22S2 AI6103 Deep Learning Application Final Project Report

Dai Xinrui G2204611C - DAIX0016@e.ntu.edu.sg
Li Ji G2204609C - jli116@e.ntu.edu.sg
Jin Hui G2204806E - HJIN003@e.ntu.edu.sg
Github Repository:https://github.com/Li-Ji-02/AI6103_Group_Project_2023Spring

Abstract

Deep learning has achieved significant progress with large-scale neural networks excelling across various domains. However, their extensive computational and storage demands limit applicability on resource-constrained devices. Our study explores optimization approaches, including knowledge distillation and pretrained embeddings, to improve smaller-scale model performance without substantially increasing resource requirements. These methods possess potential applicability across all AI fields. Our findings provide valuable insights for implementing deep learning applications on resource-limited devices.

Introduction

Deep learning has gained widespread adoption across various domains, with pre-trained models like BERT(Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova 2018)delivering remarkable results in natural language processing tasks. Nonetheless, the extensive number of parameters in these large-scale pre-trained models complicates their deployment on computationally constrained low-performance devices. This challenge is commonly faced by newcomers in the AI field, and many researchers, including ourselves, hope to address it. Consequently, to balance performance and model complexity, researchers have explored efficient model architectures and training methodologies. Knowledge distillation(Tao Huang, Shan You, Fei Wang, Chen Qian, Chang Xu 2022)and model fusion have emerged as highly effective strategies for tackling this issue.

In this study, we first explore model fusion by integrating BERT into a classic text classification model, VDCNN(Alexis Conneau, Holger Schwenk,and Yann Le Cun 2017), achieving substantial performance improvements. With BERT's weights remaining frozen, the training cost remains reasonably low. Subsequently, we combine the enhanced BERT-VDCNN model with knowledge distillation techniques to transfer knowledge to a lightweight simple CNN model. Our approach demonstrates that, through proper optimization and techniques, it is possible to boost the performance of lightweight models without significantly increasing resource demands.

Experiment

Dataset

We selected the AGnews dataset for our study. This large-scale text classification dataset covers four categories: World, Sports, Business, and Science & Technology. Crucially, AGnews includes 120,000 news articles in the training set and 7,600 separate articles for testing. The distribution of articles within the training and test sets is approximately balanced across each category, facilitating a comprehensive evaluation of the model's performance across all categories.

Data Preprocessing

In this phase, input texts and labels are extracted from the dataset. Texts are tokenized using the BERT tokenizer, applying padding, truncation, and a maximum token length of 64. Tokenized inputs and labels are then converted into tensors. The processed dataset is created as a list of zipped $input\_ids$, $attention\_mask$, $token\_type\_ids$, and $labels$. A collate helper function is written to handle data batches and combine their elements into single tensors. It stacks $input\_ids$, $attention\_mask$, and $token\_type\_ids$ tensors and converts labels into a long tensor. Finally, PyTorch dataloaders are created and initialized, ready for training and testing.

Model Constructions

In this experiment, we design and implement six models as follows:

- Model 1 & 2-Simple CNN: A basic Convolutional Neural Network (CNN) containing four convolutional layers. These models are trained with and without a teacher to examine the impact of knowledge distillation techniques.
- Model 3 & 4-Simpler CNN: An even simpler CNN with only one convolutional layer. These models are trained in the same manner as Models 1 and 2 and are primarily for comparison purposes.
- Model 5-Standard VDCNN: A standard VDCNN model with a depth of 9. We reference its expected performance documented in the paper "Very Deep Convolutional Networks for Text Classification," published in 2017, without training and testing it.

- Model 6-BERT-embedded VDCNN: A BERT-integrated VDCNN model with a depth of 9, achieving outstanding performance. It is used as the teacher model in our experiment. The implementation involves using BERT as the VDCNN's embedding layer and freezing its weights.

Loss Function

To implement Knowledge Distillation techniques, we use a loss function called DISTloss(2) that relates the teacher's prediction to the student's, enabling the student to be guided by the more advanced teacher model. We first write a method to calculate the Pearson Correlation, which is then used to compute the inter-class relation and cosine similarity between the student's and teacher's predictions y'. DISTLoss includes the following crucial calculations: Proportioned student and teacher predictions y_s and y_t from the original predictions:

$$y_s = \operatorname{softmax}\left(z_s/\mathrm{tau}\right) \tag{1}$$

$$y_t = \operatorname{softmax}\left(z_t/\ \mathrm{tau}\right) \tag{2}$$

Inter and intra-class relations:

$$\begin{aligned} inter = 1 - cosineSimiarity\left(y_s - mean\left(y_s\right),\right. \\ \left. y_t - mean\left(y_t\right), eps\right) \end{aligned} \tag{3}$$

$$intra = inter\left(y_{s'} transpose(0,1), y_{t'} transpose(0,1)\right) \tag{4}$$

