Enhancing Multi-Task Learning with BERT

Stanford CS224N Default Project

Josiah Griggs
Department of Computer Science
Stanford University
griggsjo@stanford.edu

Abstract

In this project, I explore several strategies in an effort to create robust embeddings that improve the performance of a pre-trained BERT model across three tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. My main goal was to refine the model's embeddings through various learning techniques and achieve improved downstream performance for each task. I focused on architectural modifications including gradient surgery, smoothness-inducing adversarial regularization, and cosine similarity loss. The results show noticeable improvements beyond the baseline, fine-tuned BERT model.

1 Key Information to include

- Mentor: Aditya Agrawal
- No external collaborators
- Project not shared across classes

2 Introduction

While the field of natural language processing has existed since as early as the 1940s, its rapid evolution over the last couple decades is largely due to the advent of sophisticated, pre-trained language models, one of which is BERT (Devlin et al. (2019)). These models are useful for providing a solid foundation to build models that are more effective at specific tasks.

This paper outlines my approach to improving BERT for three sentence-level tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. Although it would be reasonable to fine-tune separate models for each task, this is inefficient and potentially sacrifices gains that the model learns from overlapping objectives within the tasks. This project involves training the model to succeed at all of the three tasks simultaneously. I decided to focus on improving the model architecture to deepen my understanding of the trade-offs involved with designing flexible neural networks.

3 Related Work

My work builds on the foundation of the pre-trained BERT model by incorporating several sophisticated techniques to enhance the performance of BERT for multitask learning. In this section, I discuss the existing work that informed my approach.

Regularization Techniques: A major challenge for fine-tuning pre-trained models is overfitting, especially when dealing with smaller datasets. To address this, SMART (Smoothness-Inducing Adversarial Regularization) regularization (Jiang et al. (2020)) combines smoothness-inducing adversarial regularization with Bregman proximal point optimization, which is defined as:

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

where D_{Breg} is the Bregman divergence, which helps to prevent significant deviations from previous iterations. This method helps maintain a balance between learning task-specific patterns and preserv-

ing the model's generalization capabilities. SMART regularization makes the model more robust and generalizable by penalizing changes in the output for slightly altered inputs.

Cosine Similarity Loss: For the semantic textual similarity task, we must find a way to measure the closeness between sentence embeddings effectively. Common methods typically involve mean squared error or other regression-based losses. However, I utilized cosine similarity loss, inspired by the work of (Reimers and Gurevych (2019)). This approach maximizes the cosine similarity between the embeddings of sentence pairs. The cosine similarity loss enhances the model's ability to distinguish subtle differences in meaning between sentences, which is essential for improving the model's performance on semantic similarity tasks.

Gradient Surgery: Fine-tuning models for multiple tasks simultaneously creates the challenge of managing conflicting gradients. To address this, I incorporated gradient surgery, specifically the PCGrad algorithm (Yu et al. (2020)). PCGrad projects the gradients of conflicting tasks so that the interference between them is minimized. By ensuring that the updates to model parameters are not dominated by any single task, gradient surgery helps ensure a more balanced and effective learning process across all tasks.

4 Approach

My approach builds on the existing BERT model architecture (Devlin et al. (2019)). The model includes an embedding layer that handles word embeddings and positional embeddings. This is followed by 12 BERT encoder transformer layers, each featuring multi-head self-attention as outlined by (Vaswani et al. (2017)). Finally, each transformer layer incorporates an additive and normalization layer with a residual connection, a feed-forward layer, and another additive and normalization layer with a residual connection. We also use the Adam optimizer.

Loss Function: I incorporated Cosine Similarity Loss primarily for the semantic textual similarity task, but also out of curiosity for its impact on other tasks. This loss function measures the cosine similarity between sentence embeddings, aiming to better capture the relational structure of the data. The introduction of Cosine Similarity Loss led to an improvement in model performance, particularly on the STS dataset.

