Applying the Boyer-Moore Voting algorithm to the adversarial example on Hadoop platform

Group 6 刘正辉 1909853J-II20-0037 312776047@qq.com

吕宇 1909853G-II20-0027 1134206436@qq.com

Abstract

We are researching the use of Rover technology in the Hadoop platform and found that the Boyer-Moore Voting algorithm used by it can be well applied to the adversarial example. After using the Boyer-Moore Voting algorithm, we can perform better classification results in the final stage, and we The distributed idea was used to make some small changes to the Boyer-Moore Voting algorithm, so that it records each count and Candidate result, so that it can be run on the Hadoop platform to obtain better experimental results, and according to the inspiration With the idea of merging and distributed, our algorithm can achieve a time complexity less than the original O (N).

Key words: Boyer-Moore Voting algorithm, Hadoop, adversarial example

1 Introduction

In the big data analysis, the distributed architecture deep learning is utilized to accelerate the data convergence process. With distributed in the field of big data development, deep learning requires a lot of training and a large amount of memory. Dispersion can alleviate the demand for hardware, and can also make data scalable. A large amount of data analysis will be better to use.

With the gradual expansion of a large number of networks, more data can be obtained than ever. However, a large amount of useless data could not use to get improvement. Interesting data brings huge value to companies, such as Operation (flow analysis), intelligent decision-making (precise advertising, recommended products). Improving the accuracy of data analysis, deep learning has become a very hot direction in recent years. Since 2012, the deep learning convolutional neural network has grown

tremendously. Through deep learning, the value of data has been gradually explored. Excessive data volume leads to long calculation time and large memory consumption, however, distributed structure shown up. We use GPU to achieve the corresponding amount of data. Data parallelism makes data calculation faster, because sufficient memory makes it possible to train on multiple machines. The new work called big data analysis based on deep learning in distributed structure.

The core of distributed is to release a single machine, using multi-machine parallel computing to speed up and free up memory. The research of this topic is of great value. If you can learn more data in the distributed deep learning of big data, get more interesting data, and provide more appropriate services for the corresponding people.

At the same time, we know that the current demand for distributed data in big data is increasing, which is reflected in the following aspects. Here is an example of industry representative Ali.

Apache Hadoop is a popular software in distributed system. It can deal with the problems with massive data and computation. Thus, we choose it in our research. As a mature software, it has been 13 years to improve itself. There are a lot of papers using this software to run different deep learning algorithm, and get a huge achievement. Although it is less and less researchers to use it, we still find out its value. Nowadays, many researchers choose Tensorflow and Pytorch as distributed platform. We wonder the performance between Hadoop, Tensorflow.

The relationship between HDFS and hardware,software as follows:The upper layers of hardware os are HDFS and mapReduce. HDFS and mapreduce belong to hadoop. On top of them, is Hadoop Projects,including habase, pig, etc. and Let's introduce what's the Mapreduce.[6]

2 Related works

Szegedy et al. [8] is the first group that mentioned the concept of adversarial examples, and proposed adversarial perturbation generation as an optimization problem. Goodfellow et. al. proposed Gradient Symbolic Method "(FGSM) to improve calculation

efficiency. It uses adversarial samples to generate a training set for adversarial training. Kurakin et al. [14] proposed a "basic iterative approach" that uses FGSM to generate disturbances iteratively. Some researchers call it 'I-FGSM'. Papernot et al. [7] constructed an adversarial display map to indicate ideal locations that could be effectively affected. Moosavi et al. proposed DeepFool [5] of Further improved the effectiveness of adversarial disturbances. Moosavi-Dezfooli et al. found that image classifiers have image-independent adversarial perturbations. There is paper is same as [5], Metzen et al. [11] proposed UAP for semantic segmentation tasks. They tried to get more robustness against the iterative FGSM attack. Changing the labels predicted by each pixel. Mopuri et al. found out a data-independent universal perturbation. They proposed a new algorithm for dataless targets to generate a universal anti-disturbance, called FFF [7]. Their later work GDUAP [13] improved the effectiveness of the attack and proved the effectiveness of their method on cross-computer vision tasks.

Recently, much attention has been paid to attacks on target detectors. Lu et al. [9] attempted to generate counter-disturbances on the "stop" signs and "face" images to mislead the corresponding detector. Xie et al. [15] proposed a method of iteratively generating adversarial samples to prevent the detector from predicting correctly. Li et al. [16] proposed R-AP attack algorithm proposed focuses on attacking RPN, which is a common component of deep suggestion networks, which generally reduces its performance without knowing the structural details of the detector. Their follow-up work [17] explored the feasibility of adding invisible perturbations to the background to attack object detectors.

