Fine-tuning BERT for Multiple Downstream Tasks

Stanford CS224N Default Project

Jianhao Cai
Department of Computer Science
Stanford University
jianhaoc@stanford.edu

Abstract

In contrast to most machine learning practices which focus on one task and optimize the model to achieve the best performance on that task, transfer learning enables the model to be extended to multiple tasks. In this paper, we explored several methods to improve the performance of the pre-trained model on 3 downstream tasks: sentiment analysis, paraphrase detection and semantic textual similarities. We started with uncased base BERT, employed round-robin sampling and annealed sampling, adopted SMART, incorporated gradient treatment methods like gradient surgery and gradient vaccine, incorporated projected attention layers, and refined the loss function for SST dataset. These techniques have achieved a mean dev accuracy of 0.754.

1 Key Information to include

- Mentor: Chaofei Fan
- External Collaborators (if you have any): No
- Sharing project: No

2 Introduction

In the past few years, Natural Language Processing has been evolving fast driven by the marvelous power of pre-trained language models like BERT and has achieved great performance on multiple tasks. However, such performance achieved on real-world problems requires fine-tuning on multiple specific downstream tasks, which is in fact challenging, especially considering the large number of trainable parameters.

In this paper, we explored a bunch of techniques to improve the performance of base BERT model on sentiment classification, paraphrase detection and semantic textual similarity. We incorporated a multitask learning mechanism including adding PAL layers with BERT for reducing the extra task-specific number of parameters, cross encoding for sentence pairs, mitigating gradient conflict with gradient surgery and gradient vaccine, round-robin sampling and annealed sampling for tackling the large difference in the size of different datasets, SMART for combating aggressive fine tuning, and introducing ordinal log loss function for SST dataset. The goal is to tackle the difficulty in multiple tasks while taking advantage of the power of the pre-trained model. The result of the analysis in the following parts has demonstrated the effectiveness as well as the limitations of these approaches.

3 Related Work

BERT model proposed by Delvin (2018), is first pre-trained for masked word prediction and next sentence prediction and then fine tuned on specific tasks with additional top layers, and has achieved

state-of-the-art performances on many NLP applications. To leverage the power of this pre-trained model in multiple downstream tasks, Liu (2019) proposed to add task-specific layers on top of BERT while the lower layers are shared among diffirent tasks, which achieved astounding results on several different natural language understanding tasks. Stickland and Murray (2019) altered the BERT layers by introducing PAL, which is a low-rank multi-head attention specific to each task and is added in parallel to BERT.

To process the input sentences, Reimers and Gurevych (2019) proposed bi-encoding in which the two sentences are first processed by the model and then the outputs are concatenated for specific task. This methods works particularly well for similarity-relavant tasks. In the contrary, Delvin (2018) introduced cross-encoding in which the two sentences are concatenated with a special token [SEP] before being processed by the model. This is widely used in sentence-pair related tasks, such as paraphrase detectiona and semantic textual smilarity.

Imbalanced datasets may be a challenge to fine tuning a model. Round robin sampling select a batch of training examples from each task in a fixed order. However, the model may overfit on small datasets and underfit on large dataset, since it has looped through the smaller datasets many time before seeing every example of the large ones. Stickland and Murray (2019) proposed an annealed sampling, in which the sampling probabilities are adjusted in proportion to the dataset size.

The loss function can have a great impact on the model performance. In the sentiment analysis classification task, the label of each example has an ordinal nature, which may not be captured by the cross entropy loss. Castagnos (2022) proposed ordinal log loss, which penalizes the predictions which are far from the true labels by adding a distance term in total loss.

Competing gradients may also be a challenge for multi-task fine tuning. Gradient directions of different tasks may conflict with each other, rendering the non-optimal performance on each task. Yu (2020) introduced gradient surgery that projects the gradients of one task ofto a normal plane of another conflicting task's gradients, preventing the conflicting being applied to the model and impairing the model performance.

Finally, overfitting may be a great challenge when fine tuning models aggressively on small dataset. Jiang (2020 proposed a regularization technique called SMART. The first step is Smoothness-Inducing Adversarial Regularization, which reduces overfitting and improves generalization by adding a regularizing term to the loss function. The second step is Bregman proximal point optimization, which imposes a strong penalty to prevent aggressive parameter updates during fine-tuning. The method is shown to effectively maintain a balance between learning specific task patterns and generalizing on unseen data.

