Scale-invariant-Fine-Tuning (SiFT) for Improved Generalization

Luheng (Wes) Wang
New York University
lw2534@nyu.edu

Yilun Kuang
New York University
yk2516@nyu.edu

Yash Bharti
New York University
yb1025@nyu.edu

Abstract

In natural language processing, we are concerned about domain generalization of our model when it comes to real life scenarios. In addition, researchers find that adversarial training, while improves robustness, often restricts model's ability for generalization. The current state-of-the-art adversarial training algorithm SMoothness-inducing Adversarial Regularization and BRegman pRoximal poinT opTimization (SMART) algorithm (Jiang et al., 2020) resolves this conflict. We suggest a variation of SMART, the Scale-invariant-Fine-Tuning (SiFT) which is inspired by the brief description in the DeBERTa (He et al., 2021) paper.

1 Introduction

Domain generalization is a capability of a model that allows it to be trained in one domain and perform well in another unseen domain. The concrete definition of domain varies case by case. For example, it is intuitive to believe that a generalized model which performs well on twitter sentiment classification should also gain good results on IMDB sentiment datasets.

1.1 SMART

In the SMART paper, the authors first use multi-task learning (MTL, (Liu et al., 2019a)) with SMART to train shared embeddings to regularize and prevent over-fitting. They have the regular fine-tuned model on each task as the baseline. Then, they test the MTL model with SMART on SNLI and SciTail, which are considered out-domain tasks. The results show consistent improvement over the baselines.

More specifically, SMART consists of two sections. First, it introduces smoothness-inducing adversarial training. It creates a robust error $R_s(\theta)$ which is defined to be the symmetrized KL-Divergence of $1)$ output of the model with parameters θ and input x, and $2)$ output of the model with the same set of parameters θ but different input $\tilde{x}$, which is the perturbed embeddings. SMART combines this robust error and the regular model error with a hyperparameter coefficient α. Then it tries to optimize this regularized loss so that the output of the model does not change much when a small perturbation is injected, thus smoothness.

Second, SMART incorporates Bregman Proximal Point Optimization to optimize the regularized loss in order to avoid aggressive update. It is done by using KL-Divergence again to monitor the change in the outputs when the parameter θ is updated in each iteration.

1.2 Motivation and Baseline Results Summary

In the DeBERTa paper, the author briefly explains how SiFT differs from SMART, and claims that a more comprehensive study is left to future work. The adversarial training procedure is the same in SiFT and SMART, while embeddings are perturbed after they are normalized in SiFT, but not before they are normalized as in SMART. We believe this is a sophisticated project idea, and we have the code base for SMART on Github. Thus, we decide to do a research on whether and by how much SiFT improves generalization. Note that in the Github repository for SMART, it also contains code base for Multi-Task Deep Neural Network (MT-DNN, (Liu et al., 2019a)). Therefore, instead of cloning the entire package, we select only relevant files and make modifications to them. Our baselines consist of multiple models and results (more to be discussed in section 4). One of the essential findings is that when BERT and DeBERTa are trained on one domain, it performs poorly in another domain. For instance, when they are fine-tuned in

domain of twitter, it performs almost as poorly as random guess in domain of IMDB.

2 Background

In recent years, pretrained language models fine-tuned for downstream tasks like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), and DeBERTa (He et al., 2021) have achieved SOTA performance in various benchmarks (Vaswani et al., 2017). However, fine-tuning algorithms sometimes overfit the data and result in poor generalization performance (Jiang et al., 2020).

2.1 Random Perturbation Regularization

Classical regularization techniques augment the optimization objective with a penalization term with a hyperparameter (Miyato et al., 2018). From a Bayesian perspective, the addition of the regularization terms serve as a prior belief of the conditional output distribution $p(y|x)$ of the model (Bishop 2006; Miyato et al. 2018). For the model to have good generalization performance, the conditional output distribution $p(y|x)$ of the model should be smooth relative to the conditional input x (Miyato et al., 2018). In other words, the model should produce roughly the same output distribution $p(y|x)$ if the input data x is isotropically perturbed (Miyato et al., 2018).

Exploiting the idea of random perturbation uniformly in all directions, it has been demonstrated that training neural network models with random noise perturbation to the input x is equivalent to Tikhonov regularization (Bishop, 1995).

