Experiments on Multi-Task Learning Framework over BERT for Performing Sentiment Analysis, Paraphrase Detection, and Semantic Textual Similarity Simultaneously

Stanford CS224N Default Project

Florence Chen
Department of Computer Science
Stanford University
xiangyuu@stanford.edu

Abstract

The advent of large-scale pre-trained language models like BERT has significantly advanced natural language processing, leading to notable improvements across various tasks, and the introduction of the multi-task learning framework has utilized the power of pre-trained models to increase learning efficiency by training multiple tasks on a sharing structure. This study investigates the potential of multi-task learning for fine-tuning a pre-trained BERT model on three tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. In this study, I developed a minBERT model and utilized the multi-task learning framework to fine-tune its encodings, aiming to enhance its performance on the three tasks concurrently. By employing techniques such as task-specific output layers, shared encoder layers, and gradient surgery, I balanced learning signals across tasks and fine-tuned the BERT encodings to achieve higher performance than the baseline model with frozen BERT parameters and fine-tuned task specific head parameters. This study adds evidence to the effectiveness of multi-task learning with gradient surgery and experiments on the effectiveness of different configuration and neural network layers specifically for the there aforementioned tasks.

1 Key Information to include

- Mentor: Zhoujie Ding
- No External Collaborators
- Not Sharing project

2 Introduction

The emergence of large-scale pre-trained language models, such as BERT, has revolutionized the field of natural language processing by enabling significant performance improvements across a variety of tasks. While multi-task learning has become a popular approach to enable more efficient learning by tackling multiple tasks upon a sharing structure, it still suffers from challenges like conflicting gradients. This paper explores the potential of multi-task learning to fine-tune a pre-trained BERT model on three distinct but related tasks: sentiment analysis, paraphrase detection, and semantic textual similarity.

In this study, I implemented the main components of a minBERT model and fine-tuned it with respect to two datasets to achieve better sentiment analysis results on these two datasets respectively. In addition, I adopted a multi-task learning framework Bi et al. (2022) to fine-tunes the pre-trained

minBERT model to optimize its performance on the aforementioned three tasks simultaneously. My approach seeks to harness the shared semantic and syntactic structures between the tasks while mitigating negative transfer. I employed techniques such as task-specific output layers, shared encoder layers, and gradient surgery Yu et al. (2020) to balance learning signals effectively while optimizing on three losses concurrently.

My experimental results confirm that fine-tuning the minBERT parameters using the multi-task learning framework with gradient surgery can achieve better performance on each individual task, compared to the baseline model that builds on the pre-trained minBERT with not-fine-tuned parameters. By effectively designing the output layers and balancing the learning objectives, my approach leverages the strengths of the pre-trained BERT model and proposes a straightforward framework for multi-task learning on classification and regression tasks.

3 Related Work

In the paper “MTRec: Multi-Task Learning over BERT for News Recommendation" Bi et al. (2022) published in 2022, Bi et al. investigated the approach of Multi-task Learning over BERT for news recommendation tasks. This paper aims to solve the problem that the attentive multi-field learning doesn’t work effectively with deep BERT encoding in the context of News Recommendation. To solve the problem that the shallow feature encoding of category and entity information is not compatible with the deep BERT encoding for the news titles, this paper proposes a multi-task learning framework to incorporate the multi-field information into BERT encoding. The proposed multi-task learning framework includes training for the main task of news recommendation and two auxiliary tasks of category classification and named entity recognition(NER) simultaneously by optimizing the loss functions of the three tasks simultaneously.
This paper inspired my approach because we have similar problems to solve, despite that it only cares about the result of one task but my project desires to optimize the results of the three tasks simultaneously. The assumption this paper makes is that training along with the two auxiliary tasks can help boost the performance on the main task because learning of the auxiliary tasks can bring extra information to the model that is not retrievable when only training on the main task and can improve the model’s performance on the main task.

In addition, the proposed framework adopts and modifies the gradient surgery technique, which is used to alleviate the gradient conflicts among different tasks in multi-task learning. To address the fact that the auxiliary tasks only serve as methods to boost the performance of the main task, the proposed framework modifies the original gradient surgery technique by merging the gradient of the two auxiliary tasks and scale it to a smaller magnitude, then following the projection presented in the original gradient surgery technique Yu et al. (2020). This merge and scale of gradients aims to prioritize the main task by shrinking the gradient of the auxiliary tasks. This modification prioritizes the main task among the three tasks, respecting the fact that the ultimate goal is to perform well on the main task. While in my project all three tasks have same importance, it’s still applicable to modify the weight assigned to different tasks when we want specific tasks to gain more power in the optimization process and thus have better results.