3 Methodology

In the course of our research on the resistance of Rover technology in the direction of HDFS, we found that the voting algorithm used by this algorithm can be applied to the Adversarial examples [3], which we studied before, in the final decision-making process. We use the Boyer-Moore Algorithm to divide the categories. The algorithm's steps are as follows:

S1: Initialization element ans = arr [0], counter cnt = 1

S2: Iterate over each number in the array x from the second number:

```
If cnt > 0
   If ans=x
      cnt++;
   else if
        cnt—
else
     cnt = 1;
     ans = arr[i]
```

S3 : return ans;

We further modified the algorithm, and we noticed that our cnt is the number of the most elements in the sequence more than the total number of other elements. So we can use the idea of divide and conquer to divide the sequence into m Each sequence uses the Boyer-Moore Algorithm, then records the cnt and ans, and then uses the Boyer-Moore Algorithm [12] again, until there is only one ans remaining, and it is not difficult to prove that the upper limit of the algorithm is O (NlongN) In fact, many times his time complexity is close to O (N) (for heuristic merging [10]). With the support of distributed computing, we can further accelerate the execution time of the algorithm. We then further used the LR model to classify our results to further improve our accuracy.

The flow chart as follows:

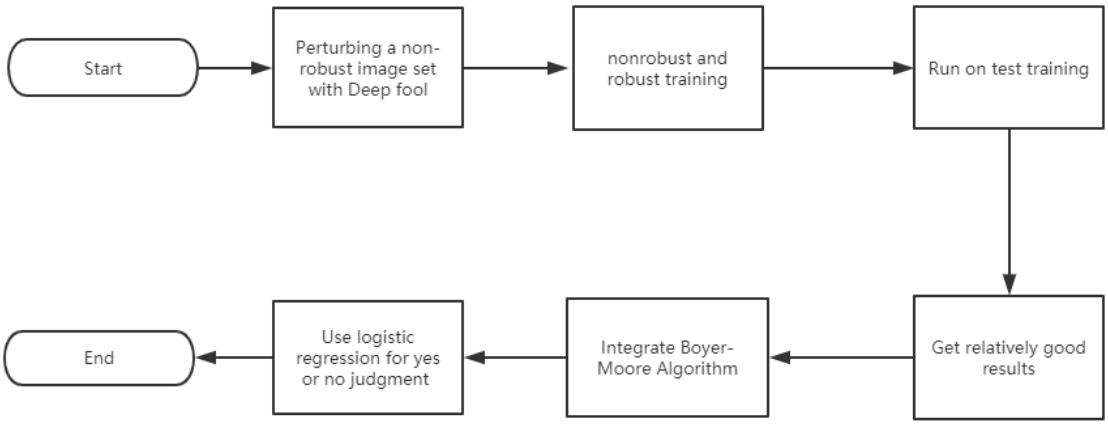

Figure 1 Applying the Boyer-Moore Voting algorithm to the adversarial example on Hadoop platform

4 Reference

[1] Toward Scalable Systems for Big Data Analytics: A Technology Tutorial Digital Object Identifier 10.1109/ACCESS.2014.2332453

[2] Jonathon Shlens Ian J. Goodfellow and Christian Szegedy. Explaining and harnessing adversarial examples. In ICLR, 2015.

[3] Ilyas A, Santurkar S, Tsipras D, et al.Adversarial examples are not bugs, they are features[J]. arXiv preprint arXiv:1905.02175, 2019.

[4] Konda Reddy Mopuri. Utsav Garg. and R. Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. In Proceedings of the British Machine Vision Conference (BMVC).2017.

[5] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In CVPR, 2016.

[6]Borthakur D. HDFS architecture guide [J]. URL http://hadoop. apache. org/docs/stable/hdfs_design. html, 2018.

[7] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In EuroS&P, 2016.

[8] Christian Szegedy. Wojciech Zaremba. Ilya Sutskever. Joan Bruna. Dumitru Erhan. Ian J. Goodfellow. and Rob Fergus. Intriguing properties of neural networks. In ICLR, 2014.

[9] Jiajun Lu. Hussein Sibai. and Evan Fabry. Adversarial examples that fool detectors. CoRR. vol.abs/1712.02494. 2017.

[10] Jafarlou M Z, Fard P Y. Heuristic and pattern based Merge Sort[J]. Procedia Computer Science, 2011, 3: 322-324.

[11] Jan Hendrik Metzen. Mummadi Chaithanya Kumar. Thomas Brox. and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In ICCV, 2017.

[12]Waga M, Akazaki T, Hasuo I. A boyer-moore type algorithm for timed pattern matching[C]//International Conference on Formal Modeling and Analysis of Timed Systems. Springer, Cham, 2016: 121-139.

[13] Konda Reddy Mopuri. Aditya Ganeshan. and R. Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. IEEE Transactions on Pattern Analysis & Machine Intelligence. vol. PP. no. 99. pp. 1–1. 2018.

[14] Alexey Kurakin. Ian J. Goodfellow. and Samy Bengio.Adversarial examples in the physical world. CoRR. vol. abs/1607.02533. 2016.

[15] Cihang Xie. Jianyu Wang. Zhishuai Zhang. Yuyin Zhou. Lingxi Xie. and Alan L. Yuille. Adversarial examples for semantic segmentation and object detection. In ICCV, 2017.

[16] Yuezun Li. Daniel Tian. Ming-Ching Chang. Xiao Bian. and Siwei Lyu. Robust adversarial perturbation on deep proposal-based models. In BMVC, 2018.

[17] Yuezun Li. Xian Bian. and Siwei Lyu. Attacking object detectors via imperceptible patches on background. CoRR. vol. abs/1809.05966. 2018.