EEE 586

Assignment 2

Pouya Ghahramanian (21804034)

May 2023

1. Abstract

In this assignment, we explore the Corpus of Linguistic Acceptability (CoLA) dataset, one of the GLUE Benchmark datasets, for text classification using BERT-based models. First, we fine-tune a pretrained BERT model (bert-base-uncased) on the GLUE dataset and tune learning rate, number of epochs, max length of input sequence, and dropout hyperparameters using the Optuna library. We use the loss and the Matthew Correlation Coefficient (MCC) as the evaluation metrics. We push our best-performing model to the Hugging Face Hub. Then, we explore an alternative approach to using ['CLS'] tokens as document representations. Similar hyperparameter investigations are conducted.

2. Introduction

Text classification is a fundamental task in natural language processing with applications ranging from sentiment analysis to document categorization. With the advent of pretrained language models like BERT (Bidirectional Encoder Representations from Transformers), significant advancements have been made in achieving state-of-the-art performance on various NLP benchmarks. In this assignment, we delve into the Corpus of Linguistic Acceptability (CoLA) dataset, a part of the General Language Understanding Evaluation (GLUE) benchmark, to explore the effectiveness of BERT-based models for text classification.

To begin our exploration, we fine-tune a pretrained BERT model, specifically the "bert-base-uncased" variant, on the GLUE dataset. This process involves adapting the model to the specific characteristics and requirements of the CoLA dataset through a process known as fine-tuning.

In order to optimize the performance of our fine-tuned BERT model, we tune several hyperparameters using the Optuna library. These hyperparameters include the learning rate, number of epochs, maximum length of input

sequences, and dropout at the classification head. Optuna provides an efficient framework for hyperparameter optimization, enabling us to search the hyperparameter space and identify the optimal configuration that yields the best performance.

To evaluate the performance of our models, we utilize both the loss function and the Matthew Correlation Coefficient (MCC) as evaluation metrics. The loss function provides an indication of the model's training progress and convergence, while the MCC takes into account both true positives and true negatives, making it a suitable metric for imbalanced datasets like CoLA.

To ensure the reproducibility and accessibility of our best-performing model, we leverage the Hugging Face Model Hub. By pushing our model to the Hugging Face Hub, we make it readily available for others in the research and NLP community to use and build upon, fostering collaboration and enabling future advancements in text classification.

Additionally, we explore an alternative approach to utilizing BERT models for text classification. Instead of solely relying on the "[CLS]" tokens as document representations, we investigate an alternative technique by using output of the all hidden states in the BERT model for document representations. We conduct a similar hyperparameter investigation, examining the effects of learning rate, maximum input sequence length, and number of epochs..

In the next section, we describe our architectures in details. We present our results in section 4, and conclude the report in section 5.

3. Architecture

For the first part I used the 'bert-base-uncased' model from HuggingFace and fine-tuned it on the GLUE dataset.

In the second part of the assignment we are required to use an alternative approach instead of using the ['CLS'] tokens for document representations. In this part, I use an alternative approach by taking the average of the hidden states of all tokens in the input sequence. Then, I use the average embeddings to a classification layer for final prediction. In other words, instead of solely relying on the [CLS] token representation for the classification task, we compute the average of the hidden states of all tokens in the input sequence to create a fixed-size document representation. This representation is then passed through a linear classifier to produce the final output. To this aim, I wrote a python class and named it as MeanPoolingBert that extends

BertForSequenceClassification model. I used the 'bert-base-uncased' model as the base model and modified the forward method to use the mean vector of the hidden states as the embedding vector.

The intuition behind this approach is that by averaging the hidden states of all tokens, the model can capture more meaningful information from the entire input sequence, which can potentially lead to better classification performance.

4. Results

In this section, we report our results in terms of loss, MCC score and optimization curves for the first and second part of the assignment.

Part 1

I used the following hyperparameters to fine-tune the model without parameter optimization:

• Batch Size: 64 • Learning Rate: 1e-3 • Training Epochs: 3 • Dropout Probability: 0.2 • Max Sequence Length: 128

The loss and Matthews correlation score of the model are obtained as 0.4742 and 0.5215, respectively. I pushed the fine-tuned model to the Hugging Face model hub available at this link.

