Improving BERT Fine-Tuning via Self-Ensemble and Self-Distillation

Yige Xu[1] , **Xipeng Qiu**[1] , **Ligao Zhou**[2] and **Xuanjing Huang**[1]
[1]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, China
[2]Huawei Technologies Co., Ltd.
{ygxu18, xpqiu, xjhuang}@fudan.edu.cn, zhouligao@huawei.com,

Abstract

Fine-tuning pre-trained language models like BERT has become an effective way in NLP and yields state-of-the-art results on many downstream tasks. Recent studies on adapting BERT to new tasks mainly focus on modifying the model structure, re-designing the pre-train tasks, and leveraging external data and knowledge. The fine-tuning strategy itself has yet to be fully explored. In this paper, we improve the fine-tuning of BERT with two effective mechanisms: *self-ensemble* and *self-distillation*. The experiments on text classification and natural language inference tasks show our proposed methods can significantly improve the adaption of BERT without any external data or knowledge.

1 Introduction

The pre-trained language models including BERT [Devlin *et al.*, 2018] and its variants (XLNet [Yang *et al.*, 2019] and RoBERTa [Liu *et al.*, 2019b]) have been proven beneficial for many natural language processing (NLP) tasks, such as text classification, question answering [Rajpurkar *et al.*, 2016] and natural language inference [Bowman *et al.*, 2015]. These pre-trained models have learned general-purpose language representations on a large amount of unlabeled data, therefore, adapting these models to the downstream tasks can bring a good initialization for and avoid training from scratch. There are two common ways to utilize these pre-trained models on downstream tasks: feature extraction (where the pre-trained parameters are frozen), and fine-tuning (where the pre-trained parameters are unfrozen and fine-tuned). Although both these two ways can significantly improve the performance of most of the downstream tasks, the fine-tuning way usually achieves better results than feature extraction way [Peters *et al.*, 2019]. Therefore, it is worth paying attention to find a good fine-tuning strategy.

As a widely-studied pre-trained language model, the potential of BERT can be further boosted by modifying model structure [Stickland and Murray, 2019; Houlsby *et al.*, 2019], re-designing pre-training objectives [Dong *et al.*, 2019; Yang *et al.*, 2019; Liu *et al.*, 2019b], data augmentation [Raffel *et al.*, 2019] and optimizing fine-tuning strategies with external knowledge [Liu *et al.*, 2019a; Sun *et al.*, 2019]. However, the fine-tuning strategy itself has yet to be fully explored. Sun *et al.* [2019] investigated different fine-tuning strategies and hyper-parameters of BERT for text classification and showed the "further-pretraining" on the related-domain corpus can further improve the ability of BERT. Liu *et al.* [2019a] fine-tuned BERT under the multi-task learning framework. The performance of BERT on a task could benefit from other related tasks. Although these methods achieve better performance, they usually need to leverage external data or knowledge.

Adapter, PALs

In this paper, we investigate how to maximize the utilization of BERT by better fine-tuning strategy without utilizing the external data or knowledge. BERT is usually fine-tuned by using stochastic gradient descent (SGD) method. In practice, the performance of fine-tuning BERT is often sensitive to the different random seeds and orders of training data, especially when the last training sample is noise. To alleviate this, the ensemble method is widely-used to combine several fine-tuned based models since it can reduce the overfitting and improve the model generalization. The ensemble BERT usually achieves superior performance than the single BERT model. However, the main disadvantages of the ensemble method are its model size and training cost. The ensemble model needs to keep multiple fine-tuned BERTs and has a low computation efficiency and high storage cost.

We improve the fine-tuning strategy of BERT by introducing two mechanisms: *self-ensemble* and *self-distillation*.

(1) **Self-Ensemble**. Motivated the success of widely-used ensemble models, we propose a self-ensemble method, in which the base models are the intermediate BERT models at different time steps within a single training process [Polyak and Juditsky, 1992]. To further reduce the model complexity of the ensemble model, we use a more efficient ensemble method, which combines several base models with parameter averaging rather than keeping several base models.