Subsequent studies by (Szegedy et al., 2014) and (Goodfellow et al., 2015) show that random perturbation in the adversarial direction leads to dramatic changes in the output distribution. Algorithms like adversarial training in supervised settings are proposed to add perturbations to the most anisotropic direction in the optimization objective (Miyato et al., 2018).

2.2 Adversarial Training

It is suggested that adversarial examples consistent leads to misclassification in a varieties of machien learning and neural network models (Goodfellow et al., 2015). The adversarial training is then proposed with the optimization formulation as follows

$$\mathcal{L}_{adv}(x_l, \theta) := l_s[q(y|x_l), p(y|x_l + r_{adv}, \theta)] \quad (1)$$

$$r_{adv} := argmax_{r;||r||\leq\epsilon}\, l_s[q(y|x_l), p(y|x_l + r, \theta)] \quad (2)$$

where $l_s(q, p)$ is a function that measures the difference between two distribution $q(\cdot)$ and $p(\cdot)$, $q(y|x_l)$ is the true output distribution in the supervised setting, $p(y|x_l)$ is the output distribution of the model, and $\mathcal{D}_l = \{(x_l^{(n)}, y_l^{(n)})\}_{n=1}^{N_l}$ is a labeled dataset (Miyato et al., 2018). Notice that under L_∞ norm, $r_{adv} \approx \epsilon sign(\nabla_{x_l}\mathcal{D}[q(y|x_l), p(y|x_l, \theta)])$, which is the original fast gradient sign method regularization technique proposed by (Goodfellow et al., 2015) for adversarial trianing.

By minimizing the loss $\mathcal{L}_{adv}$ with respect to the model output distribution $p(y|x_l + r_{adv}, \theta)$ that is adversarially perturbated, the output distributions of the model are forced to be smooth along the anisotropic direction. Adversarial training thereby regularizes the model to be robust against adversarial perturbations (Miyato et al., 2018).

2.3 Virtual Adversarial Training

To extend adversarial training to semi-supervised learning domain, consider

$$\mathcal{L}_{vadv}(x_l, \theta) := l_s[q(y|x_*), p(y|x_* + r_{qadv}, \theta)] \quad (3)$$

$$r_{qadv} := argmax_{r;||r||\leq\epsilon}\, l_s[q(y|x_*), p(y|x_*+r, \theta)] \quad (4)$$

where x_* comes from either the labeled dataset $\mathcal{D}_l = \{(x_l^{(n)}, y_l^{(n)})\}_{n=1}^{N_l}$ and the unlabeled dataset $\mathcal{D}_{ul} = \{x_{ul}^{(m)}\}_{m=1}^{N_{ul}}$ (Miyato et al., 2018). Given that $q(y|x_{ul})$ is not accessible in the semi-supervised setting, we can approximate $q(y|x_{ul})$ by the current estimate $p(y|x_*, \hat{\theta})$ which is a "virtual" label (Miyato et al., 2018). Then the averaged $\mathcal{L}_{vadv}$ can be augmented to the full loss as a regularization term. By extending adversarial training to the semi-supervised domain, we can then regularize the model against adversarial attacks.

2.4 SMART, ALUM, SiFT

Several virtual adversarial training algorithm like SMART (Jiang et al., 2020), ALUM (Liu et al., 2020), and SiFT (He et al., 2021) are proposed. SMART uses a similar adversarial regularization term with modification of symmetric KL-divergence as adversarial loss compared to the standard VAT adversarial loss. Bregman proximal point optimization is used to constrain aggressive parameter updates (Jiang et al., 2020).

```
(deberta): DebertaModel(
  (embeddings): DebertaEmbeddings(
    (word_embeddings): Embedding(50265, 768, padding_idx=0)
    (LayerNorm): DebertaLayerNorm()
    (dropout): StableDropout()
```

Figure 1: DeBERTa Embedding Layer.

ALUM builds on the SMART VAT training objective by replacing of the bregman proximal point optimization with a curriculum learning approach, which train the model using standard object first and then switch to virtual adversarial training (Liu et al., 2020). The curriculum learning approach leads to faster training time and spare the use of the Bregman proximal point method (Liu et al., 2020).

