악성 댓글 탐지기에 대한 대항 예제 생성

손수현, 이상근
한양대학교 컴퓨터공학과
max2747@hanyang.ac.kr, sangkyun@hanyang.ac.kr

Generating adversarial examples on toxic comment detection

Soohyun Son, Sangkyun Lee
Dept of Computer Engineering, Hanyang University

Abstract

In this paper, we propose a method to generate adversarial examples for toxicity detection neural networks. Our dataset is represented by a one-hot vector and we constrain that only one character is allowed to be modified. The location to be changed is founded by the maximum area of input gradient, which represents the most affecting character the model to make decisions. Despite the fact that we have strong constraint compared to the image-based adversarial attack, we have achieved about 49% successful rate.

1. Introduction

Recently, deep learning and machine learning are showing great performance in the areas including vision, NLP(Natural Language Processing), and speech processing [1].

However, there are many methods for attacking models, such as slowing the progress of learning or disturbing the results of model inference [2]. One of the attacks is generating adversarial examples that make the model result in an incorrect output[3]. Though the adversarial example contains malicious behaviors causing the target model to malfunction, it is not distinguishable in human eyes the difference between adversarial examples and normal ones as shown in Figure 1. Therefore, there have been studies how to defend and robust the model from adversarial examples.

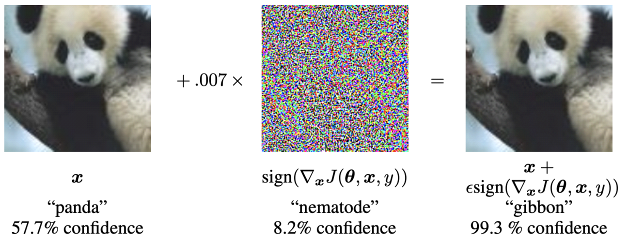

+ .007 ×
=
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure 1. Adversarial examples in image[3]

In this paper, we introduce a method for creating adversarial examples for deep learning model that detecting toxic comments. We assume a white-box attack scenario that an attacker can access the learning parameters of the target model, and set a constraint only one character can be modified based on the input gradient information. This constraint is stronger than the image domain that there is no constraint on where to apply perturbed noise, so it is easy to change the result by applying a small amount of noise through the whole input space[3]. Through the allowed area is limited, we can create adversarial examples roughly 49% of toxic comments in test dataset are classified as normal comments.

2. Method

We generate adversarial examples by modifying one character. It uses the gradient of one-hot input space from loss and modify one character to evade classifier[4].

2.1 Notations

$J(X,y)$ denotes the loss of the model from input X and true label y. Let V refer to the alphabet and special character set which is predefined, and $|V|$ be the size of the set. Input X is comment of length L characters and composed by $x_{i,j} \in 0,1^{|V|}$, meaning j-th character in V of i-th character in comment. The k-th data can be represented by

$$X_k = [(x_{11},\cdots,x_{1n})^T,\cdots,(x_{m1},\cdots,x_{m,n})^T]$$

where max-length of comment is m, and the one-hot vector size of character is n.

As to represent the change one vector in the input space, flip vector $\vec{v}_{i,j,k}$ is defined, which j-th character of the i-th input sequence is replaced by k-th character:

$$\vec{v}_{i,j,k} = (\vec{0},\ldots,(0,\ldots,-1,0,\ldots,1,0)_i^{T,}\ \ldots,\vec{0})$$

where -1 and 1 represent j-th and k-th character in the V respectively.

2.2 Single character attack

In this paper, we assume the white-box adversarial attack, which attackers can access parameters of the target model. In this scenario, the gradient of the input dimension can be calculated from $J(X,y)$.

When character is modified, the change of the loss function can be obtained by calculating the derivative of $\vec{v}_{i,j,k}$:

$$\nabla_{\vec{v}_{i,j,k}} J(X,y) = \nabla_x J(X,y)^T \cdot \vec{v}_{i,j,k}$$

We search k-th character to change the loss most.

$$\operatorname{argmax}_k \nabla_x J(X,y)^T \cdot \vec{v}_{i,j,k} = \operatorname{argmax}_k \frac{\partial J(X,y)}{\partial x_{i,k}} - \frac{\partial J(X,y)}{\partial x_{ij}}$$

However, it is hard to get the global optimal k of the above equation by trying all possible cases. We only need to find k for fooling the target model. To reduce the time for generating adversarial examples, we use the input gradient to find the area to modify. The gradient of input space is interpretable that where the model focus on the part of the input data to make a decision[5]. For example, if $\nabla_{x_k} J(X,y) = 0$ for the variable x_k, then it means that x_k does not affect the result of inference. We modify the character which most influence the model. The method proposed in this paper can be applied to the model trained by one-hot represented dataset. Also, it is easy to control the perturbed area based on the input gradient.

```
input
    - $x_i$ : i-th toxic comment data
    - M : target model
    - $L_t$ : target class label('0'; 'clean' class)

for i=1, ..., M in (toxic comment dataset):
    g <- get_gradient($x_i$, M, $L_t$)
    $g_{\max}$ <- argmax(g, len($x_i$) )
    for vocab in vocab_dictionary:
        adv_data <- modify($x_i$, $g_{\max}$, vocab)
        if $L_t$ = M(adv_data):
            return adv_data
```

Figure 2. The algorithm for generating adversarial examples

3. Experiments

3.1 Dataset

Toxic comment classification challenge dataset[1] is used to train the model. The dataset contains 6 class labels: toxic, severe toxic, obscene, threat, insult, and identity_hate. While in this paper, we transform this dataset to binary classification problem. If the comment is categorized in one of 6 classes, it is classified as 'toxic' class and the remaining data are classified as 'clean' class. After transform it, the dataset is comprised of 2 classes, 'toxic' class has 16K and 'clean' class has 140K instances.