Improving training: I implemented gradient surgery, a technique designed to address gradient conflicts in multi-task learning. Gradient surgery reduces interference between tasks by orthogonalizing their gradients, making the training process more stable and efficient. This technique improved the model's ability to generalize on the SST and paraphrase datasets. I also incorporated SMART into the training process, but it was not as effective as the other two techniques.

5 Experiments

5.1 Data

I utilize three datasets to fine-tune the pre-trained BERT model for each downstream task with seperate train, dev, and test splits for each dataset as provided:

Sentiment classification: I used the Stanford Sentiment Treebank (SST) dataset, which contains sentences extracted from movie reviews, each labeled as negative, somewhat negative, neutral, somewhat positive, or positive. This dataset comprises 8,544 training examples, 1,101 development examples, and 2,210 test examples.

Paraphrase detection: I used the Quora Question Pairs (QQP) dataset, which includes question pairs labeled to indicate whether they are paraphrases of each other. This dataset includes 283,010 training examples, 40,429 development examples, and 80,859 test examples.

Semantic textual similarity: I utilized the SemEval STS Benchmark dataset, consisting of sentence pairs rated on a scale from 0 (not at all related) to 5 (equivalent meaning). This dataset contains 6,040 training examples, 863 development examples, and 1,725 test examples.

5.2 Evaluation method

I used the methods described in the default final project handout. For the SST and QQP datasets, accuracy was measured by comparing the true and predicted labels. For the STS task, I used the Pearson correlation coeffecient which measures the linear correlation between correct and predicted similarity values.

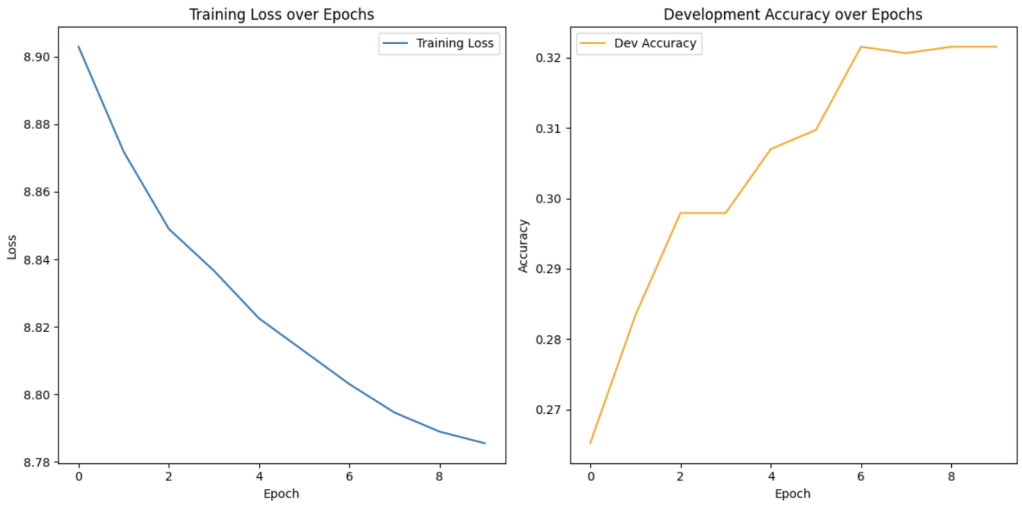
5.3 Experimental details

For my baseline, I assessed the base BERT model's performance after training on the three datasets without any adjustments. I used a learning rate of 1e-5, trained for 10 epochs, batch size of 8, and used hidden layer dropout probability 0.3. This model achieved accuracies of 0.306 on the SST dataset, 0.347 on the paraphrase dataset, and 0.054 on the STS dataset.

From there, I implemented SMART by itself to see if this would have any impact on the accuracy, but it made no noticable difference.

Next, I implemented Gradient Surgery to address conflicts between the different learning tasks. Using the same hyperparameters, this resulted in a huge boost in the performance on the SST task and Paraphrase task increasing them to 0.402 and 0.632 respectively. However, STS was not noticeably different.