Then, I used the Optuna library to find the best-performing model with hyperparameter search. I used the following hyperparameter spaces with 10 trials in the hyperparameter search:

• Training Epochs: [1, 10] • Learning Rate: (1e-6, 1e-3) • Max Sequence Length: (32, 128) • Dropout Probability: (0.2, 0.5)

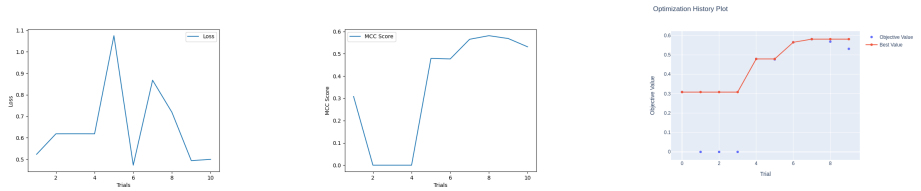
The hyperparameter values for the best performing model is obtained by Optuna as follows:

• Training Epochs: 5 • Learning Rate: 7e-5 • Max Sequence Length: 74 • Dropout Probability: 0.48

I used Matthews score as objective to tune the hyperparameters. The loss and Matthews correlation score of the fine-tuned model after hyperparameter tuning with Optuna are obtained as 0.71 and 0.58.

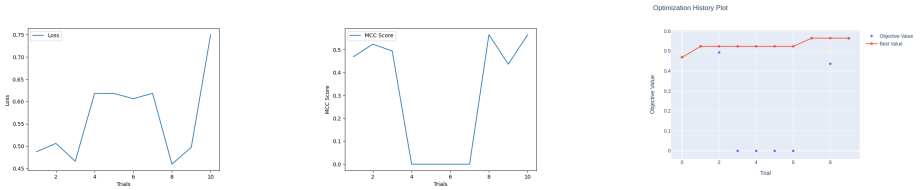
Figure 1 shows the loss, MCC score and optimization history plot over trials with optuna.

We observe that the best performing model is obtained at trial 7 as we get the highest MCC score.



(a) Loss over trials (b) MCC score over trials (c) Optimizatin history

Figure 1: Loss, MCC score and optimization plot over trials for part 1.



(a) Loss over trials (b) MCC score over trials (c) Optimizatin history

Figure 2: Loss, MCC score and optimization plot over trials for part 2.

Part 2

For the second part, I used the following hyperparameters to fine-tune the model without parameter optimization:

• Batch Size: 64 • Learning Rate: 2e-5 • Training Epochs: 2 • Dropout Probability: 0.2 • Max Sequence Length: 128

The loss and Matthews correlation score of the model with costum classifier are obtained as 0.4949 and 0.4341, respectively. I pushed the fine-tuned model to the Hugging Face model hub available at this link.

I used the similar hyperparameter spaces to find the best performing model. The optimal hyperparameters are obtained by Optuna as follows:

• Training Epochs: 5 • Learning Rate: 1e-4 • Max Sequence Length: 91 • Dropout Probability: 0.36

The loss and Matthews correlation score of the fine-tuned model after hyperparameter tuning with Optuna are obtained as 0.5653 and 0.4596.

Figure 2 shows the loss, MCC score and optimization history plot over trials with optuna. For this part, the best MCC score is obtained at trial 7, indicating the best-performing model.

In the next section, we discuss the obtained results and conclude the report.

5. Discussions and Conclusions

In this assignment, we investigated the performance of BERT-based models on the Corpus of Linguistic Acceptability (CoLA) dataset, a part of the GLUE Benchmark datasets, for text classification tasks. We fine-tuned a pretrained BERT model (bert-base-uncased) and optimized its hyperparameters, including learning rate, number of epochs, max length of input sequence, and dropout using the Optuna library. The evaluation metrics used in this assignment were the loss and the Matthew Correlation Coefficient (MCC).

We observed that for both the primary approach using [CLS] tokens as document representations and our alternative method, the Optuna hyperparameter tuning effectively improved the performance of the models in terms of Matthew scores. Interestingly, the loss values increased for the best-performing models in both parts. This can be attributed to the fact that we used the Matthew score as our objective in hyperparameter tuning with Optuna.

Due to time constraints, we limited our experiments with Optuna to 10 trials in each part to find the best-performing model and optimal hyperparameters. Nevertheless, an analysis of the optimization history plot suggested that using a higher number of trials could potentially lead to even better hyperparameter tuning, as the objective score could continue to increase.

A comparison between the default 'bert-base-uncased' model using the [CLS] token as document representation and our modified classifier in the alternative approach revealed that the former outperformed the latter. This indicates that our alternative approach did not effectively improve the performance of the model in the CoLA task.

In conclusion, this assignment demonstrated the effectiveness of BERT-based models in the CoLA task and the importance of hyperparameter tuning in achieving high performance. We published our models on the Hugging Face model hub, for which the links are given in the previous section.