Achieving 90% In Data-Centric Industry Deep Learning Task

This paper was downloaded from TechRxiv (https://www.techrxiv.org).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

06-12-2021 / 17-12-2021

CITATION

Guo, Tong (2021): Achieving 90% In Data-Centric Industry Deep Learning Task. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.17128475.v1

DOI

10.36227/techrxiv.17128475.v1
Achieving 90% In Data-Centric Industry Deep Learning Task

Tong Guo
779222056@qq.com

Abstract. In industry deep learning application, our manually labeled data has a certain number of noisy data. To solve this problem and achieve more than 90 score in dev dataset, we present a simple method to find the noisy data and re-label the noisy data by human, given the model predictions as references in human labeling. In this paper, we illustrate our idea for a broad set of deep learning tasks, includes classification, sequence tagging, object detection, sequence generation, click-through rate prediction. The experimental results and human evaluation results verify our idea.

1 Introduction

In recent years, deep learning [1] model have shown significant improvement on natural language processing (NLP), computer vision and speech processing technologies. However, the model performance is limited by the human labeled data quality. The main reason is that the human labeled data has a certain number of noisy data. Previous work [2] has propose the simple idea to find the noisy data and correct the noisy data. In this paper, we first review the way we achieve more than 90 score in classification task, then we further illustrate our idea for sequence tagging, object detection, sequence generation, click-through rate (CTR) prediction.

2 Background

In previous work [2], we illustrate our idea in these steps:
1. It is a text classification task. We have a human labeled dataset-v1.
2. We train a BERT-based [3] deep model upon dataset-v1 and get model-v1.
3. Using model-v1 to predict the classification label for dataset-v1.
4. If the predicted labels of dataset-v1 do not equal to the human labels of dataset-v1, we think they are the noisy data.
5. We label the noisy data again by human, while given the labels of model and human as reference. Then we get dataset-v2.
6. We loop this re-labeling noisy data steps and get the final dataset. Then we get model-v2. We can further loop this steps to get model-v3.
3 Same Idea and More Applications

3.1 sequence tagging

We take named entity recognition (NER) as example for the sequence tagging like tasks. In NER task, we extract several classes of key phrase from a sentence. Follow our idea, we view each class of NER task as a classification task. Then our steps are:

1. It is a NER task. We have a human labeled dataset-v1.
2. We train a BERT-based [3] deep model upon dataset-v1 and get model-v1.
3. Using model-v1 to predict the sequence labels of one class for dataset-v1.
4. If the predicted labels of dataset-v1 do not equal to the human labels of dataset-v1, we think they are the noisy data.
5. We label the noisy data again by human, while given the labels of model and human as reference. Then we get dataset-v2.
6. We loop this re-labeling noisy data steps for all the classes of NER and get the final dataset. Then we get model-v2.

3.2 object detection

Object detection is a computer vision technique that allows us to identify and locate objects in an image or video. Follow our idea, we view each kind of bounding box as a classification task. Then our steps are:

1. It is a object detection task. We have a human labeled dataset-v1.
2. We train a Swin Transformer[4] upon dataset-v1 and get model-v1.
3. Using model-v1 to predict the bounding boxes of one class for dataset-v1.
4. If the predicted bounding box of dataset-v1 is far from the human labeled bounding box of dataset-v1, we think they are the noisy data.
5. We label the noisy data again by human, while given the bounding boxes of model and human as reference. Then we get dataset-v2.
6. We loop this re-labeling noisy data steps for all the classes of object detection and get the final dataset. Then we get model-v2.

3.3 sequence generation

The key step of our idea is about how to judge the noisy data. For sequence generation, we can use BLEU score or other sequence similarity evaluation method. Then our steps are:

1. We take text generation task as example. We have a human labeled dataset-v1.
2. We train a Encoder-Decoder Transformer[5] upon dataset-v1 and get model-v1.
3. Using model-v1 to predict the generated sentences for dataset-v1.
4. If the BLEU score of generated sentences of dataset-v1 is far from the human labeled generated sentences of dataset-v1, we think they are the noisy data.
5. We label the noisy data again by human, while given the generated sentences of model and human as reference. Then we get dataset-v2.

6. We loop this re-labeling noisy data steps and get the final dataset. Then we get model-v2.

3.4 click-through rate prediction

For CTR task, we use the method of [6] that automatically set the label again for the noisy data. CTR task is a click-or-not prediction task, we choose a threshold between the predicted score and the 0/1 online label score to judge whether the data is the noisy data. In this way, we could improve the AUC in dev dataset but the online performance should test online.

4 Experimental Results

We do the experiments of text classification and NER to verify our idea. The results is shown in Table 1 and Table 2. We also do a lot of other classification task and NER task of other dataset. The improvement is also significant and we do not list the detail results.

5 Conclusion

We argue that the key point to improve the industry deep learning application performance is to correct the noisy data. We propose a simple method to achieve our idea and show the experimental results to verify our idea.

Table 1. The experiment result of text classification. The model-v1 and model-v2 are corresponding to the previous section

| model     | Dev Dataset Accuracy | Human Evaluation Accuracy |
|-----------|----------------------|--------------------------|
| model-v1  | 83.3%                | 88.0%                    |
| model-v2  | 91.7%                | 97.2%                    |
| model-v3  | 93.8%                | 97.5%                    |

Table 2. The experiment result of NER. The model-v1 and model-v2 are corresponding to the previous section

| model     | Dev Dataset F1 | Human Evaluation Precision |
|-----------|----------------|----------------------------|
| model-v1  | 73.7%          | 86.0%                      |
| model-v2  | 88.7%          | 97.0%                      |
References

1. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25: 1097-1105.
2. Guo T. Learning From How Human Correct[J]. arXiv preprint arXiv:2102.00225, 2021.
3. Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
4. Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[J]. arXiv preprint arXiv:2103.14030, 2021.
5. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.
6. Guo T. Self-Refine Learning For Data-Centric Text Classification[J]. 2021.