(2) **Self-Distillation**. Although the self-ensemble can improve the model performance, the training of the base model is the same as the vanilla fine-tuning strategy and cannot be affected by the ensemble model. We further use knowledge distillation [Hinton *et al.*, 2015] to improve fine-tuning efficiency. At each time step in training, the current BERT model (called *student model*) is learned with two teachers: the gold labels and self-ensemble model (called *teacher model*). The teacher model is an average of student models at previous time steps. With the help of the teacher model, the student is more robust

and accurate. Moreover, a better student model further leads to a better teacher model. A similar idea is also used in semi-supervised learning, such as Temporal Ensembling [Laine and Aila, 2016] and Mean Teacher [Tarvainen and Valpola, 2017]. Different from them, our proposed self-distillation aims to optimize the student model without external data.

The experiments on text classification and natural language inference tasks show our proposed methods can reduce the test error rate by more than 5.5% on the average on seven widely-studied datasets.

The contributions of this paper are summarized as follows:

- We show the potential of BERT can be further stimulated by a better fine-tuning strategy without leveraging external knowledge or data.
- The self-ensemble method with parameter averaging can improve BERT without significantly decreasing the training efficiency.
- With self-distillation, the student and teacher models can benefit from each other. The distillation loss can also be regarded as a regularization to improve the generalization ability of the model.

2 Related Work

We briefly review two kinds of related work: pre-trained language models and knowledge distillation.

2.1 Pre-trained Language Models

Pre-training language models on a large amount of unlabeled data then fine-tuning in downstream tasks has become a new paradigm for NLP and made a breakthrough in many NLP tasks. Most of the recent pre-trained language models (e.g., BERT [Devlin *et al.*, 2018], XLNet [Yang *et al.*, 2019] and RoBERTa [Liu *et al.*, 2019b]) are built with Transformer architecture [Vaswani *et al.*, 2017].

As a wide-used model, BERT is pre-trained on *Masked Language Model Task* and *Next Sentence Prediction Task* via a large cross-domain unlabeled corpus. BERT has two different model size: $\mathrm{BERT}_{\mathrm{BASE}}$ with a 12-layer Transformer encoder and $\mathrm{BERT}_{\mathrm{LARGE}}$ with a 24-layer Transformer encoder. Both of them take an input of a sequence of no more than 512 tokens and outputs the representation of the sequence. The sequence has one segment for text classification task or two for text matching task. A special token `[CLS]` is added before segments, which contain the special classification embedding. Another special token `[SEP]` is used for separating segments.

Fine-tuning BERT can deal with different natural language tasks with task-specific output layers. For text classification or text matching, BERT takes the final hidden state $\mathbf{h}$ of the first token `[CLS]` as the representation of the input sentence or sentence-pair. A simple softmax classifier is added to the top of BERT to predict the probability of label y:

$$p(y|\mathbf{h}) = \mathrm{softmax}(W\mathbf{h}), \tag{1}$$

where W is the task-specific parameter matrix. The cross-entropy loss is used to fine-tune BERT as well as W jointly.

2.2 Knowledge Distillation for Pre-trained Models

Since the pre-trained language models usually have an extremely large number of parameters, they are difficult to be deployed on the resource-restricted devices. Several previous works leverage the knowledge distillation [Hinton *et al.*, 2015] approach to reducing model size while maintaining accuracy, such as TinyBERT [Jiao *et al.*, 2019] and DistilBERT [Sanh *et al.*, 2019]. Knowledge Distillation aims to transfer the knowledge of a large teacher model to a small student model by training the student model to reproduce the behaviors of the teacher model.

The teacher model usually is well-trained and fixed in the processing knowledge distillation. Unlike the common way of knowledge distillation, we perform knowledge distillation in online fashion. The teacher model is an ensemble of several student models at previous time steps within the fine-tuning stage.

3 Methodology

The fine-tuning of BERT usually aims to minimize the cross-entropy loss on a specific task with stochastic gradient descent method. Due to the stochastic nature, the performance of fine-tuning is often affected by the random orderings of training data, especially when the last training samples is noise.

Our proposed fine-tuning strategy is motivated by the ensemble method and knowledge distillation. There are two models in our fine-tuning strategy: a student model is the fine-tuning BERT and a teacher model is a self-ensemble of several student models. At each time step, we further distillate the knowledge of the teacher model to the student model.