In the dataset, there are various characters including Chinese, Welsh and so on. We experiment on 68 characters containing alphabet and specific special characters.

For preprocessing the comment data, lower the alphabet and remove character except represented regular expression:

[a-z0-9,;.!?:\'\"/\\|_@#$%^&*~`+-=<>()\[\]{}\s]

We split the dataset to a train set, a validation set

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

, and a test set each are about 89K, 22K and 47K.

3.2 Target model

We use character-level CNN [6] to generate adversarial examples. The input of character-level CNN is a one-hot vector and performs convolution operation. The last is fully connected layer and classify binary problem.

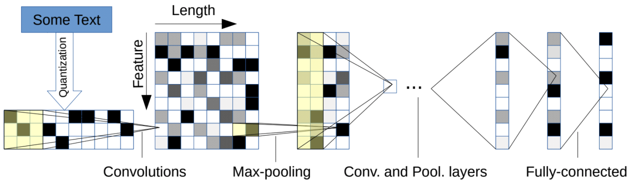


Figure 3. Character-level CNN[6]

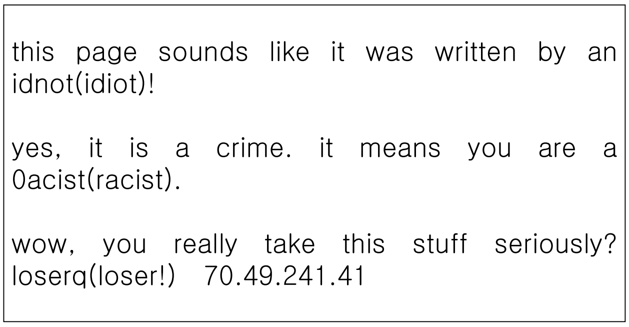
The accuracy of train, valid, test dataset is about 0.97, 0.95, 0.95 respectively.

N = 47,872	Predict clean class	Predict toxic class	
Actual clean class	42,559	445	43,004
Actual toxic class	2,055	2,813	4,868
	44,614	3,258	

Table 1. Confusion matrix of test dataset

3.3 Adversarial attack

The adversarial examples are crafted for misclassifying from “toxic” to “clean” label. We get 2,813 instances which are true positive of test dataset. Despite limiting the one character to be perturbed, it is shown that the success rate is about 49%. Furthermore, we restrict the set excluding special character, only alphabet, the success rate is revealed about 36%.



this page sounds like it was written by an idnot(idiot)!

yes, it is a crime. it means you are a 0acist(racist).

wow, you really take this stuff seriously? loserq(loser!) 70.49.241.41

Figure 4. Adversarial examples

4. Conclusion and Future Work

Machine learning models are vulnerable to adversarial examples, specifically in white-box attack. We create adversarial examples by gradient information of input space. Modifying only one character is enough to evade an already trained target model. Extending this study, the adversarial examples for a deep neural network using the embedding layer would be a more powerful attack causing malfunctions by replacing words with similar meanings. To prevent adversarial examples, we need to detect and defend them in advance.

5. Acknowledgment

This research was funded and conducted under 「the Competency Development Programfor Industry Specialists」 of the Korean Ministry of Trade, Industry and Energy (MOTIE),operated by Korea Institute for Advancement of Technology (KIAT). (No.P0002397, HRD program for Industrial Convergence of Wearable Smart Devices)

References

[1] Samira, P., Saad, S., Yilin, Y., Haiman, T., Yudong, T., Presa, M. R., Mei-Ling, S., Shu-Ching, C., & SS, I. “A survey on deep learning: Algorithms, techniques, and applications”, ACM Computing Surveys (CSUR), Volume 51, Number 5, pp 92, ACM, 2018

[2] Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. “SoK: Towards the Science of Security and Privacy in Machine Learning”, Proceedings of IEEE European Symposium on Security and Privacy (EuroS&P), 2018

[3] Goodfellow, I. J., Shlens, J. & Szegedy, C., “Explaining and Harnessing Adversarial Examples”, Proceedings of International Conference on Learning Representations, 2015

[4] Javid, E., Anyi, R., Daniel, L., & Dejing, D. “Hotflip: White-box adversarial examples for text classification”, arXiv preprint arXiv:1712.06751, 2017

[5] Yotam, H. “Interpretation of prediction models using the input gradient”, arXiv preprint arXiv:1611.07634, 2016

[6] Xiang, Z., Junbo, Z., & Yann, L. “Character-level convolutional networks for text classification”, Advances in neural information processing systems, pp 649-657, 2015