The Scale Invariant Fine-Tuning (SiFT) algorithm builds on SMART by applying perturbations to normalized embeddings (He et al., 2021). It is claimed that the extra normalization applied to the word embedding improve the generalization performance of the DeBERTa and DeBERTa$_{1.5B}$ model (He et al., 2021). In this paper, we explore the SiFT algorithm, which is one variant of the VAT algorithm proposed by (Miyato et al., 2018). By comparing SiFT with the standard fine-tuning baseline and SMART, we can see how SiFT differs from SMART in generalization improvement. Since ALUM shares the similar loss objective with SMART with improved training technique, it would also be beneficial to see if SiFT makes substantial improvement against ALUM's leaderboard result.

3 Experiment Setup

Since SiFT Algorithm is proposed in the DeBERTa paper, we follow their work and first try to implement SiFT on DeBERTa. Second, we also incorporate BERT as our alternative model.

We use the following datasets: 1) Twitter Hate Speech dataset[1], 2) UCI Sentiment dataset [2], 3) CoLA dataset (Warstadt et al., 2019). The usage will be detailed in section 4.

4 Experiment Design and Baselines

As mentioned, SMART resolves the conflict between generalization and robustness researchers have been detecting in past years. Thus, we want to implement SiFT based on SMART. In Figure 1, DeBERTa embedding layer is shown. SMART, along with other adversarial training in the past adds perturbation to the word embeddings at line 3 in Figure 1. However, SiFT adds perturbation to the output of line 4, which are the normalized embeddings. For BERT it is the same story. We achieve this by modifying the SmartPerturbation class from SMART Github repository, and adding an instance[3] of it to our training loop. We have one bug as of now, but the overall progress is satisfying and close to success. Note that SMART only supports ranking and classification tasks. We choose to focus on classification ones.

	0%	10%
BERT	0.55	0.61
DeBERTa	0.65	0.56

Table 1: Horizontal axis represents how much data from UCI is used for further fine-tuning; Vertical axis represents the type of model.

Our baseline consists of the followings. First, we fine-tune BERT and DeBERTa on COLA and with resulting Matthews Correlation Coefficient of 0.514 and 0.54 respectively. We consider this as a baseline because we will compare these baseline results with the results of the same two models with SiFT implemented. Second, we define another baseline specific to show domain generalization. We fine-tune BERT and DeBERTa on Twitter Hate Speech dataset. Then, we test directly on UCI sentiment dataset without any further training. This baseline effectively shows how generalized our model is with regard to unseen domain. In addition, we also include quantative analalysis - after fine-tuning the two models on Twitter Hate Speech dataset, we do another fine-tuning on UCI sentiment dataset, but with only 10% of the data. This is a setup we learned from SMART paper. This shows quantitatively the adaption of the models on out-of-domain tasks.

[1] https://www.kaggle.com/arkhoshghalb-sentiment-analysis-hatred-speech

[2] http://archive.ics.uci.edu/ml/datasets.php

[3] shown here: https://github.com/YashBit/MLLU-Class-Project/blob/main/SMART$_{i}mplementation_{o}n_{B}ERT.ipynb$

5 Collaboration Statement

5.1 Luheng (Wes) Wang

1. Provide a basic implementation of BERT and DeBERTa on CoLA
2. Implement SMART on BERT for CoLA (close to success)
3. Write Abstract and Section 1,3,4 (including citations) in the Partial Draft
4. Help Yash fix bugs in his experiment code

5.2 Yilun (Mark) Kuang

1. Propose a basic implementation of random noise perturbation on DeBERTa and modify experimental design
2. Help Wes implement SMART on BERT for CoLA
3. Write Section 2 (including citations) in the Partial Draft

5.3 Yash Bharti

1. Run experiments of BERT and DeBERTa fine-tuned on Twitter HateSpeech dataset, and test on UCI (with 0% and 10% further training on UCI).

6 Github

All code files for the current project can be found here:
https://github.com/YashBit/MLLU-Class-Project

References

Chris M. Bishop. 1995. Training with noise is equivalent to tikhonov regularization. *Neural Computation*, 7(1):108–116.

Christopher M Bishop. 2006. *Pattern recognition and machine learning*. springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach.

Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: A regularization method for supervised and semi-supervised learning.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments.