Adversarial Training on Chinese NLP Tasks

Peiji Li
Fudan NLP
ID: 20307140044
20307140044@fudan.edu.cn

Abstract

Adversarial Training is a "defense" strategy in deep learning. During the training, the model is attached to improve the robustness of the model, so as to try to achieve better results in the test process. Adversarial training was initially applied to CV tasks, but experiments have proved that it can also achieve better results in NLP tasks. In NLP field, the commonly used adversarial training method is to disturb the word embedding layer. Although this method is not so "explainable" as in CV field, as the disturbed word vector does not seem to be able to find the corresponding word map in the dictionary, previous experiments have already proved that it can indeed improve the accuracy of the model in the test set. This paper attempts to use several mainstream adversarial training methods like FGM, PGD and FreeLB, and apply them to the training process of Chinese NLP tasks using pretraining model.

1 Introduction

In this project, I applied three adversarial training methods(FGM,PGD and FreeLB) in two Chinese NLP tasks: SSC(Sentence Sentiment Classification) and NER(Name Entity Recognition). I used pretrained language model(bert-base-chinese) and fine-tune them with three adversarial training methods in my tasks. The adversarial attack in training process is used on the word embedding layer in bert module. The model architecture used is bert + linear and bert + CRF, which is generally used in many paper.

The experiment shows that using adversarial training methods could usually provide better result in the two Chinese NLP tasks compared with common training after training for same epochs until the model converged. In Chinese SSC, FGM and PGD methods performs better than common training methods,the accuracy in test set has been promoted apparently, PGD reached 95.4% accuracy in test set. And in Chinese NER, all the three methods perform better in test dataset. PGD reached 99.25% on total accuracy and FreeLB reached 97.84% on content accuracy on test dataset.

2 Dataset

ChnSentiCorp: A dataset for Chinese sentence-level sentiment classification (SSC), with two labels:0 and 1, representing negative or positive sentiment.

Split: 'train':9600, 'validation':1200, 'test':1200

peoples_daily_ner: People's Daily NER Dataset is a commonly used dataset for Chinese NER, with text from People's Daily, the largest official newspaper. The dataset is in BIO scheme. Entity types are: PER (person), ORG (organization)and LOC (location). So there are 7 labels in total.

Split: 'train':20865,'test':4637,'validation':2319.

Nevertheless, restricted by GPU memory, in my tasks, I filtered the NER dataset with word length of sentence less than 48.

Preprint. Under review.

3 Methodology

3.1 Model Architecture

In my project, I use pretrained model *bert-base-chinese* from *hugginface* as my base module, then I use linear layer or CRF layer connected to bert. The Bert module is not freezed, they will be trained together in the downstream tasks, also with adversarial training methods which make disturbance to the word embedding layer.

CRF: Conditional random fields are a class of statistical modeling methods often applied in pattern recognition and machine learning and used for structured prediction. Whereas a classifier predicts a label for a single sample without considering "neighbouring" samples, a CRF can take context into account. To do so, the predictions are modelled as a graphical model, which represents the presence of dependencies between the predictions. What kind of graph is used depends on the application. For example, in natural language processing, "linear chain" CRFs are popular, for which each prediction is dependent only on its immediate neighbours. In image processing, the graph typically connects locations to nearby and/or similar locations to enforce that they receive similar predictions.This method is depicted in Figure 1.

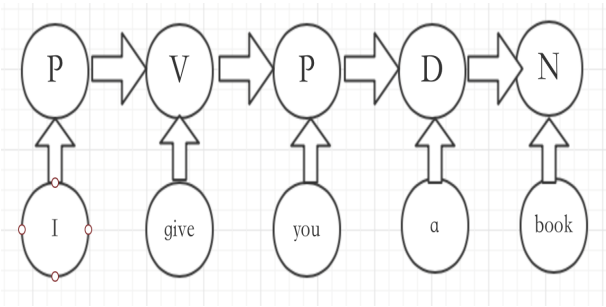

Figure 1: Conditional random fields

Bert: Bidirectional Encoder Representations from Transformers is a transformer-based machine learning technique for natural language processing pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google. In 2019, Google announced that it had begun leveraging BERT in its search engine, and by late 2020 it was using BERT in almost every English-language query. A 2020 literature survey concluded that "in a little over a year, BERT has become a ubiquitous baseline in NLP experiments", counting over 150 research publications analyzing and improving the model.This model is depicted in Figure 2

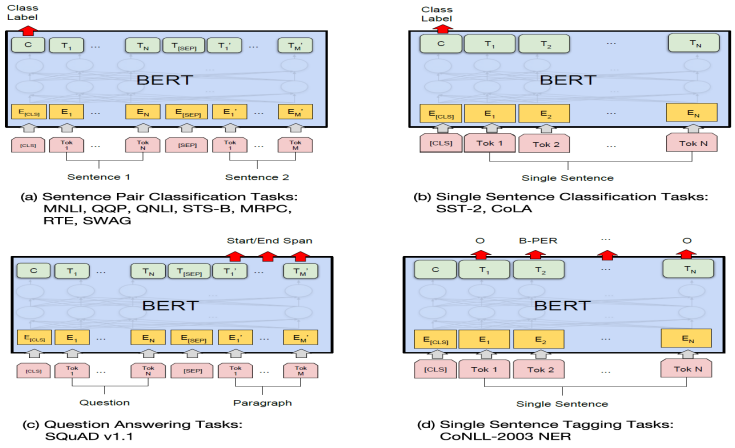

Figure 2: Bert

3.2 Adversarial Training Methods

I use three mainstream adversarial training methods in my training process.They are FGM,PGD and FreeLB. The key idea of adversarial training is the **Min-Max Formula**.It could be described as follows.

$$min_\theta E_{(x,y)} \sim D[max_{r_{adv} \in S} L(\theta, x + r_{adv}, y)] \tag{1}$$

3.2.1 FGM

Fast Gradient Method is proposed by *Goodfellow* in ICLR 2017,the key idea is as followed.

$$r_{adv} = \epsilon g / ||g||_2 \tag{2}$$

$$g = \Delta_x L(\theta, x, y) \tag{3}$$

3.2.2 PGD

Projected Gradient Descent is proposed in ICLR2018, it is more "accurate" than FGM. The key idea is as followed.

$$x_{t+1} = \prod_{x+S} (x_t + \alpha g(x_t)/||g(x_t)||_2) \tag{4}$$

$$g(x_t) = \Delta_x L(\theta, x_t, y) \tag{5}$$

3.2.3 FreeLB

"Free" Large-Batch Adversarial Training is proposed in paper **Enhanced Adversarial Training for Language Understanding** in 2019. It is similar but different from PGD. The key ideas of FreeLB Algorithm is depicted in Figure 3.

Algorithm 1 "Free" Large-Batch Adversarial Training (FreeLB-K)

Require: Training samples $X = \{(\boldsymbol{Z}, y)\}$, perturbation bound ϵ, learning rate τ, ascent steps K, ascent step size α

Initialize $\boldsymbol{\theta}$
for epoch $= 1 \ldots N_{ep}$ **do**
  for minibatch $B \subset X$ **do**
    $\boldsymbol{\delta}_0 \leftarrow \frac{1}{\sqrt{N_\delta}} U(-\epsilon, \epsilon)$
    $\boldsymbol{g}_0 \leftarrow 0$
    for $t = 1 \ldots K$ **do**
      Accumulate gradient of parameters θ
        $\boldsymbol{g}_t \leftarrow \boldsymbol{g}_{t-1} + \frac{1}{K} \mathbb{E}_{(\boldsymbol{Z},y) \in B} [\nabla_{\boldsymbol{\theta}} L(f_{\boldsymbol{\theta}}(\boldsymbol{X} + \boldsymbol{\delta}_{t-1}), y)]$
      Update the perturbation δ via gradient ascend
        $\boldsymbol{g}_{adv} \leftarrow \nabla_{\boldsymbol{\delta}} L(f_{\boldsymbol{\theta}}(\boldsymbol{X} + \boldsymbol{\delta}_{t-1}), y)$
        $\boldsymbol{\delta}_t \leftarrow \Pi_{\|\boldsymbol{\delta}\|_F \leq \epsilon} (\boldsymbol{\delta}_{t-1} + \alpha \cdot \boldsymbol{g}_{adv} / \|\boldsymbol{g}_{adv}\|_F)$
    end for
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \tau \boldsymbol{g}_K$
  end for
end for