3.1 Ensemble BERT

In practice, the ensemble method is usually adopted to further improve the performance of BERT.

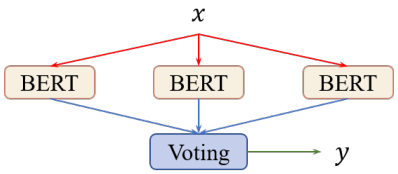
Voted BERT The common ensemble method is voting-based. We first fine-tune multiple BERT with different random seeds. For each input, we output the best predictions made by the fine-tuned BERT along with the probability and sum up the probability of predictions from each model together. The output of the ensemble model is the prediction with the highest probability.

Let $\mathrm{BERT}(x;\theta_k)$ $(1 \le k \le K)$ are K BERT models fine-tuned on a specific task with different random seeds, the ensemble model $\mathrm{BERT}_{\mathrm{VOTE}}(x;\Theta)$ is defined as

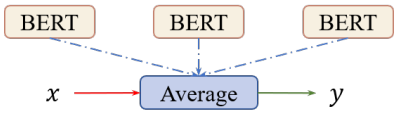
$$\mathrm{BERT}_{\mathrm{VOTE}}(x;\Theta) = \sum_{k=1}^{K} \mathrm{BERT}(x;\theta_k), \tag{2}$$

where θ_k denotes the parameters of the k-th model and Θ denotes the parameters of all the K models.

We call this kind of ensemble model as *voted BERT*. The voted BERT can greatly boost the performance of single BERT in many tasks, such as question answering [Rajpurkar *et al.*, 2016]. However, a major disadvantage of the voted BERT is it needs keeping multiple different BERTs and its efficiency is low in computing and storage.



(a) Voted BERT



(b) Averaged BERT

Figure 1: Two methods to ensemble BERT. The averaged BERT has better computational and memory efficiency then voted BERT.

Averaged BERT To reduce the model complexity of ensemble model, we use a parameter-averaging strategy to combine several BERTs into a single model, called *averaged BERT*. The averaged BERT is defined as

$$\mathrm{BERT}_{\mathrm{AVG}}(x;\bar{\theta}) = \mathrm{BERT}(x;\frac{1}{K}\sum_{k=1}^{K}\theta_k), \tag{3}$$

where $\bar{\theta}$ is the averaged parameters of K individual fine-tuned BERTs.

Figure 1 illustrates two kinds of ensemble BERTs. Since the averaged BERT is indeed a single model and has better computational and memory efficiency than the voted BERT.

3.2 Self-Ensemble BERT

Although the ensemble BERT usually brings better performance, it needs to train multiple BERTs and its cost is often expensive. To reduce the training cost, we use a self-ensemble model to combine the intermediate models at different time steps in a single training phase. We regard the BERT at each time step as a base model and combine them into a self-ensemble model.

Here we only describe the self-ensemble BERT with parameter averaging, since the voted version of self-ensemble is impracticable to keep all the intermediate models.

Let θ_t denote parameters when fine-tuning BERT at time step t, the self-ensemble BERT is defined as

$$\mathrm{BERT}_{\mathrm{SE}}(x;\bar{\theta}) = \mathrm{BERT}(x;\frac{1}{T}\sum_{\tau=1}^{T}\theta_t), \tag{4}$$

where $\bar{\theta}$ is the averaged parameters of BERTs over T time steps.

Averaging model weights over training steps tends to produce a more accurate model than using the final weights directly [Polyak and Juditsky, 1992].

3.3 Self-Distillation BERT

Although the self-ensemble can improve the model performance, the base model is trained in the same manner to the vanilla fine-tuning strategy and cannot be affected by the ensemble model. We further use knowledge distillation [Hinton *et al.*, 2015] to improve the base model. At each time step in training, the current BERT model (called *student model*) is learned from two teachers: the gold labels and the self-ensemble model (called *teacher model*). The teacher model is an average of student models at previous time steps. With the help of the teacher model, the student is more robust and accurate.