4 Approach

4.1 Baseline

The explorations are based on the base BERT model, which consists of an embedding layer and 12 BERT transformer layers. For the baseline, we added a dropout layer and a linear layer on top of the base BERT model for each task, and all the original BERT parameter are frozen except the task specific regression and classification heads. The goal of this architecture is to see the how well the pre-trained model performs on the specific downstream tasks when the minimum number of parameters are adjusted. Round robin sampling is used in which each task batch is selected in a fixed order. The learning rate used here is 1e-3.

4.2 Handling sentence pairs

Typically, the input to the BERT model is single sentences. However, paraphrase detection and semantic textual similarity involve sentence pairs. We have two approaches to handle this case.

The first one is bi-encoding, introduced by Reimers and Gurevych (2019). Two input sentences are passed through the BERT layers independently. For paraphrase detection, the two output embeddings are concatenated and linearly transformed for classification. For semantic textual similarity, the two output embeddings are measured by cosine similarity.

The second one is cross-encoding, introduced by Delvin (2019). The two input sentences are concatenated by a [SEP] token, and are then passed through the BERT layers as well as the task-specific layers to generate predictions directly. This method is able to learn broader context and capture the relationship of the sentence pair by taking into consideration of both sentences simultaneous. We will focus on this method in the sentence-pair-involved downstream tasks.

4.3 Sampling for multi-tasks

The dataset of the three downstream tasks differ in size significantly. The simple round robin sampling is firstly adopted, selecting a batch of training examples from the smaller sst and sts dataset until every example is seen, cycling through them in a fixed order. Meanwhile, the larger para dataset is randomly sampled. However, some information is lost since since the para dataset is not fully utilized. Alternatively, we applied round robin to all the three datasets. However, by the time every example in the larger para dataset is seen, the smaller sst and sts dataset has been looped for several times, leading to significant overfitting on these smaller dataset.

To address the limitation of round robin sampling and random sampling, Stickland and Murray introduced annealed sampling strategy, where we see more examples from the larger dataset. Specifically, we select a batch of examples from task i with probability p proportional to the power of the size of the dataset of this task. The power term is set to be 1 - 0.8 * (e -1) / (E - 1). This method contributes to training on tasks more equally towards the end of the training by mitigating the interference resulted from training on one task for too many times.

4.4 Conflicting gradient treatment

In multi-task fine tuning, gradients directions of different tasks may conflict with each other. Such opposing task gradients may impair the optimization process of the training. Yu (2020) recommended a technique called gradient surgery in which one gradient is projected on another gradient if they have negative cosine similarity. To be specific, $g_i = g_i - (g_i * g_j)/(||g_j||^2) * g_j$.

Wang, Tsvetkov, Firat and Cao (2020) recommended a more complex method, gradient vaccine, in which gradient similarity is measured along the optimization trajectory. To be specific, $g_i = g_i + g_i * ||g_i|| * (a_ij.T * (1 - a_ij^2)^(0.5) + a_ij * (1 - a_ij.T^2)^)0.5))/(||g_i|| * (1 - a_ij.T^2)^(0.5)$.

In this paper., we integrated both competing gradient treatment method to accelerate the optimization process of multi-task learning.

4.5 Regularization

To mitigate the problem of overfitting resulted from aggressive fine tuning on small dataset, we used the SMART regularization approach recommended by Jiang (2020). The first step is Smoothness-Inducing Adversarial Regularization, which aims at reducing overfitting and foster smoothness, providing better generalization on tasks with limited amount of data. To be specific, the loss function is added with a smoothness-inducing adversarial regularizing term: min F() = L() + lambda * R(),

where L() is the loss function for downstream task. lambda > 0 is a tuning parameter, and R() is a smoothness inducing adversarial regularizer, defined as sum(max $l_s(f(\tilde{x};$), f(x;))) / n, where $\|\tilde{x}$ - $x\|_p <= eps$.

The second step is Bregman proximal point optimization, which acts as a powerful regularizer to prevent aggressive parameter updates during fine-tuning. To be specific, it prevents t+1 update from deviating too much from t.

In out trial, we incorporate the SMART regularization into out downstream tasks, especially sst and sts dataset, where overfitting problem is most likely to be significant.

4.6 Projected Attention Layers (PAL)

Stickland and Murray (2019) recommened adding some task-specific attention layers, named Projected Attention Layers, to further boost the performance on multiple downstream tasks. Meanwhile, to avoid too many parameters to be estimated, Stickland and Murray proposed Low-rank Layers method: TS(h) = $V_D g(V_E h)$.

