Reproducibility Report: Synthesizing Robust Adversarial Examples

Prabhant Singh
Institute of Computer Science
University of Tartu
(Dated: January 12, 2018)

The code used to produce the results described in this report is provided in the following GitHub repository:

I. ABSTRACT

In this project we sought to reproduce the results of Synthesizing Robust Adversarial Examples[1]. This paper provided an algorithm which can generate robust adversarial examples that are robust across any chosen distribution of transformations. Authors generated 2D and 3D adversarial examples.

II. INTRODUCTION

A. Theory Overview

1. Adversarial examples

Neural network-based classifiers parallel or exceed human-level accuracy on many common tasks and are used in practical systems. Yet, neural networks are susceptible to adversarial examples, carefully perturbed inputs that cause networks to misbehave in arbitrarily chosen ways[11]. An adversarial example is a sample of input data which has been modified very slightly in a way that is intended to cause a machine learning classifier to misclassify it. In many cases, these modifications can be so subtle that a human observer does not even notice the modification at all, yet the classifier still makes a mistake. Adversarial examples pose security concerns because they could be used to perform an attack on machine learning systems, even if the adversary has no access to the underlying model[?]. Adversarial examples generated using standard techniques like Fast Gradient sign method[3] require complete control over the direct input to the neural network, this is not the case in real world scenarios.[4] For example, standard adversarial examples are not robust under rotation, cropping and various image transformation methods. Expectation over transformation: In this paper[1], the authors introduced a new algorithm, Expectation over transformation to generate robust adversarial examples.

2. Transferability

Adversarial examples are transferable: adversarial examples generated to evade a specific model also mislead other models trained on the same task. Transferability is an obstacle to the deployment of machine learning models as the model is still vulnerable to the adversarial examples though the architecture of the model is not known to the attacker while crafting the inputs. We performed various experiments for the adversarial examples crafted with Expectation over transformation.

3. Types of attacks

There are two goals which can be achieved by generating adversarial examples.

1. Confidence Reduction: Reducing the confidence of the predictions.
2. Misclassification: Using adversarial examples to make a model make wrong classification.

We have two kind of attacks under Misclassification

1. Non-targeted Adversarial Attack. The goal of the non-targeted attack is to slightly modify source image in a way that image will be classified incorrectly by generally unknown machine learning classifier.
2. Targeted Adversarial Attack. The goal of the targeted attack is to slightly modify source image in a way that image will be classified as specified target class by generally unknown machine learning classifier.

B. Reproducibility methodology

To reproduce the results, we first implemented the Expectation over transformation algorithm and crafted examples for InceptionV3[10] model pre-trained on Imagenet data with Tensorflow[9]. Then we generated adversarial examples with various degrees of transformations. We used Tabby Cat image and Adversarial Turtle[5] to generate and check for the robustness and transferability.
1. **Transferability Experiments**

After generating the adversarial examples we used them to check for transferability of those adversarial examples. To check the transferability of the generated adversarial examples we used five pre-trained model architectures:

1. InceptionV3
2. Resnet50
3. Xception
4. VGG 19
5. VGG 16

All the models were trained on Keras[12] with ImageNet dataset[].

2. **Robustness Experiments**

To check for robustness of the generated examples we used the following image transformation techniques:

1. Cropping
2. Zoom In
3. Zoom Out
4. Rotation

III. **RESULTS**

A. **Robustness**

Our generated adversarial examples were robust for a given set transformation. Even the adversarial turtle image which was taken in an uncontrolled setting which was taken in uncontrolled environment was found to be robust and caused misclassification for the given model. Following observations were noticed under Robustness check experiment:

1. Cropped images: Caused targeted misclassification though there was a slight decrease in the confidence of misclassification.
2. Zoom In images: Caused Non-targeted Misclassification.
4. Rotation: Caused Targeted Misclassification with no confidence reduction.

B. **Transferability**

The adversarial examples were found to be highly transferable for all the models. The results for the following experiments can be found in table 1. Generated adversarial image using EOT Parameters:

- Learning rate: 2e-1.
- Epsilon: 8.0/255.0
- True class: Tabby cat
- Target class: Guacamole

Next we used the image of adversarial turtle and checked the transferability for that adversarial example.

- True class: Turtle
- Target class: Rifle
FIG. 3. Adversarial example crafted with EOT

TABLE I. Transferability experiments 1

<table>
<thead>
<tr>
<th>Model</th>
<th>Predicted class</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception V3</td>
<td>Flatworm</td>
<td>100</td>
</tr>
<tr>
<td>Xception</td>
<td>Necklace</td>
<td>92.5</td>
</tr>
<tr>
<td>Resnet50</td>
<td>Tabby Cat</td>
<td>35</td>
</tr>
<tr>
<td>VGG19</td>
<td>Tabby Cat</td>
<td>47.9</td>
</tr>
<tr>
<td>VGG16</td>
<td>Tabby Cat</td>
<td>34.8</td>
</tr>
</tbody>
</table>

Both images of adversarial turtle and cat were detected incorrectly by inception related architectures with a high confidence. The adversarial examples were able to reduce the confidence by a high margin, about 50-60 percent in case of Tabby Cat. Only VGG16 was able to classify the turtle correctly but by a very low confidence of 0.036 Similar results were found when we rotated, cropped and zoomed out of the image. In case of adversarial turtle, the photo was taken out of the distribution (Not inside the chosen distribution as mentioned in the paper i.e camera distance between 2.5cm -3.0cm) ,still the image was misclassified.