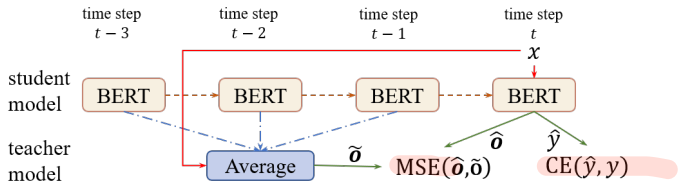


Figure 2: The proposed fine-tuning strategy $\mathrm{BERT}_{\mathrm{SDA}}$. The teacher model is the average of student models over recent K time steps, and $K = 3$ here. (x, y) is the input training sample. $\hat{y}$ is the probability of label y predicted by student model. $\hat{\mathbf{o}}$ and $\tilde{\mathbf{o}}$ are the logits output by student model and teacher model reflectively. MSE and CE denote the mean square error and cross-entropy loss.

Self-Distillation-Averaged (SDA) We first denote a fine-tuning strategy $\mathrm{BERT}_{\mathrm{SDA}}$, in which the teacher model is self-ensemble BERT with parameter averaging.

Let $\mathrm{BERT}(x,\theta)$ denote the student BERT, the objective of $\mathrm{BERT}_{\mathrm{SDA}}$ strategy is

$$\begin{aligned}\mathcal{L}_\theta(x,y) =& \mathrm{CE}\Big(\mathrm{BERT}(x,\theta), y\Big) \\ &+\lambda\mathrm{MSE}\Big(\mathrm{BERT}(x,\theta), \mathrm{BERT}(x,\bar{\theta})\Big),\end{aligned} \tag{5}$$

where CE and MSE denote the cross-entropy loss and mean squared error respectively, and λ balances the importance of two loss functions. The teacher model $\mathrm{BERT}(x,\bar{\theta})$ is a self-ensemble BERT with recent time steps. At time step t, $\bar{\theta}$ is the averaged parameters of recent K time steps:

$$\bar{\theta} = \frac{1}{K}\sum_{k=1}^{K}\theta_{t-k}, \tag{6}$$

where K is a hyperparameter denoting the teacher size.

Figure 2 shows the training phase of our proposed method. In the training phase, we can compute $\bar{\theta}$ efficiently by moving average.

Since the teacher model aggregates information of student models after every time step, it is usually more robust. Moreover, a better student model further leads to better a teacher model.

Self-Distillation-Voted (SDV) As a comparison, we also propose an alternative self-distillation method by replacing the teacher model with self-voted BERT, called $\mathrm{BERT}_{\mathrm{SDV}}$. The objective of $\mathrm{BERT}_{\mathrm{SDV}}$ strategy is

$$\begin{aligned}\mathcal{L}_\theta(x,y) &= \mathrm{CE}\Big(\mathrm{BERT}(x,\theta), y\Big) \\ &+ \lambda\mathrm{MSE}\Big(\mathrm{BERT}(x,\theta), \frac{1}{K}\sum_{k=1}^{K}\mathrm{BERT}(x,\theta_{t-k})\Big).\end{aligned} \tag{7}$$

Type	Dataset	Num of Labels	Train samples	Dev samples	Test samples
Text Classification	IMDb	2	25,000	-	25,000
	AG's News	4	120,000	-	7,600
	DBPedia	14	560,000	-	70,000
	Yelp Polarity	2	560,000	-	38,000
	Yelp Full	5	650,000	-	50,000
NLI	SNLI	3	549,367	9,842	9,824
	MNLI-(m/mm)	3	392,702	9,815/9,832	9,796/9,847

Table 1: Summary statistics of seven widely-studied text classification and natural language inference (NLI) datasets.

The training efficiency of $\mathrm{BERT}_{\mathrm{SDV}}$ strategy is lower than $\mathrm{BERT}_{\mathrm{SDA}}$ strategy since $\mathrm{BERT}_{\mathrm{SDV}}$ needs to process the input with recent K student models.

4 Experiments

In this paper, we improve BERT fine-tuning via self-ensemble and self-distillation. The vanilla fine-tuning method of BERT is used as our baseline. Then we evaluate our proposed fine-tuning strategies on seven datasets to demonstrate the feasibility of our self-distillation model.

4.1 Datasets

Our proposed method is evaluated on five Text Classification datasets and two Natural Language Inference (NLI) datasets. The statistics of datasets are shown in Table 1.

Text Classification

- **IMDb** IMDb [Maas *et al.*, 2011] is a binary sentiment analysis dataset from the Internet Movie Database. The dataset has 25,000 training examples and 25,000 validation examples. The task is to predict whether the review text is positive or negative.
- **AG's News** AG's corpus [Zhang *et al.*, 2015] of the news articles on the web contains 496,835 categorized news articles. The four largest classes from this corpus with only the title and description fields were chosen to construct the AG's News dataset.
- **DBPedia** DBPedia [Zhang *et al.*, 2015] is a crowdsourced community effort that includes structured information from Wikipedia. The DBPedia dataset is constructed by picking 14 non-overlapping classes from DBPedia 2014 with only the title and abstract of each Wikipedia article.
- **Yelp** The Yelp dataset is obtained from the Yelp Dataset Challenge in 2015, built by Zhang *et al.* [2015]. There are two classification tasks in this dataset: Yelp Full and Yelp Polarity. Yelp Full predicts the full number of stars (1 to 5) which given by users, and the other predicts a polarity label that is positive or negative.

Natural Language Inference

- **SNLI** The Stanford Natural Language Inference Corpus [Bowman *et al.*, 2015] is a collection of 570k human-written English sentence pairs manually labeled for balanced classification with the labels entailment, contradiction, or neutral.
- **MNLI** Multi-Genre Natural Language Inference [Williams *et al.*, 2018] corpus is a crowdsourced entailment classification task with about 433k sentence pairs. The corpus has two test sets: matched (MNLI-m) and mismatched (MNLI-mm).

4.2 Hyperparameters

All the hyperparameters of our proposed methods are the same as the official BERT [Devlin *et al.*, 2018] except self-distillation weight λ and teacher size K. We use AdamW optimizer with the warm-up proportion of 0.1, base learning rate for BERT encoder of 2e-5, base learning rate for softmax layer of 1e-3, dropout probability of 0.1. For sequences of more than 512 tokens, we truncate them and choose head 512 as model input.

We fine-tune all models on one RTX 2080Ti GPU. For $\mathrm{BERT}_{\mathrm{BASE}}$, the batch size is 4, and the gradient accumulation steps is 4. For $\mathrm{BERT}_{\mathrm{LARGE}}$, the batch size is 1, and the gradient accumulation steps is 16.

For ensemble BERT (See section 3.1), we run $\mathrm{BERT}_{\mathrm{BASE}}$ with 4 different random seeds and save the checkpoint.

4.3 Model Selection

As shown in section 3.3, there are two main hyperparameters in our fine-tuning methods ($\mathrm{BERT}_{\mathrm{SDA}}$ and $\mathrm{BERT}_{\mathrm{SDV}}$): self-distillation weight λ and teacher size K.

Self-Distillation Weight We first evaluate our methods on IMDb dataset to investigate the effect of self-distillation weight λ. Figure 3 shows that $\lambda \in [1.0, 1.5]$ has better results. This observation is the same as the other datasets. Therefore, we set $\lambda = 1$ in the following experiments.

Teacher Size We choose different teacher size K and evaluate our models in three datasets. Table 2 shows that teacher size is sensitive to datasets. Therefore, we select the best teacher size for each dataset in the following experiment.

4.4 Model Analysis

Training Stability Generally, distinct random seeds can lead to substantially different results when fine-tuning BERT even with the same hyperparameters. Thus, we conduct experiments to explore the effect of data order on our models.

This experiment is conducted with a set of data order seeds. One data order can be regarded as one sample from the set of permutations of the training data. 1,500 labeled examples from SNLI dataset (500 on each class) are randomly selected to construct a new training set. With the same initialization

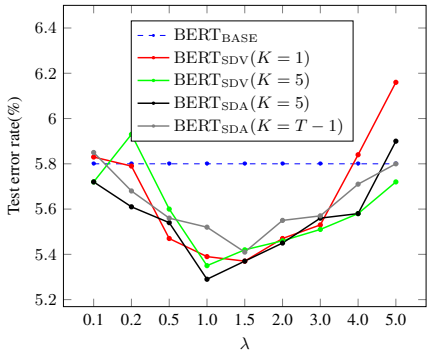


Figure 3: Test error on IMDb dataset over different setting of self-distillation weight λ. T denotes the total number of iterations.