The idea of the gradient surgery technique mentioned above that has been adopted and modified by Bi et al. when training for news recommendation task was originally proposed by Yu et al. in their paper “Gradient Surgery for Multi-Task Learning" Yu et al. (2020) published in 2020. Yu et al. proposed a form of gradient surgery that aims to resolve the problem of conflicting gradients when performing Multi-Task Learning. This paper reveals the phenomenon that the optimizer could reach the valley of a task and not be able to traverse the valley when there are conflicting gradients, higher curvature, and a large difference in gradient magnitudes. To prevent the interfering components of the gradient from being applied to the network, the authors devised a form of gradient surgery called *projecting conflicting gradients* (PCGrad) that alters the gradients by projecting each onto the normal plane of the other, when two gradients are conflicting.
The algorithm of PCGrad described and tested in this paper provides a great start point to implement and use gradient surgery in my project.

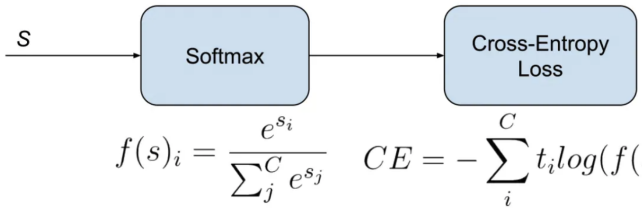

Figure 1: formula for Cross Entropy Loss

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)]$$

- N *is the number of samples*
- y_i i *is the true label for the* i^{th} *sample* (0 *or* 1)
- p_i *is the predicted probability that the* i^{th} *sample belongs to Positive class*

Figure 2: formula for Binary Cross Entropy Loss

4 Approach

This study is built on top of a BERT model, which follows the architecture described in the original BERT paper Devlin et al. (2019). Beyond the minBERT, this study adopts the multi-task learning framework and the gradient surgery technique to fine-tune the minBERT parameters in order to optimize performance on the three aforementioned tasks simultaneously. The details of the implementation of minBERT, the design of neural network layers, and the use of multi-task learning framework will be explained under this section.

4.1 minBERT

The BERT model converts sentence input into tokens before performing any additional processing. Each token is represented by the sum of three embeddings: token embeddings, segment embeddings, and position embeddings. The token embeddings map individual input ids into vector representations based on a pre-trained vocabulary, the segment embeddings are used to differentiate between sentence types (we don't use so implemented with placeholder in this study), and position embeddings encode the position of each token in the sequence, allowing BERT to take into account the order of words. In addition to the embedding layer that consists of token and position embeddings, the BERT model makes use of 12 encoder transformer layers. Each transformer layer consists of multi-head self attention layer, followed by an additive and normalization layer with a residual connection, a feed-forward layer, and another additive and normalization layer with a residual connection.

4.2 Neural Network Architecture

In addition to the forward pass that convert sentences into BERT embeddings, I added output layers for the three downstream tasks. For the sentiment analysis task, which is a classification task with five categories, I implemented a linear output layer that transforms the BERT embedding of the input sentence into five logits. For the paraphrase detection task, which is a binary classification task, I implemented a linear output layer that transforms the concatenated BERT embedding of the sentence pair into a single logits. For the semantic textual similarity task, which is a regression task with output range between 0 and 5, I calculated the correlation and cosine similarity between the two BERT embedding, each has value ranged from -1 to 1. I then took the average of the two measurements, add 1 to shift range from [-1,1] to [0,2], and multiply by 2.5 to scale range to [0,5], and output as the logits.

