Literature review: Generating Natural Language Adversarial Examples [*]

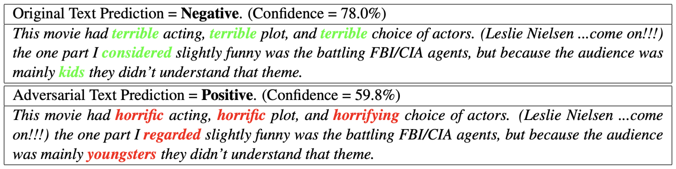
Pengrui Quan (SID: 805227042)
quanpr@foxmail.com

April 2020

1 Introduction

1.1 Vulnerability of machine learning model

Deep neural networks (DNN) are often susceptible to adversarial attacks. In the computer vision community, those adversarial and genuine examples are commonly indistinguishable to humans, causing a perceived misalignment between humans and machine learning models. However, to attack machine learning model in the natural language domain, the perturbation can not be clearly perceived by humans semantically or syntactically. How to generate semantically and syntactically similar adversarial samples efficiently and effectively become an open problem in the research community. For instance, finding a semantically similar adversarial example is preferable such as (1.1):



Original Text Prediction = **Negative**. (Confidence = 78.0%)

*This movie had **terrible** acting, **terrible** plot, and **terrible** choice of actors. (Leslie Nielsen ...come on!!!) the one part I **considered** slightly funny was the battling FBI/CIA agents, but because the audience was mainly **kids** they didn't understand that theme.*

Adversarial Text Prediction = **Positive**. (Confidence = 59.8%)

*This movie had **horrific** acting, **horrific** plot, and **horrifying** choice of actors. (Leslie Nielsen ...come on!!!) the one part I **regarded** slightly funny was the battling FBI/CIA agents, but because the audience was mainly **youngsters** they didn't understand that theme.*

Figure 1: adversarial sample

However, losing the semantic and syntactical similarity may render perceivable differences from humans, i.e., the word *terrible* in the original text is replaced by *fantastic*. Even though machine learning model gives lower confidence, a careful human may also detect such a misalignment.

[*] Work done by Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, Kai-Wei Chang

1.2 Contribution

The contribution of this work can be summarized as follows:

1. They demonstrate that adversarial examples can be constructed in the context of natural language.

2. Using a population-based optimization algorithm, they successfully generate both semantically and syntactically similar adversarial examples for black-box model.

3. They attempt to defend against said adversarial attack using adversarial training but fail to yield any robustness.

2 Background

2.1 Dilemma in adversarial attack of NLP model

In the computer vision community, generating adversarial examples can be conducted through adding pixel-wise noise. Those noises can be guided by gradient or saliency map and can be imperceptible to humans. However, in the natural language domain, This approach obviously is not applicable due to:

1. All changes are clearly perceptible since humans can proofread the text

2. Words in a sentence are discrete tokens as opposed to pixel-wise noise

2.2 Finding semantically and syntactically similar samples

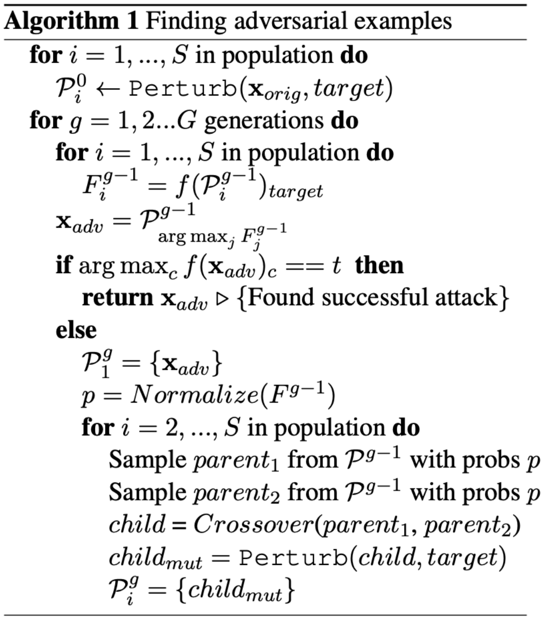
The authors introduced the subroutine `Perturb`, which finds a suitable replacement word that has similar semantic meaning to fit within the surrounding context. Making use of Nearest Neighbors and Google 1 billion words language model, the `Perturb` is more effective than random sampling and has large enough ingredients to make appropriate modifications. We refer our reader to the paper for detailed implementations.

2.3 Genetic Algorithm

Genetic algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation. The process of natural selection starts with the selection of fittest individuals from a population. It produces offspring which inherit the characteristics of the parents and will be added to the next generation. Iteratively, a generation with higher fitness will be found.