Model	K	IMDb	AG's News	SNLI
BERT$_{\text{SDV}}$	1	5.39	**5.38**	**91.2**
	2	5.44	5.39	91.1
	3	5.40	5.50	**91.2**
	4	5.47	5.49	**91.2**
	5	**5.35**	5.55	91.1
BERT$_{\text{SDA}}$	$T-1$	5.41	**5.29**	91.0
	2	5.46	5.49	**91.2**
	3	5.48	5.55	91.1
	4	5.44	5.52	91.1
	5	**5.29**	5.41	91.1

Table 2: Results on IMDb dataset over different teacher sizes. BERT$_{\text{SDV}}(K=1)$ is same as BERT$_{\text{SDA}}(K=1)$. For IMDb and AG's News, we report test error rate (%). For SNLI, we report accuracy (%). T denotes the total number of iterations.

seeds but different data order seeds, we run 10 times for each fine-tuning strategy and record the result as Figure 4.

Results show that our strategies have higher accuracy and smaller variance than the vanilla BERT$_{\text{BASE}}$ fine-tuning. This proves that the fine-tuned BERT with the self-distillation strategy inherits the property of the ensemble model and is less sensitive to the data order.

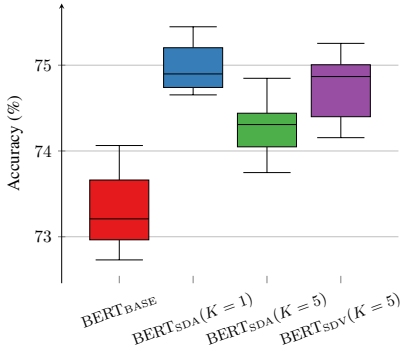


Figure 4: Boxplot for training stability. Only 1,500 labeled example in SNLI training set are used for training. Here BERT$_{\text{SDA}}(K=1)$ is same as BERT$_{\text{SDV}}(K=1)$.

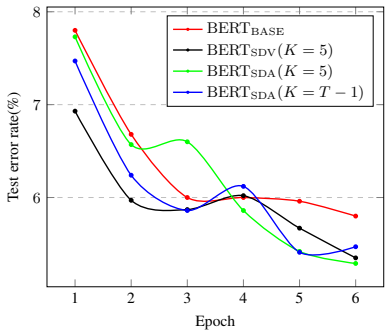


Figure 5: Test error rates(%) on IMDb dataset durning different epoch. T denotes the total number of iterations.

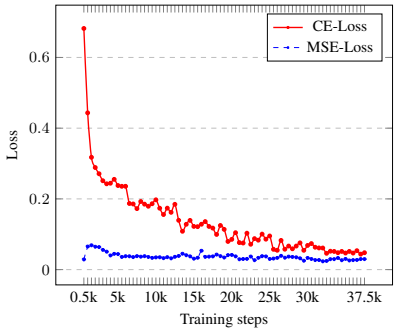


Figure 6: Loss curve of BERT$_{\text{SDA}}(K=1)$ on IMDb dataset.

Convergence Curves To understand the effects of using self-distillation, we record the converge curve while training. The training curves on IMDb dataset are shown in Figure 5. Fine-tuning BERT$_{\text{BASE}}$ cannot get significant improvement in the last 3 epochs (from 6.00% to 5.80%). But with self-distillation mechanisms, the test error rate can further decrease to 5.35% (for BERT$_{\text{SDV}}$) and 5.29% (for BERT$_{\text{SDA}}$).

To further analyze the reason for this observation, we also record the loss curve of cross-entropy (CE) loss and mean-squared-error (MSE) loss, as shown in Figure 6. When training begins, the CE loss dominates the optimization objective. In the last phase of training, the cross-entropy loss becomes small, and a large proportion of gain also comes from self-distillation. Therefore, although optimizing the CE loss at the end of the training phase cannot continue improving the performance of BERT, self-distillation with ensemble BERT as the teacher will continuously enhance the generalization and robustness of BERT.

4.5 Model Performance

In this section, we evaluate our proposed fine-tuning strategies for the BERT-base and BERT-large models on text classification and NLI tasks.