Instead of only adding parameters to the top of the model, we modified the BERT layers themselves. We added task-specific parameters in parallel with each layer of the BERT model. To be specific, h = LN(h + SA(h) + TS(h)). While the parameter of the original attentions are shared among datasets, the parameters of the attention of each specific task can be fine tuned. In this paper, we used the identity function for g, and we shared the encoding and decoding matrices across the layers.

4.7 Loss for sst dataset

The labels of the sst dataset representing negative, somewhat negative, neutral, somewhat positive and positive. There is ordinal nature in the labels, which is unfortunately not taken into consideration by the frequently used cross entropy loss. In fact, the cross entropy loss thinks of the labels as categorical without any relationship in between. To cope with this problem, Castagnos (2022) recommended a ordinal log loss function which multiplied the original loss with an additional term, the distance between the true label class and the predicted label class. With this additional term, greater loss will be generated when the predictions deviate further from the ground truth, thus the ordinal nature of the labels is taken into consideration.

5 Experiments

5.1 Data

For sentiment analysis task, we use the Stanford Sentiment Treebank (SST) dataset, which consists of sentences extracted from movie reviews, and each sentence has a label of negative, somewhat negative, neutral, somewhat positive, or positive. There are 8,544 train examples, 1,101 dev examples, and 2,210 test examples. We also use the CFIMDB dataset, which consists of highly polar movie reviews, and each review has a label of negative or positive. There are 1,701 train examples, 245 dev examples, and 488 test examples.

For paraphrase detection, we use the Quora dataset, which consists of questions pairs with binary labels indicating whether they are paraphrases of one another. There are 283,010 train examples, 40,429 dev examples, and 80,859 test examples.

For semantic textual similarity, we use the SemEval STS Benchmark dataset, which consists of sentence pairs of varying similarity on a scale of 0 (not at all related) to 5 (equivalent meaning).

There are 6,040 train examples, 863 dev examples, and 1,725 test examples.

5.2 Evaluation method

For the sst and para datasets, which are used for the sentiment analysis and paraphrase detection, we measure the performance of the model using accuracy between predicted labels and true labels.

For the sts dataset, which is used for the semantic textual analysis, we measure the performance of the model using Pearson Correlation of the true similarity values against the predicted similarity values.

5.3 Experimental details

Our explorations are based on the base BERT model with pre-trained weights. The hidden size is always 768. The hidden layer dropout probability is 0.2. The batch size we used was typically 32. All the models were trained with AdamW optimizer, where beta$_1 = 0.9, beta_2 = 0.999, lambda = 0.01$.

For the baseline, we only trained the top head layers with all the BERT layers frozen with a learning rate of 1e-3. For other models with all the parameters trainable, we used the learning rate of 1e-5.

For the smart regularization, the number of optimization steps to find noise was 1, the step size to improve noise was 1e-3, the noise norm constraint was 1e-6, and the initial noise variance was 1e-5.

We first explored the sentence pair embedding technique. We tried two approaches: bi-encoding and cross encoding on para dataset and sts dataset incorporated in the baseline model. The results are shown below:
method para-accuracy sts-accuracy
bi-encoding 0.519 0.101
cross-encoding 0.622 0.428

From the result above, cross-encoding significantly improved the accuracy on both para dataset dev dataset and sts dev dataset, since the method better captures the relationship between the two sentences. Therefore, we used cross-encoding for future explorations.

Due to the difference in the size of different datasets, we experimented with two sampling approaches, round robin sampling and annealed sampling, on the three tasks. The results in a specific full model fine tuning process are shown below:
method sst-accuracy para-accuracy sts-accuracy
round-robin 0.451 0.766 0.813
annealed 0.469 0.805 0.822

The result revealed the improvement on accuracy using annealed sampling method, especcially for the para dataset. The time consumed was approximately the same for both method. Therefore, we used annealed sampling for the rest of the experiments.

