A New Ensemble Method for Small Data Fine-tuning

Jiasen Li
Shanghai Jiao Tong University
Shanghai, China
LIJIASEN0921@SJTU.EDU.CN

ABSTRACT

The Corpus of Linguistic Acceptability (CoLA) is a very common grammatical acceptability classification task in the Natural Language Processing. In previous methods, a single model is adopted and fine-tuned, which is likely to over-fit on its small data and does not work quite well on very small data set. In this paper, I proposed a special kind of ensemble method for this case, which can lead to about 2% improvement in CoLA.

CCS CONCEPTS

- Computing Methodologies → Nature Language Processing

KEYWORDS

Corpus of Linguistic Acceptability, Ensemble, Pre-trained Model Fine-tuning, Small training data

ACM Reference Format

Jiasen Li. 2020. An Ensemble Method for GLUE-COLA. https://doi.org/......

1. Introduction

The Corpus of Linguistic Acceptability (CoLA) contains about 10657 sentences. They comes from 23 publications and are annotated by experts for their acceptability. It is not a very easy task not only because the feature according which we predict the acceptability is not easy to be learnt by machine learning, but also because its training data is too small to train a larger model.

Previous works, like Bert [1], use pre-trained model fine-tuning to enlarge its model's model with the idea of transfer learning. This method can improve a lot against its small data because it can learn CoLA's Complicated Pattern with more other data.

However, the Bert Series have the problem that it may over-fit. With Bert Layers frozen, it cannot learn anything about CoLA. With Bert Layers active, it will certainly over-fit after a large number of steps of training. However, early stop cannot solve it well, because early stop may not exploit all learnable feature in CoLA and its random training order may also take in randomness. Also, traditional Ensemble does not work well due to the bias of training data learnt by the model.

In this paper, I proposed a new ensemble method for small data fine-tuning. This method can exploit the training data when not over-fitting the training data.

2. Related Work

Before BERT, fine-tuning technique was not very common in Nature Language Processing. Every new task requires writing a new model according to the specific task. After BERT [2] was proposed, fine-tuning becomes very common. Nearly all high-score models in GLUE [3] are pre-trained models.

BERT Fine-tuning is a good solution compared to train a new model when dealing with a small training data, because the size of data may not support the size of the model, which means the model needs more data from other sources. However, fine-tuning Bert may require tricks. It is hard to balance the training steps in fine-tuning. Too large steps may cause serious over-fitting on small training. Too small steps may not enable the fine-tuned model to exploit the data.

ALBERT [4] partly solves this problem by cutting the number of parameters. With fewer parameters, the model is less likely to over-fit on small fine-tuning training data. However, it loses it accuracy. ALBERT needs more layers to acquire the same score as BERT.

RoBerta [5] optimized more robustly, but it did not care about the problem in small fine-tuning data.

3. Method

3.1. Model

I used the Electra Model [6] as my base model. I ensemble many fine-tuned RoBerta Large and acquire a large improvement.

3.2. Very Early Stop

In order not to over-fit, I adopt the "very early stop" strategy. I stop the model even before its evaluation score reaches the highest point.

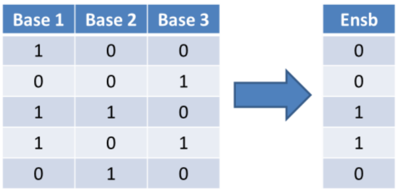
In fine-tuning, the maximum evaluation score comes later than the minimum loss by several epochs. It is true that the model at the epoch with the highest evaluation score is more likely to acquire a higher test score. However, it may slightly over-fit on the training set. The higher test score comes from removing the variance of the model and taking in more bias of the training data.

I do not have to do the trade-off. I can remove the variance through ensemble method instead of through training more steps and taking in more bias. And very early stop can also enable the order change to add randomness in the prediction, which can help the ensemble.

3.3. Equal Voting

I used voting to ensemble. I directly predict 0 and 1 with the base models and let them vote equally to get the final decision.

Traditional Ensemble Method is to take the average value before the softmax function. However, it may fail when one model is convinced of a false answer, because it may give a very high score to the false answer, requiring more working models to bring the true answer back.



Base 1	Base 2	Base 3
1	0	0
0	0	1
1	1	0
1	0	1
0	1	0

Ensb
0
0
1
1
0

Equal voting can avoid this situation. A model that does not work on one entry of the test data may more affect the answer too much. This voting can help enlarge the ensemble size.

3.4. Warm Up Steps

Lower Learning Rate can help the model converge more stably to an acceptable score. However, it cuts the variance of the model and takes in the bias of the training data. In order to prevent the model from taking in too much bias, I let the learning rate of the last epochs be higher to add variance. This is the same trick as warm up, which helps the ensemble.

4. Experiments and Results

4.1. Experiments

I fine-tune the Electra Large Discriminator with learning rate=1e-5, warm up step=320, max step=374, batch=32*8. The Matthew's Corrs of acceptable models vary between 68 and 69.

Model	Matthew's Corr
Electra1	68.0
Electra2	68.4
Electra3	67.8
Ensemble-3 without Selection	70.4
Ensemble-13 with Selection	**72.9**

I manually select the acceptable models according to its MC score. In order words, if the MC score is too low, the model will not be taken into ensemble. There are 34 fine-tuned Electra Models, and I select the 13 Models with MC score larger than 68.0.

I ensemble the 13 Models with equal voting and acquire the final test result with MC score 72.9.

Also, random selection can obtain the score 70.4.

4.2. Results

Model	Matthew's Corr
BiLSTM+Attn	18.6
BERT	59.2
Human Baseline	66.4
Roberta	67.8
FreeLB-Roberta + Ensemble	68.0
Albert + Ensemble	69.1
XLNet + Ensemble	70.2
Electra Large + Tricks	71.7
Electra	68.0-69.2
Electra + My Ensemble	**72.9**

Generally, traditional ensemble can only improve within 0.5 in the small dataset fine-tuning. See Roberta and Roberta Ensemble. This is because traditional ensemble has the fine-tuned model be biases by the training data dramatically.

My ensemble technique works better than all the tricks adopted by Electra Team by 1.2%. And my ensemble can improve about 4% from the Standard Electra with no tricks.

5. Conclusion and Future Work

5.1. Conclusion

This paper proposed a new method in ensemble which can improve more than other tricks. My ensemble outperform any other known tricks in the CoLA benchmark.

My method can also be applied to any transfer learning or any pre-train model fine-tuning with a small dataset. It does not hurt to apply the method on a large dataset.

Compared to using a larger pre-trained model, my method can more easily run parallel and run faster.

Compared to using more tricks, my method indicates "simple often wins".

My method raises a new type of method against the bias of the dataset in fine-tuning and transfer learning.

5.2. Future Work

There are still more work to do. I only tested on CoLA due to the limit of time and device. Some other Benchmarks and other models in Glue will be tested with my ensemble later. Also, I will try to do compression to the ensemble model to make it smaller.

Reference

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" arXiv:1810.04805

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" arXiv:1810.04805

[3] Wang, Alex and Singh, Amanpreet and Michael, Julian and Hill, Felix and Levy, Omer and Bowman, Samuel R. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding" ICLR2019

[4] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations" arXiv:1909.11942

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov "RoBERTa: A Robustly Optimized BERT Pretraining Approach" arXiv:1907.11692

[6] Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators" ICLR2020