Algorithm 1 PCGrad Update Rule

Require: Model parameters θ, task minibatch $\mathcal{B} = \{\mathcal{T}_k\}$

$\mathbf{g}_k \leftarrow \nabla_\theta \mathcal{L}_k(\theta) \ \forall k$

$\mathbf{g}_k^{\text{PC}} \leftarrow \mathbf{g}_k \ \forall k$

for $\mathcal{T}_i \in \mathcal{B}$ **do**

  for $\mathcal{T}_j \overset{\text{uniformly}}{\sim} \mathcal{B} \setminus \mathcal{T}_i$ in random order **do**

    if $\mathbf{g}_i^{\text{PC}} \cdot \mathbf{g}_j < 0$ **then**

      // Subtract the projection of $\mathbf{g}_i^{\text{PC}}$ *onto* $\mathbf{g}_j$

      Set $\mathbf{g}_i^{\text{PC}} = \mathbf{g}_i^{\text{PC}} - \frac{\mathbf{g}_i^{\text{PC}} \cdot \mathbf{g}_j}{\|\mathbf{g}_j\|^2} \mathbf{g}_j$

return update $\Delta\theta = \mathbf{g}^{\text{PC}} = \sum_i \mathbf{g}_i^{\text{PC}}$

Figure 3: Algorithm for updating PCGrad yu-etal-2020-NEURIPS

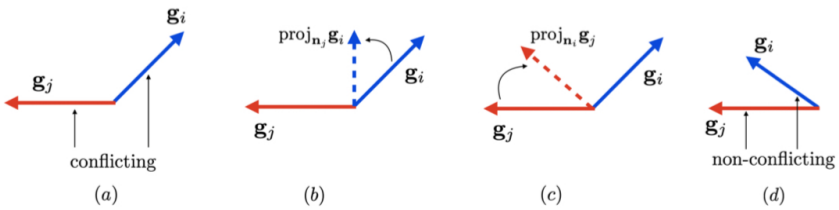

Figure 4: Diagram showing the PCGrad projection of conflicting gradients Yu et al. (2020)

4.3 Multi-Task Learning Framework and Gradient Surgery

I adopted the multi-task learning framework to train the three tasks simultaneously. To do so, in the training process, I calculated loss for each task and combine the losses together to perform the backpropagation.

For the sentiment analysis task, I applied the cross-entropy loss on the raw logits returned by the output layer because cross-entropy loss is a popular and effective loss metrics for a multi-class classification problem and it applies the log softmax activation function internally.

For the paraphrase detection task, which is a binary classification problem, I used the BCEWithLogitsLoss that combines a sigmoid layer and the binary cross-entropy loss by substituting the value of sigmoid function $\sigma(z)$ in place of p_i in binary cross entropy loss function.

For the semantic textual similarity task, I used the mean-squared-error loss(MSELoss) because the logits returned by the output layer is the estimated similarity on scale 0 to 5, which matches the format of the labels, and MSE is a classic loss function used to evaluate regression problems whose outputs are continuous values.

How to combine the three loss functions is an important choice to make. Inspired by the experiment conducted by Bi et al. on news recommendation Bi et al. (2022) and the study of gradient surgery by Yu et al. Yu et al. (2020), I adopted the form of gradient surgery called *projecting conflicting gradients* (PCGrad), which mitigates the problem of gradient interference by projecting each of the conflicting pair of gradients onto the normal plane of the other. The algorithm of the PCGrad is shown in Figure 1. The authors of the original paper released their implementation of PCGrad in TensorFlow, and I followed the PyTorch implementation provided by Tseng Tseng (2020).

4.4 Baseline Model

The baseline model uses the pre-trained BERT embeddings, the three output layers described in the **Neural Network Architecture** section, and the Adam Optimizer with weight decay regularization for stochastic optimization. In the training process, the losses of the three tasks are weighted by batch size and averaged together to form the total loss, which I performed backpropogation on. The baseline model is trained under the 'last-linear-layer' mode, in which the BERT parameters are frozen and the task specific head parameters are updated.

5 Experiments

5.1 Data

Three datasets are being used in this study, each corresponding to the sentiment analysis task, the paraphrase detection task, and the semantic textual similarity task:

1. The Stanford Sentiment Treebank Dataset (SST), which consists of 11,855 single sentences extracted from movie reviews, being parsed into 215,154 unique phrases with labels indicating each phrase's sentiment as one in {negative, somewhat negative, neutral, somewhat positive, positive}.
   This dataset contains a training set of 8,544 examples, a dev set of 1,101 examples, and a test set of 2,210 examples.
2. The Quora Dataset (QQP), which consists of 404,298 question pairs with labels indicating whether the questions in each pair are paraphrases of one another.
   This dataset contains a training set of 283,010 examples, a dev set of 40,429 examples, and a test set of 80,859 examples.
3. The SemEval STS Benchmark Dataset (STS), which consists of 8,628 different sentence pairs with scores on a scale from 0 (unrelated) to 5(equivalent meaning) indicating the similarity between the two sentences in each pair.
   This dataset contains a training set of 6,040 examples, a dev set of 863 examples, and a test set of 1,725 examples.