The overall dist loss, where CE is the classic cross-entropy loss of the student prediction:

$$\text{dist loss } = CE + \text{ beta }^{*} \text{ inter } + \text{ gamma } * \text{ intra} \tag{5}$$

Other Experiment Settings

- Learning rate: We set the learning rate (lr) to 0.001. A smaller learning rate ensures that the model converges more stably during the optimization process, thereby avoiding drastic fluctuations in the loss function's optimization.
- Optimizer: We choose the Adam optimizer as the optimization algorithm for the training process. The Adam (Adaptive Moment Estimation) optimizer combines the advantages of momentum and RMSProp, enabling adaptive adjustments of the learning rate, which in turn allows for faster convergence rates and higher stability during the training process.
- Training epochs: We set the number of training epochs for every student models to 20 and the number of training epochs for BERT-VDCNN to 30.

Results

Evaluation Metric

The accuracy on AGNews' default testing dataset is our primary evaluation metric, which will be called test accuracy in the following results section. For comparison, we've trained and tested several CNN models with different settings, as described in above sections. The performances of these models are recorded down below using test accuracy for performance measure.

Performance Records

- Model 1 - Simple plain CNN

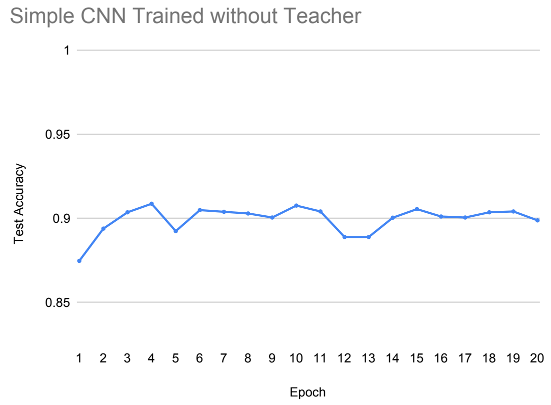

Figure 1: Simple CNN Trained without Teacher

Model	Test Accuracy(in percentage)
Simple CNN Plain	90.87%

Table 1: Best Test Acc of Simple CNN Plain

- Model 2 - Simple CNN with Teacher

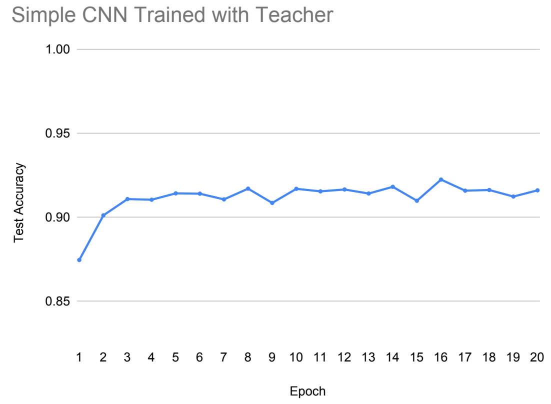

Figure 2: Simple CNN Trained with Teacher

Model	Test Accuracy(in percentage)
Simple CNN with Teacher	92.25%

Table 2: Best Test Acc of Simple CNN with Teacher

- Model 3 - Simpler plain CNN

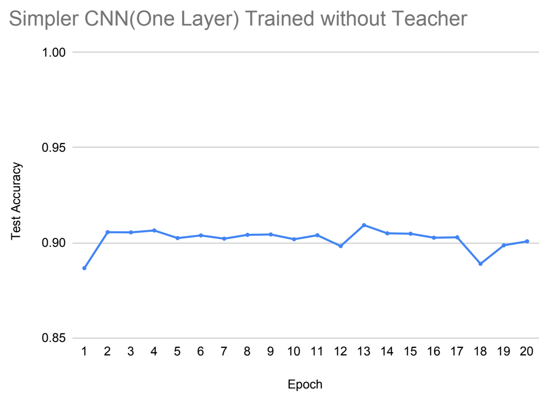

Figure 3: Simpler plain CNN(One Layer)

Model	Test Accuracy(in percentage)
Simple CNN Plain	90.92%

Table 3: Best Test Acc of Simpler CNN Plain(One Layer)