Effects on Fine-tuning BERT-Base Table 3 shows the results of fine-tuning the BERT-base model on five text classification datasets and two NLI datasets. For ensemble BERT, both the voted BERT (BERT$_{\text{VOTE}}$) and averaged BERT (BERT$_{\text{AVG}}$) outperform the single BERT (BERT$_{\text{BASE}}$). The average improvement of BERT$_{\text{VOTE}}$ is 5.44% (for text clas-

Model	IMDb	AG's News	DBPedia	Yelp P.	Yelp F.	Avg. Δ	SNLI	MNLI (m/mm)	Avg. Δ
	Test Error Rate (%)						Accuracy (%)		
ULMFiT [Howard and Ruder, 2018]	4.60	5.01	0.80	2.16	29.98	/	/	/	/
$\mathrm{BERT_{BASE}}$ [Sun *et al.*, 2019]*	5.40	5.25	0.71	2.28	30.06	/	/	/	/
$\mathrm{BERT_{BASE}}$	5.80	5.71	0.71	2.25	30.37	-	90.7	84.6/83.3	-
$\mathrm{BERT_{VOTE}}$ ($K=4$)	5.60	5.41	0.67	2.03	29.44	5.44%	91.2	85.3/84.4	5.50%
$\mathrm{BERT_{AVG}}$ ($K=4$)	5.68	5.53	0.68	2.03	30.03	4.07%	90.8	85.1/84.2	3.24%
$\mathrm{BERT_{SE}}$ (ours)	5.82	5.59	0.65	2.19	30.48	2.50%	90.8	84.2/83.3	-0.51%
$\mathrm{BERT_{SDV}}$ (ours)	5.35	5.38	**0.68**	2.05	**29.88**	5.65%	**91.2**	**85.3/84.3**	**5.30%**
$\mathrm{BERT_{SDA}}$ (ours)	**5.29**	**5.29**	**0.68**	**2.04**	**29.88**	**6.26%**	91.2	85.0/84.3	4.65%

Table 3: Effects on fine-tuning the BERT-base model ($\mathrm{BERT_{BASE}}$). '*' indicates using extra fine-tuning strategies and data preprocessing. '/' means no available reported result. We implemented a "$\mathrm{BERT_{BASE}}$" without any extra fine-tuning strategy as our baseline. "$\mathrm{BERT_{VOTE}}$" and "$\mathrm{BERT_{AVG}}$" means ensemble BERT (See section 3.1). "$\mathrm{BERT_{SE}}$" means self-ensemble BERT (See section 3.2). "$\mathrm{BERT_{SDV}}$" and "$\mathrm{BERT_{SDA}}$" means self-distillation BERT (See section 3.3). 'Avg. Δ' means the average of relative change, respectively. We **bold** the better self-distillation results.

sification) and 5.50% (for NLI), while $\mathrm{BERT_{AVG}}$ follows closely with 4.07% and 3.24%. $\mathrm{BERT_{VOTE}}$ outperforms $\mathrm{BERT_{AVG}}$ on all tasks, which adheres to our intuition since $\mathrm{BERT_{VOTE}}$ is more complicated.

The self-ensemble BERT ($\mathrm{BERT_{SE}}$) has a slight improvement in classification tasks of 2.50%, but it does not work on NLI tasks. This is also a reason why we need self-distillation to improve the base models.

Overall, self-distillation model has significant improvement on both classification and NLI tasks. Table 3 shows that $\mathrm{BERT_{SDA}}$ and $\mathrm{BERT_{SDV}}$ outperform $\mathrm{BERT_{BASE}}$ on all datasets. Generally speaking, $\mathrm{BERT_{SDA}}$ performs better than $\mathrm{BERT_{SDV}}$ on text classification tasks with the improvement of 6.26% vs. 5.65%, but the latter performs better on NLI tasks ($\mathrm{BERT_{SDA}}$ vs. $\mathrm{BERT_{SDV}}$ is 4.65% vs. 5.30%).

Our proposed fine-tuning strategies also outperform the previous method in [Sun *et al.*, 2019] on text classification tasks, which makes extensive efforts to find sophisticated hyperparameters.