5.2 Evaluation method

For the sentiment analysis task and the paraphrase detection task, which are classification problems, the model's performance is evaluated by the prediction accuracy between the true and predicted labels. For the semantic textual similarity task, which is a regression-like problem, the model's performance is evaluated by the Pearson correlation coefficient between the true and predicted similarity values, measuring the linear correlation between predictions and true labels.

5.3 Experimental details

Table 1: Experiment Details

Model Specification	Learning Rate	Batch Size	Epochs	Dropout	BERT frozen
Baseline	1e-5	8	10	0.3	Yes
fine-tuned BERT + multi-task learning with PCGrad	1e-5	8	10	0.3	No

5.4 Results

Table 2: Experiment Dev Results

Model Specification	SST Dev Accuracy	QQP Dev Accuracy	STS Dev Correlation
baseline	0.359	0.648	0.062
fine-tuned BERT + multi-task learning with PCGrad	0.474	0.702	0.408

Table 3: Experiment Test Results

Model Specification	SST Test Accuracy	QQP Test Accuracy	STS Test Correlation
fine-tuned BERT + multi-task learning with PCGrad	0.496	0.702	0.386

6 Analysis

The neural network structure I built is a rather straightforward one. The sentiment analysis and paraphrase detection tasks each has an extra output layer above the BERT embeddings, and the semantic textual similarity is calculated from the BERT embeddings without extra neural network layers. The experiments results show that this fine-tuned model performs the worst on the semantic textual similarity task, which may be improved by adding extra layers while the design of layers is a potential project to work on.
The baseline model performs really bad on the semantic textual similarity task, I attribute this to two possible causes. The first cause is that I failed to scale the computed cosine similarity to be in range [0, 5], but rather times the cosine similarity by 5 to get range [-5, 5]. By checking out the predictions, I see some negative valued predictions, which are out of range and result in bad correlation measures. I fixed this problem in my fine-tuned model. The second cause is that the baseline model was trained under the 'last-linear-layer' mode, in which only the task specific parameters are fine-tuned but not the BERT parameters. Given the model adds no extra layer for the semantic textual similarity task, no parameters are fine-tuned for this task.
The fine-tuned BERT model with gradient surgery outperforms the baseline model dramatically, especially on the semantic textual similarity task. This result suggests the applicability of fine-tuning on BERT parameters for multiple tasks concurrently, and the effectiveness of the gradient surgery technique to facilitate the stochastic optimization process. However the fine-tuned model still doesn't perform as well on the semantic textual similarity task. This could be attribute to the lack of output layer for this specific task. Given the other tasks have extra output layers and parameters, this task is less represented in the training process and could be negatively affected by the fine-tuning on other tasks' specific parameters.
Given the similarity in dev accuracy / correlation and test accuracy / correlation, I don't see any signal of overfitting for this model.

7 Conclusion

This study experiments on the multi-task learning framework and the gradient surgery technique, proves their effectiveness on fine-tuning BERT embeddings to optimize multiple different but related tasks simultaneously.
The limitation of this study lies on the lack of experiments on more complex neural network layers for the downstream tasks and the lack of hyperparameter optimization. The model performance on dev and test sets even suggests an issue of underfitting. Furthermore, for future extension of this study, possible improvements include data-related methods and regularization optimization.

Ethical Challenges and Possible Social Risks. One ethical concern is the risk that the models could perpetuate and even amplify existing biases in society if they are trained on biased or skewed datasets. For example in the sentiment analysis task, biased training data could result in encoding negative sentiment to some neutral tokens. One strategy to mitigate this risk is to increase the diversity of data, so different or opposed opinions to the same token are equally trained into the model. Another strategy to mitigate this risk is to clean the training data by removing biased data.
Another ethical concern is that if the data contains sensitive information or personally identifiable details of individuals, their privacy rights could be violated. One strategy to mitigate this risk is to carefully select data in the data collecting process, to make sure data are anonymous and contain no personally identifiable information.

References

Qiwei Bi, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Hanfang Yang. 2022. MTRec: Multi-task learning over BERT for news recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2663–2669.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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.