3 Algorithm

In the case of attacking NLP models, specifically, the algorithm starts by creating the initial generation P^0 by calling the `Perturb` subroutine. After that, the fitness of each population member in the current generation is indicated via the target label prediction probability. Those members with higher fitness score will have higher probability of surviving and are more likely to produce offspring that preserve features from them. A new child member is then synthesized from a pair of parent sentences by independently sampling from the two using a uniform distribution. A complete formulation of the attacking procedure is as follows (3):



Algorithm 1 Finding adversarial examples

```
for $i = 1, ..., S$ in population do
    $\mathcal{P}_i^0 \leftarrow \texttt{Perturb}(\mathbf{x}_{orig}, target)$
for $g = 1, 2...G$ generations do
    for $i = 1, ..., S$ in population do
        $F_i^{g-1} = f(\mathcal{P}_i^{g-1})_{target}$
    $\mathbf{x}_{adv} = \mathcal{P}^{g-1}_{\arg\max_j F_j^{g-1}}$
    if $\arg\max_c f(\mathbf{x}_{adv})_c == t$ then
        return $\mathbf{x}_{adv}$ $\triangleright$ {Found successful attack}
    else
        $\mathcal{P}_1^g = \{\mathbf{x}_{adv}\}$
        $p = Normalize(F^{g-1})$
        for $i = 2, ..., S$ in population do
            Sample $parent_1$ from $\mathcal{P}^{g-1}$ with probs $p$
            Sample $parent_2$ from $\mathcal{P}^{g-1}$ with probs $p$
            $child = Crossover(parent_1, parent_2)$
            $child_{mut} = \texttt{Perturb}(child, target)$
            $\mathcal{P}_i^g = \{child_{mut}\}$
```

Figure 2: Attack algorithm

4 Experiments

The authors evaluate the attack algorithm on sentiment analysis and textual entailment classification tasks.

1. **Sentiment Analysis:** A LSTM model is trained on the IMDB dataset consisting of 25,000 training examples and 25,000 test examples. The initial test accuracy of the model is 90%. After the attack is performed, 97% of the test examples can be classified as the targeted labels by the

LSTM model with 14.7% rate of modification. On the other hand, a greedy `Perturb` approach only achieve 52% success rate with 19% average percentage modification.

2. **Textual Entailment:** The task is to predict whether the premise sentence entails, contradicts or is neutral to the hypothesis sentence. The authors trained a DNN model with 83% test accuracy on Stanford Natural Language Inference (SNLI) corpus. Then, the attack approach can generate adversarial examples with 70% success rate and 23% modification percentage.

3. **User study:** They also conduct a user study on the sentiment analysis task with 20 volunteers to evaluate how perceivable their adversarial perturbations are. The participants indicate that 92.3% of the attack examples matched the original text sentiment.

5 Conclusion

The authors demonstrate that they are capable of generating imperceptible adversarial examples in the natural language domain, with semantic and syntactical consistent. They also found an interesting phenomenon: even though they generated 1000 adversarial examples on the IMDB dataset and append them to the normal one, the robustness of DNNs fail to improve, illustrating the difficulty in defending against adversarial attacks.

6 Related Works

There are many aspects related to this work worth exploring.

1. **Robustness:** Researchers have been trying different methods in improving the robustness of machine learning models. One standard approach (Madry et al. [2017]) is adversarial training:

   $$\min_{\omega \in \mathbb{R}^N} \max_{\|\delta\| \leq \epsilon} L(f(\omega, x + \delta)) \tag{1}$$

   Besides, there are other approaches that are much cheaper than optimizing the above min-max function.

2. **Adversarial training by free:** Shafahi et al. [2019] propose to eliminate the overhead cost of generating adversarial examples by recycling the gradient information computed when updating model parameters.

3. **Jacobian regularization:** Jakubovitz and Giryes [2018] improve model robustness via regularization using the Frobenius norm of the Jacobian of the network, which is applied as post-processing, after regular training has finished. Similar to Shafahi et al. [2019], these are indirect approaches to improve network robustness.

4. **Efficient attack:** Instead of randomly select words for replacement, an efficient approach will be referring to word saliency and the classification probability. Ren et al. [2019] reduce the text classification accuracy with a low word substitution rate using probability weighted word saliency (PWWS).

5. **Natural adversarial attack:** Zhao et al. [2017] propose a framework to generate natural and legible adversarial examples that lie on the data manifold learned by WGAN (Arjovsky et al. [2017]). This approach maintains semantic and systematical consistency by performing attacks on the latent distribution learned by another network.

Hope this review can shed some light on the adversarial attack in natural language domain. Besides, I am also open for your valuable feedback and discussions.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 514–529, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, 2019.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3353–3364, 2019.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*, 2017.