Model	IMDb	AG's News	Avg. Δ	SNLI	Δ
MT-DNN [Liu *et al.*, 2019a]	/	/	/	91.6	/
BERT-L (our implementation)	4.98	5.45	-	90.9	-
$\mathrm{BERT\text{-}L_{SDA}}(K=1)$	4.66	5.21	5.62%	**91.5**	**6.59%**
$\mathrm{BERT\text{-}L_{SDA}}(K=T-1)$	**4.58**	**5.15**	**7.02%**	91.4	5.49%

Table 4: Effects on fine-tuning the BERT-large model (BERT-L). For IMDb and AG's News, we report test error rate (%). For SNLI, we report accuracy (%). MT-DNN fine-tunes BERT with multi-task learning.

Effects on Fine-tuning BERT-Large We also investigate whether self-distillation has similar findings for the BERT-large model (BERT-L), which contains 24 Transformer layers. Due to the limitation of our devices, we only conduct an experiment on two text classification datasets and one NLI datasets and evaluate strategy $\mathrm{BERT_{SDA}}$, namely self-distillation with averaged BERT as a teacher. We set two different teacher sizes for comparison. As shown in Table 4, self-distillation also gets a significant gain while fine-tuning the BERT-large model. On two text classification tasks, $\mathrm{BERT\text{-}L_{SDA}}(K=T-1)$ gives better results and the average improvement is 7.02%. For NLI task, $\mathrm{BERT\text{-}L_{SDA}}(K=1)$ gives better result and the improvement is 6.59%.

Moreover, although our self-distillation fine-tuning strategy does not leverage the external data or knowledge, it also gives a comparable performance of MT-DNN [Liu *et al.*, 2019a], which fine-tunes BERT with a specific projection layer under the multi-task learning framework.

Discussion In general, $\mathrm{BERT_{SDA}}$ has a similar phenomenon compared to $\mathrm{BERT_{SDV}}$, while having better computational and memory efficiency. Considering that $\mathrm{BERT\text{-}L_{SDA}}(K=1)$ is same as $\mathrm{BERT\text{-}L_{SDV}}(K=1)$, and it performs better than $\mathrm{BERT\text{-}L_{SDA}}(K=T-1)$. The SDA models are generally worse than SDV models on NLI tasks. All the above illustrates that parameter averaging is worse than logits voting when dealing with difficult tasks such as NLI, but better on simple tasks such as text classification.

5 Conclusion

In this paper, we propose simple but effective fine-tuning strategies for BERT without external knowledge or data. Specifically, we introduce two mechanisms: self-ensemble and self-distillation. The self-ensemble method with parameter averaging can improve BERT without significantly decreasing the training efficiency. With self-distillation, the student and teacher models can benefit from each other. Our proposed strategies are orthogonal to the approaches with external data and knowledge. Therefore, we believe that our strategies can be further boosted by more sophisticated hyperparameters and data augmentation.

In future, we will investigate a better fine-tuning strategy by integrating our proposed method into an optimization algorithm.

References

[Bowman *et al.*, 2015] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2015.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Dong *et al.*, 2019] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*, 2019.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Houlsby *et al.*, 2019] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *arXiv preprint arXiv:1902.00751*, 2019.

[Howard and Ruder, 2018] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[Jiao *et al.*, 2019] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

[Laine and Aila, 2016] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[Liu *et al.*, 2019a] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.

[Liu *et al.*, 2019b] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[Maas *et al.*, 2011] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

[Peters *et al.*, 2019] Matthew Peters, Sebastian Ruder, and Noah A Smith. To tune or not to tune? adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*, 2019.

[Polyak and Juditsky, 1992] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

[Raffel *et al.*, 2019] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

[Rajpurkar *et al.*, 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[Sanh *et al.*, 2019] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[Stickland and Murray, 2019] Asa Cooper Stickland and Iain Murray. BERT and PALs: Projected attention layers for efficient adaptation in multi-task learning. *arXiv preprint arXiv:1902.02671*, 2019.

[Sun *et al.*, 2019] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune BERT for text classification? *arXiv preprint arXiv:1905.05583*, 2019.

[Tarvainen and Valpola, 2017] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[Williams *et al.*, 2018] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1112–1122, 2018.

[Yang *et al.*, 2019] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*, pages 5754–5764, 2019.

[Zhang *et al.*, 2015] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.