Figure 3: FreeLB Algorithm

3.3 Experiment

In my project, I conducted my two NLP tasks separately, both using all the three adversarial training methods, in comparison with common training process without adversarial training. All the experiments are conducted several times with a same set of random seeds. The result showed in **Section 4** are the average overcome on the dataset.

I use some machine-learning framework in my code, such as pytorch, huggingface and so on. Also, I use Nvidia GeForce 4090 with cuda memory of 24 GB.

In SSC tasks, the batch size is 64, init_learning rate is 2e-5, with max sentence length trunctaed by 128, training epoch set as 3.

In NER tasks, the batch size is 32, the other hyperparamters is as same as SSC.

There are also ther hyperparameters of adversarial training methods.

In FGM, the ϵ is set to 1 as default value.

In PGD, $(k, \epsilon, \alpha) = (3, 1, 0.3)$

In FreeLB, $(k, \epsilon, \alpha) = (3, 0.2, 0.05)$

4 Results

The result is showed in table 1 and table 2.

In table 1, the row "paper" is refered to the outcome provided by paper *Revisiting Pre-trained Models for Chinese Natural Lauguage Processing*, which could be served as a baseline for my experiment.

Table 1: Result for SSC

Method	Dev Set	Test Set
paper	94.3	94.7
base	94.7	94.6
FGM	95.1	95.3
PGD	**95.3**	**95.4**
FreeLB	95.0	94.8

In the table 2, there are two columns of scores, $score_1$ represent the total accuracy of all the words in the datasets, which means, label "O" is also counted in $score_1$. And $score_2$ only considers the accuracy of the word which is a real **Name Entity**, it could also be referred as "content-accuracy".

Table 2: Result for NER using Bert+CRF

Method	$score_1$	$score_2$
base	99.05	**97.84**
FGM	99.22	97.63
PGD	**99.25**	97.60
FreeLB	99.23	**97.84**

5 Conclusion

From the results showed above,we could deduce that using adversarial training methods could provide better result in Chinese NLP tasks most times. In Chinese SSC tasks, FGM and PGD methods performed better than common training methods,the accuracy in test set has been promoted apparently, especially for PGD which reached 95.4% compared with the baseline accuracy of 94.6%, which is a considerable improment. And in Chinese NER tasks, all the three methods perform better on $score_1$. PGD reached 99.25% on $score_1$, which is the highest accuracy.And FreeLB reached 97.84% on $score_2$ as well as 99.23% on $score_1$, which also performs satisfactory.

This experiment shows that using adversarial training methods may be a good choice for us when we are training a NLP model, especially for the models based on Chinese corpus.

At last, please allow me to express my genuine appreciation to Dr Hang Yan, Professor Xipeng Qiu and Master Peng Li of Fudan NLP Lab. They helped me greatly during my work.

References

[1] EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES Ian J. Goodfellow, Jonathon Shlens

[2] Towards Deep Learning Models Resistant to Adversarial Attacks

[3] Revisiting Pre-trained Models for Chinese Natural Language Processing Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, Guoping Hu

[4] FREELB: ENHANCED ADVERSARIAL TRAINING FOR NATURAL LANGUAGE UNDERSTANDING