been covered.

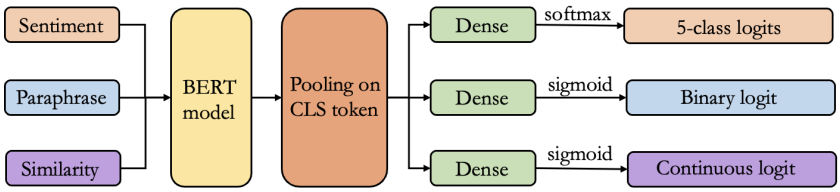

Figure 3: Multitask BERT Model Basic Architecture

3.3 Regularized Optimization

Several extensions are implemented on the foundation of the multitask BERT model, for example the SMART regularization framework to avoid the generalization issue caused by aggressive finetuning. The first step is the smoothness-inducing adversarial regularization that adds a smoothness-inducing

regularizer to the original loss function of the target task:

$$\min_{\theta} = \mathcal{L}(\theta) + \lambda_s \mathcal{R}_s(\theta), \tag{1}$$

where $\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i;\theta), y_i)$ is the original loss specific to each task, $\lambda_s > 0$ is a tuning parameter, and the smoothness-inducing regularizer is defines as:

$$\mathcal{R}_s(\theta) = \frac{1}{n}\sum_{i=1}^{n} \max_{\|\tilde{x}_i - x_i\|_p \leq \epsilon} \ell_s\left(f(\tilde{x}_i;\theta), f(x_i;\theta)\right),$$

For classification tasks, ℓ_s is the symmetrized KL-divergence $\ell_s(P,Q) = \mathrm{D}_{\mathrm{KL}}(P\|Q) + \mathrm{D}_{\mathrm{KL}}(Q\|P)$, and for regression tasks, ℓ_s is the squared loss.

This smoothness-inducing adversarial regularizer serves as a measure of the local Lipschitz continuity of f under the metric ℓ_s, and by minimizing the loss objective in (1), f is encouraged to be smooth within the neighborhoods, and such method effectively prevents overfitting and improves generalization for the given task.

The second step to is to use Bregman proximal point optimization methods to solve (1). This approach prevents aggressive parameter update that deviates too heavily from previous updates:

$$\theta_{t+1} = \arg\min_{\theta} F(\theta) + \mu D_{\text{Breg}}(\theta, \theta_t), \tag{2}$$

where $\mu > 0$ is a tuning parameter, and $D_{\text{Breg}}(\cdot,\cdot)$ is the Bregman divergence defined as

$$D_{\text{Breg}}(\theta, \theta_t) = \frac{1}{n}\sum_{i=1}^{n} l_s(f(x_i;\theta), f(x_i;\theta_t))$$

3.4 Gradient Surgery (PCGrad)

To mitigate the issue of conflicting gradients, the gradient surgery technique proposed in (Yu et al., 2020) is applied. When the gradient of the i-th task g_i is in the opposite direction of the j-th task g_j , g_i is modified as its projection onto g_j 's normal plane:

$$g_i = g_i - \frac{g_i \cdot g_j}{\|g_j\|^2} \cdot g_j$$

The projecting conflicting gradient process is visualized in Figure 4, originally from (Yu et al., 2020).

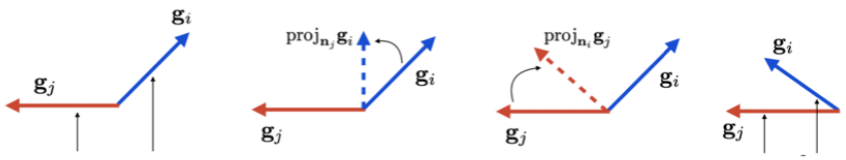

Figure 4: Gradient Surgery Demonstration

This specific method of gradient surgery is referred to as projecting conflicting gradients (PCGrad) in later sections. In order to fully exploit the potential synergies between different tasks and to fit the pattern of PCGrad, the sequential training is replaced with the round robin training, which involves cycling through all three tasks. The model is first trained on a small batch of one task, then switches to the next task; during the process, when the angle of two task gradients are more than 90 degrees, one gradient is projected to the normal plane of the other. The round robin training with PCGrad enables more balanced learning across different tasks while preventing interference between tasks.

3.5 Baseline

The baseline model is the multitask BERT architecture with only one hidden layer in the dense feed-forward neural network.

4 Experiments

This section contains the setups and results for the experiments, including the datasets, evaluation methods, experimental setups, and results.

4.1 Data

The default datasets for the project are used for training and testing purposes.

For sentiment analysis, the datasets used include the Stanford Sentiment Treebank (SST) and CFIMDB dataset. The SST dataset consists of 11,855 sentence extracted from movie reviews with a 5-class categorical label of negative (0), somewhat negative (1), neutral (2), somewhat positive (3) and positive (4) (Socher et al., 2013). Among the SST reviews, 8,544 examples are in training set, 1,101 examples in validation set, and 2,210 in test set. The CFIMDB dataset consists of 2,434 binary movie reviews as negative or positive, among which 1,701 examples are in training, 245 validation and 488 test.

The dataset for paraphrase detection is the Quora dataset, with each data is a pair of sentences with binary label indicating whether the pair is paraphrase (1) or not (0). An example question pair contains question 1 “I am a Capricorn Sun Cap moon and cap rising... what does that say about me?", and question 2 “I’m a triple Capricorn (Sun, Moon and ascendant in Capricorn) What does this say about me?", with the label “Yes" indicating the pair of questions are paraphrases of each other. A subset of the full Quora dataset is used for experiment and further split into training set (141,506 examples), validation (20,215 examples) and test set (40,431 examples).

For semantic similarity task, the SemEval STS Benchmark dataset is utilized, which consists of 8,628 different sentence pairs of varying similarity on a scale from 0 (unrelated) to 5 (equivalent meaning) (Agirre et al., 2013). Some examples include “The bird is bathing in the sink. Birdie is washing itself in the water basin” as completely equivalent (5), “John said he is considered a witness but not a suspect, ‘He is not a suspect anymore.’ ” as roughly equivalent but some important information differs (3), and “John went horseback riding at dawn with a whole group of friends. Sunrise at dawn is a magnificent view to take in if you wake up early enough for it.” as completely unrelated (0). For the STS dataset, 6,041 examples are in the training set, 864 in validation, and 1,726 in the test set.

4.2 Evaluation Method

The default evaluation method in the project handout is used in this project. For sentiment analysis and paraphrase detection, the evaluation metric is the accuracy score on the training and validation dataset. For semantic similarity task, the evaluation metric is the Pearson correlation score of the true similarity values against the predicted similarity values.

4.3 Experimental Details

All the experiments, including pretrain and finetune, are run on my personal computer with GPU Nvidia RTX-3060.

The multitask BERT classification head is first pretrained with the fixed learning rate of 10^{-3} for 10 epochs, and then finetuned with a smaller tuning learning rate (e.g. 10^{-5}) for 10 epochs.

Multiple different model configurations are experimented. The baseline model as described before has only one hidden layer in the dense feed-forward neural network. The BERT model with dense configuration, as well as all other models with extensions, have two hidden layers in the dense part. The baseline model and BERT with two hidden layers in the dense part are trained with the default hyperparameter values (10^{-5} learning rate and 0.2 dropout probability) and sequential learning manner.

The second-stage experiments focus on tuning the hyperparameters, including learning rate, dropout probability, batch size, etc. Multiple combinations of different learning rate and dropout probability are experimented, and the combination of $5 \cdot 10^{-6}$ learning rate, 0.2 dropout probability and batch size as 16 turns out as the best, and is therefore fixed for later experiments with SMART regularization and PCGrad to ensure fair comparison between different models.

The third-stage experiments take the best multitask BERT model configuration from earlier experiments, with the SMART and PCGrad implementation. The SMART implementation is adapted from the SMART-Pytorch library and adjusted to fit the architecture (Jiang et al., 2020). The PCGrad implementation is adapted and adjusted from the Pytorch-PCGrad (Tseng, 2020).

4.4 Results

The experiment results for different model configurations are shown in Table 1.

Model	Sentiment Accuracy	Paraphrase Accuracy	Similarity Correlation	Average Score
Baseline BERT	0.351	0.739	0.802	0.664
BERT + dense	0.483	0.746	0.877	0.724
BERT + hyperparameter tuning	0.469	0.874	0.836	0.754
BERT + SMART	0.495	0.864	0.88	0.766
BERT + SMART + PCGrad	**0.518**	**0.88**	**0.886**	**0.78**

Table 1: Dev Accuracy Results for Different Model Configurations

From Table 1, it is clear that the BERT multitask model with dense feed-forward neural network and hyperparameter tuning outperforms the baseline BERT model on all three tasks. Interestingly, during hyperparameter tuning, the hyperparameter that results in overall best accuracy as reported doesn't result in the best accuracy for each individual task. For sentiment classification and similarity correlation analysis, the BERT with the best hyperparameters has lower accuracy than with default hyperparameter setting. The sacrifice of accuracy between different tasks is an important finding that motivates the SMART regularized optimization and PCGrad.

With SMART regularized optimization, the BERT model shows a remarkable improvement in sentiment accuracy (from 0.469 to 0.495) and in similarity correlation (from 0.836 to 0.88) despite a small degradation in paraphrase accuracy. The proposed regularized optimization effectively improves the finetuning accuracy on validation dataset without aggressive overfitting. In addition to the SMART regularized optimization, PCGrad also help further boost the accuracy scores on all three tasks, especially for sentiment analysis and paraphrase accuracy.

The best validation and test accuracy obtained are shown in Table 2. Both the best validation and test accuracy are from the model configuration with SMART regularization and PCGrad.

Dataset	Sentiment Accuracy	Paraphrase Accuracy	Similarity Correlation	Average Score
Dev	0.518	0.88	0.886	0.78
Test	0.531	0.88	0.882	0.784

Table 2: Dev and Test Accuracy Results for Best Model

5 Analysis

The base multitask BERT model demonstrates some sacrifices and compromises between the three different tasks. Towards the later stage of the training epochs, the training loss of the base multitask BERT model is still decreasing but the validation accuracy has stopped increasing, indicating overfitting from the aggressive finetuning. With SMART regularized optimization, the issue of

aggressive finetuning is largely resolved. However, SMART regularized optimization mainly focuses on robust and efficient finetuning, but doesn't necessarily address the difficulties of multitask learning. With PCGrad to reduce the negative effect of interference between tasks, the model reaches optimal performance on each individual tasks with less compromise between tasks.

Across all models, the accuracy on sentiment classification falls far behind the paraphrase detection accuracy and similarity correlation score. One main reason is that the sentiment classification is fine-grained sentiment classification with five possible classes, which is more difficult compared to binary classification as in the case of paraphrase detection where the possible outcomes are binary. Another reason why the sentiment classification shows poor performance is the limited training sample and imbalanced classes. The SST dataset contains fewer datapoints compared to the Quora dataset, and it faces the issue of imbalanced classes. Both the training sample size and the imbalanced classes intensify the challenge and difficulty of the fine-grained sentiment classification task.

6 Conclusion

This project implements a multitask BERT model and explores SMART optimization and PCGrad methods to improve multitask learning. Although SMART regularized optimization helps with the issue of overfitting from aggressive finetuning, the performance gain is limited and the issue of overfitting still exists with the proposed robust finetuning.

One main limitations of this project is the sacrifice of accuracy between different tasks. It is noticed that during hyperparameter tuning, the hyperparameter that results in overall best accuracy doesn't result in the best accuracy for each individual task. For sentiment analysis, smaller learning rate outperforms that of larger learning rate, but the opposite for paraphrase accuracy and similarity correlation.

Future improvements of this project include data preprocessing such as data augmentation and up-sampling for imbalanced data classes, and other methods such as contrastive learning to better the sentence embeddings (Gao et al., 2022).

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. 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.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *CoRR*, abs/1901.11504.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1, pages 1–5.

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.

Wei-Cheng Tseng. 2020. Weichengtseng/pytorch-pcgrad.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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