TABLE II. Transferability experiments 2

<table>
<thead>
<tr>
<th>Model</th>
<th>Predicted class</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception V3</td>
<td>Pencil Sharpner</td>
<td>67.7</td>
</tr>
<tr>
<td>Xception</td>
<td>Table Lamp</td>
<td>84.8</td>
</tr>
<tr>
<td>Resnet50</td>
<td>Bucket</td>
<td>20</td>
</tr>
<tr>
<td>VGG19</td>
<td>Mask</td>
<td>10.9</td>
</tr>
<tr>
<td>VGG16</td>
<td>Turtle</td>
<td>3.6</td>
</tr>
</tbody>
</table>

IV. DISCUSSIONS

A. Consistency with the original paper

Our results were consistent with the original paper in case of 2D adversarial examples. We were not able to generate 3D adversarial examples.

B. Clarity of the paper and code

We used some snippets of code provided by the authors. Though we found that the paper was lacking a thorough description of the algorithm. The paper seems to be very confident in the later section and comparison of the results with the related work.

C. Access to the dataset

We used Imagenet dataset for generating adversarial examples. In original paper the author is ambiguous weather they used examples from the ImageNet dataset or images from other sources.

D. Computational Requirements

In the original paper there was no mention of the computation requirements to generate adversarial examples. This it is hard for readers to estimate the resources needed to reproduce the results. The authors mentioned 1000 2D adversarial examples for testing. One adversarial example generated with EOT takes upto 30 minutes on a laptop (8GB RAM without GPU). Thus it was possible to generate adversarial examples for checking reproducibility of the paper.

E. Limitations of our approach

Our experiment only generated 2D adversarial examples which were robust towards various image transformation. The main achievement of the paper was 3D robust adversarial examples. As the authors generated 3D adversarial examples with their custom implementation of differentiable rendering software, It was not possible for us to generate 3D adversarial examples.

F. Observations

The following observations were significant but out of the scope of current paper.
1. VGG16 model was robust against adversarial examples to an extent, the possible explanation of this observation can be the sequential architecture of the VGG model and the branched architecture of the Inception Model.

2. Confidence reduction due to out of transformation changes: In the case the adversarial examples was rotated out of the selected degree when generated with Expectation over transformation. The adversarial examples reduced the network confidence as well as induced some confidence of the target class. For example in figure 4 we generated the adversarial example in range of $\pi/4$ and rotated it more than the prescribed range.

FIG. 4. Out of distribution sample causing confidence reduction

V. CONCLUSION

The goal of the paper was to reproduce a subset of experiments in the paper Synthesizing robust adversarial examples. The experiments related to 2D examples was successfully reproduced. The transferability and robustness of adversarial examples was successfully verified on different models.

G. Open Questions

1. What can be possible defenses against these adversarial examples?

2. Performance of the adversarial examples generated with Expectation Over Transformation on the ensemble of neural networks. As some studies suggest that Ensemble training of neural networks might make them robust to adversarial examples to an extent, it will be interesting to see if Expectation over transformation examples can be robust against them too or not.

H. Suggestions for improving reproducibility

1. Open sourcing the custom implementation of Differentiable 3D renderer to produce 3D adversarial examples.

2. Providing better theoretical description of Expectation Over Transformation Algorithm.

I. Suggestions for future work

1. Further investigation on the reason Why VGG was able to outperform various inception based architectures while checking for transferability.

2. Generating adversarial examples for ResNet50 and checking the transferability on Inception based architectures.

REFERENCES


[5] Picture of the original adversarial turtle mentioned in the paper taken with an Iphone5


[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei Dept. of Computer Science, Princeton University, USA
ImageNet Large Scale Visual Recognition Challenge

Martin Abadi and Paul Barham and Jianmin Chen and Zhifeng Chen and Andy Davis and Jeffrey Dean and Matthieu Devin and Sanjay Ghemawat and Geoffrey Irving and Michael Isard and Manjunath Kudlur and Josh Levenberg and Rajat Monga and Sherry Moore and Derek G. Murray and Benoit Steiner and Paul Tucker and Vijay Vasudevan and Pete Warden and Martin Wicke and Yuan Yu and Xiaoqiang Zheng

TensorFlow: A system for large-scale machine learning

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna

Rethinking the Inception Architecture for Computer Vision arXiv:1512.00567

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus

Intriguing properties of neural networks

keras

https://keras.io/

VI. APPENDIX

Images used during experiments:

FIG. 5. Adversarial Cat
FIG. 6. Adversarial turtle 1

FIG. 7. Adversarial turtle 2

FIG. 8. Adversarial turtle 3