In an effort to improve the pefromance on the STS task, I explored the impact of implementing Cosine Similarity Loss. Using the same hyperparameters, except for batch size was increased to 32, I achieved the best performance overall which was 0.328 on the SST task, 0.616 on the Paraphrase task, and 0.440 on the STS task.



5.4 Results

Model	SST Accuracy	QQP Accuracy	STS Pearson Correlation
Baseline	0.306	0.347	0.054
SMART	0.306	0.347	0.054
SMART + Gradient Surgery	0.402	0.632	0.054
SMART + Gradient Surgery + Cosine Similarity Loss	0.328	0.616	0.440

Table 1: Performance of different models on SST, QQP, and STS datasets.

My results were not as good as I expected. I think should have tried more techniques and waited for overfitting to become a bigger issue before implementing SMART. This would have saved me a lot of time figuring out of the implementation was correct or not. It was also disappointing to see my implementation of Cosine Similarity Loss had a large hit on the SST accuracy. This tells me that

modifying the model architecture alone is not enough to build a robust model, and I should look more into alternative datasets or processing the datasets better.

6 Analysis

SMART : This technique performed far worse than any of the others. It is possible I implemented it incorrectly. I could have also tuned the hyperparameters better to introduce a little more noise and see if this had an impact.

Cosine Similarity : As mentioned earlier, this technique led to improved performance on the STS task, but poorer performance on the SST task. For the STS task, the improvement makes sense because cosine similarity is designed to produce similar embeddings for similar sentences, which aligns with the STS task. For the SST, task, I can see how cosine similarity does not optimize for classification accuracy, but since I only applied it to the predict sentiment method, I did not expect it to impact the other tasks.

Data : I would be curious to see the impact of using larger datasets for each task because the Quora accuracy was the highest and that dataset had the most examples by far. If I had more data for the other tasks, or maybe trained the model on a more equal amount of examples per task, it may have generalized better.

7 Conclusion

In this project, I explored the impact of various machine learning techniques on a fine-tuned base BERT model. SMART regularization on its own was not nearly enough to make an impact, but combining it with Gradient Surgery and Cosine Similarity Loss led to a noticable improvement beyond the baseline in each of the three tasks. When I started this class, I only had a rough notion of what embeddings were. Now, I feel way more confident to continue exploring NLP and neural networks in general. Getting my hands dirty and understanding the math and code behind fine-tuning neural networks has been such a gratifying experience and I am so excited to continue building and learning.

8 Ethics Statement

Fine-tuning pre-trained language models like BERT for tasks such as sentiment analysis, paraphrase detection, and semantic textual similarity raises ethical concerns around bias and fairness, and the illegal use of data by companies.

Bias in training data can lead to models that perpetuate and amplify negative discriminatory stereotypes, possibly leading to unfair treatment of certain demographics. This is particularly problematic in sentiment analysis, where biased training data could result in consistently skewed sentiment scores for texts associated with specific demographics. In a time where tech companies like Google or OpenAI have so much influence over what information consumers interact with and believe to be factual, this is particularly concerning.

Another significant concern is the potential for companies to use data illegally to train their models without proper consent or consideration of data privacy laws. This misuse of data raises questions around consent and the right to privacy. As more and more companies realize the potential of growth from generative AI and the necessity for reliable training data, their willingness to cut corners could increase more and more as the pressure to release the best models also increases.

To address the ethical concerns of bias and fairness in fine-tuning pre-trained language models, one effective mitigation strategy is to implement stringent bias detection and mitigation frameworks. This can involve pre-processing steps such as removing biased language and balancing datasets to ensure diverse, non-stereotypical representation across different demographic groups. It could also involve post-processing where the outputs of models are screened before they reach the user.

For the issue of illegal data usage, a mitigation strategy could be establishing strict data usage policies that comply with legal standards across the tech industry and prioritize user consent. This would require big tech companies agreeing on a data governance framework, and also conducting regular compliance checks ideally overseen by a third party. Another solution would be using synthetic or anonymized datasets to help protect user privacy while still allowing for effective model training.

References

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)*.

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*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.