- Model 4 - Simpler CNN with Teacher

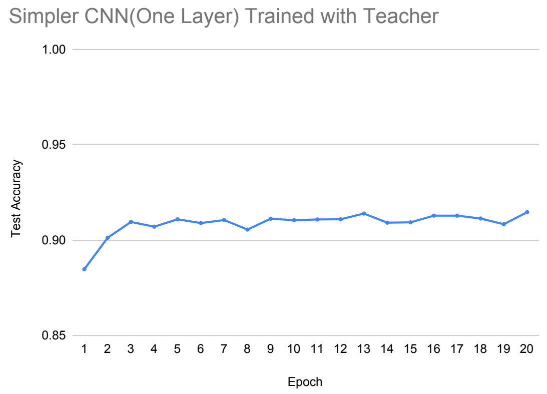

Figure 4: Simpler CNN with Teacher(One Layer)

Model	Test Accuracy(in percentage)
Simple CNN with Teacher	91.46%

Table 4: Best Test Acc of Simpler CNN with Teacher(One Layer)

- Model 5 & 6 - plain VDCNN and VDCNN with BERT embeddings

Depth-9 Models	Test Accuracy(in percentage)
VDCNN	90.83%
BERT-VDCNN	93.26%

Table 5: VDCNN and BERT-VDCNN Test Accuracy

Analysis

In our previous project, we found that when fine-tuning the pretrained BERT, tandem a CNN model with several layers gave better performance than just tuning the parameters of the final fully connected layer of the BERT. Moreover, this approach is much less computationally complex than fine-tuning the entire BERT model. Therefore, this tandem approach is a good compromise. As can be seen in Table 5 in the results section, this approach gives better performance than just tuning the fully connected layer of pretrained BERT and VDCNN itself.

In the loss we use, the inter term controls the distribution of the output of the student model as close as possible to the output of the teacher model, and the intra term controls the order of the probabilities of belonging to each class in the output of the student model as close as possible to the output of the teacher model. In this way, the student model effectively imparts knowledge from the teacher model, and the experimental results show that the performance of the 4-layer cnn and 1-layer CNN improves by 1.38% and 0.54%, respectively, to 92.25% and 91.46% when using the teacher model with an accuracy of 93.26%.

In conducting this experiment, we tried a range of models and found that we could not obtain a meaningful performance improvement with DISTLoss when the size difference between the teacher model and the student model was too large. We suspect that this is because the student model cannot accurately describe the output of the teacher model for the same input when the difference in capacity between the teacher and student models is too large (think of it simply as the teacher model having a two-dimensional output space, but the student output space being one-dimensional). This phenomenon can also be seen in our experimental results, where the improvement in accuracy is small when the student model is a 1-layer CNN.

Limitation

- We verified the effectiveness of DISTloss on the text classification task, but since the gap in accuracy between the student model and the teacher model in our experiments is not significant, We cannot estimate how much performance improvement DISTLoss can bring to the student model.
- As we mentioned in the previous section, DISTLoss does not provide a meaningful performance gain if the difference in model capacity is too large. There may be a maximum value for the difference in capacity between the student and teacher models such that a large performance gain is obtained from the teacher model while the size of the student model is as small as possible. We have not explored this aspect.
- In DISTLoss, we set beta (the weight of the inter term) and gamma (the weight of the intra term) to 2, and the weight of the CE term to 1. These are hyperparameters, we can know that the output of the student model will be more dependent on the output of the teacher model when the values of beta and gamma are larger, and more dependent on the output of the teacher model when the weight

of the CE term is larger. There may be optimal values for improving the performance of the student model (and even outperforming the teacher model), but we have not explored this aspect.

Possible Application

Our current project is to improve the performance of a small model. It would be important if the performance of the small model could be approximated to the performance of the large model. Since such small models require very little computing power and memory, we can apply the small models trained using this technique to:

- Devices with small computing power: face recognition devices, surveillance devices, small mobile phones, etc.
- Scenarios where cloud computing is not possible: some small devices cannot use cloud computing due to pool signal or privacy reasons, e.g. face recognition devices in private places, animal detectors in forests.

Conclusion

In summary, we have successfully improved the performance of small models on text classification tasks by taking advantage of relatively new transfer learning techniques in computer vision.

References

Alexis Conneau, Holger Schwenk,and Yann Le Cun. 2017. Very Deep Convolutional Networks for Text Classification. https://arxiv.org/abs/1606.01781.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. 2017. Attention Is All You Need. https://arxiv.org/abs/1706.03762.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805.

Tao Huang, Shan You, Fei Wang, Chen Qian, Chang Xu. 2022. Knowledge Distillation from A Stronger Teacher. https://arxiv.org/abs/2205.10536.