Considering that the goal of our model was to improve the performance on all the three downstream tasks, we added PAL layers specific to each task in parallel to the BERT layer. The result of a fine tuning process was shown below:
method sst-accuracy para-accuracy sts-accuracy
without-PAL 0.488 0.805 0.841
with-PAL 0.492 0.810 0.843

5.4 Results

The best model was mainly composed of PAL layers, annealed sampling, cross-encoding, transferring the parameters of the baseline to the full model, SMART regularization. The result of the model on the dev set are (baseline result in the parenthesis):
SST dev accuracy: 0.506 (0.383)
Paraphrase dev accuracy: 0.818 (0.622)
STS dev correlation: 0.875 (0.427)

The result of the model on the test set are:
SST test accuracy: 0.531
Paraphrase test accuracy: 0.819
STS test correlation: 0.869
Overall test score: 0.762

Among all the enhancement, the most remarkable improvement was to turn to cross-encoding, which boosted the accuracy on sts dataset by over 0.3. With the sentence concatenated, the model was able to capture the internal dependency as well as broader context. Another improvement benefit from transfer learning, which is to transfer the parameters learnable in the baseline. These top layers are trained to learn the patterns of each specific task. By transfering them to the full model, the model performed better than random initialization. The annealed sampling also improved the accuracy by mitigating the overfitting and underfitting given the imbalanced dataset. The improvement brought by these techniques are within expectation.

One surprise was that the PAL layers did not boosted the performance as expected, even though they do enhanced the accuracy a bit. One guess might be that the transfered top layers have already capture the specific pattern such that the PAL layers were not able to learn much additionally. Another thing was that the gradient surgery technique did not simultaneously improve the performance on all the tasks. Sometimes, the accuracy was even lower than the model without any gradient treatment approaches.

6 Analysis

For sentiment classification, the model is able to correctly predict the most of the sentiments. However, there's difficulty in distinguish neighboring sentiment labels. One explanation might be that human beings who label the remarks might be affected by the variation in their judgements.

For paraphrase detection, the model has difficulty with sentence pairs that are highly similar in overall meanings, and mistakenly classifies them as paraphrase. For example, "What are some simple ways to save money?" and "What are some simple ways of saving money for your dream trip?" pair are misclassified as paraphrase. The model is limited in capturing minor differences when the the pair are highly alike. We may consider to alter the attention mechanism to learn the detailed discrepancies between the pairs.

For the semantic textual similarity, generally the correlation is strong. But sometimes the model failed when the words from the two sentences overlap. For example, "Work into it slowly." and "It seems to work." are seen as similar by the model. The model may not actually understand the meaning of the sentence, just capture the meaning of single words. We may consider either changing the attention mechanism or alter the architecture.

7 Conclusion

In this project, we explored a bunch of techniques to improve the pre-trained model's performance on three downstream tasks, including bi-encoding, cross-encoding, round robin sampling, annealed sampling, transfering pre-trained weights, SMART regularization, projected attention layers, ordinal

log loss functions and gradient surgery. We demonstrated that cross-encoding, annealed sampling, transfer learning, PAL contributed to the overall performance of the model. We also analyzed the limitation of the model qualitatively and put forwarded possible solutions for refinement. We may do more improvement like incorporating different layers like RNN, modifying the loss functions to better capture the inter-sentence relationship, and using larger pre-trained models.

8 Ethics Statement

While the three models are powerful in analyzing human languages, they pose several ethical challenges. We illustrate our concern with sentiment analysis.
Sentiment analysis models can inherit biases present in the data they are trained on. If the training data is skewed towards a particular demographic, culture, or language, the model may not accurately represent the sentiments of diverse groups. This can lead to unfair or discriminatory outcomes. Analyzing sentiment often involves processing personal data from social media posts, reviews, or other sources. Ensuring that this data is handled ethically, with proper consent and in compliance with privacy regulations, is crucial to protect individuals' privacy rights. If sentiment analysis is used to censor or suppress certain opinions or expressions deemed as "negative," it can infringe on individuals' rights to free speech and expression. Balancing the need to moderate harmful content with preserving freedom of expression is a significant ethical consideration.
Addressing these ethical challenges requires careful consideration of data collection and processing practices, model design and evaluation, as well as regulatory frameworks to ensure that sentiment analysis technology is used responsibly and ethically.

References

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. 2020. Gradient vaccine: Investigating and improving multitask optimization in massively multilingual models. CoRR, abs/2010.05874.
Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020a. Gradient surgery for multi-task learning.
Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning.
Weizhu Chen Xiaodong Liu Jianfeng Gao Haoming Jiang, Pengcheng He and Tuo Zhao. 2019. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In arXiv preprint arXiv:1911.03437.
Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
François Castagnos, Martin Mihelich, and Charles Dognin. 2022. A simple log-based loss functionfor ordinal text classification. In Proceedings of the 29th International Conference on Computational Linguistics, pages 4604–4609, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.