Testing and Analyzing Sample Attack Generation Algorithms

Mentee: Rittika Adhikari | Mentor: Wei Yang
University of Illinois Department of Computer Science

Purpose

Deep Neural Networks are increasingly being used to facilitate everyday life. This increasing use of DNNs requires the network to be more robust in order to avoid potentially dangerous misclassification. How can we best create training/validation/test datasets that consider more possible boundary cases?

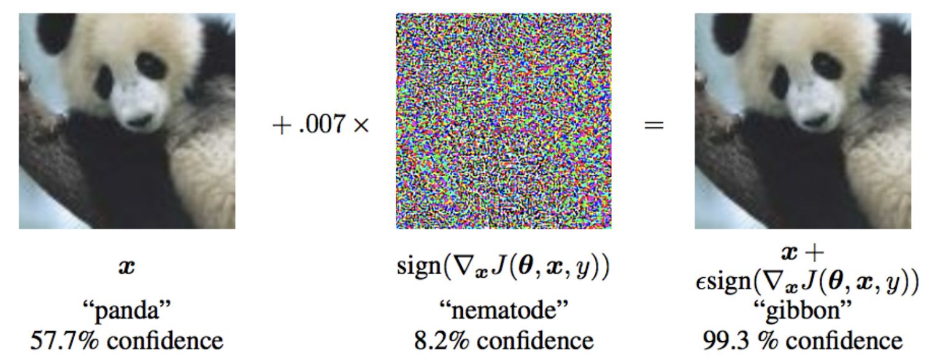
Background

Adversarial Attacks:

- Samples from the training dataset with an additional worst-case perturbation overlayed on image
- **FGSM =** fast gradient sign method
  - Iteratively compute the sign of the gradient of the targeted neural network's cost function

$$\boldsymbol{\eta} = \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right)$$

  - Overlay the computed value as a "perturbation" to the original training image


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

Adding a perturbation generated by FGSM to original image

Neuron Coverage:

- Measures how much of a neural network's logic is exercised
  - # of activated neurons / total # of neurons

$$NCov(T, \boldsymbol{x}) = \frac{|\{n \mid \forall \boldsymbol{x} \in T, out(n, \boldsymbol{x}) > t\}|}{|N|}$$

DeepXplore:

- Generates test inputs that maximize neuron coverage and differential behaviors

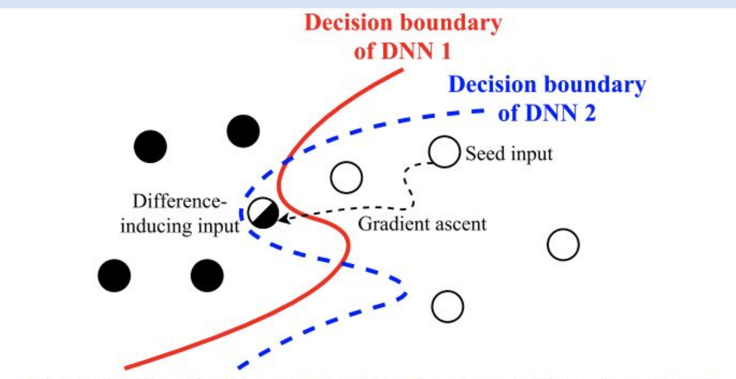
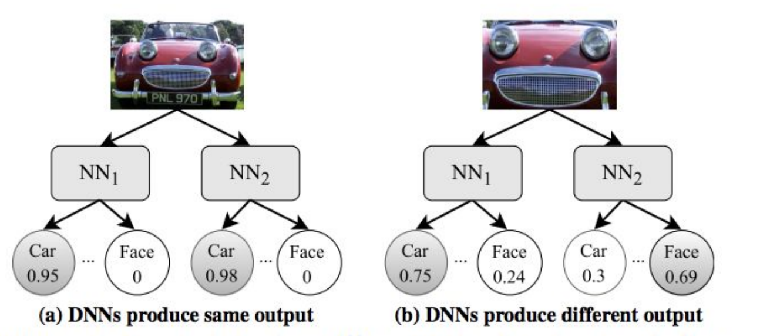

Decision boundary of DNN 1
Decision boundary of DNN 2
Seed input
Difference-inducing input
Gradient ascent

Figure 7: Gradient ascent starting from a seed input and gradually finding the difference-inducing test inputs.



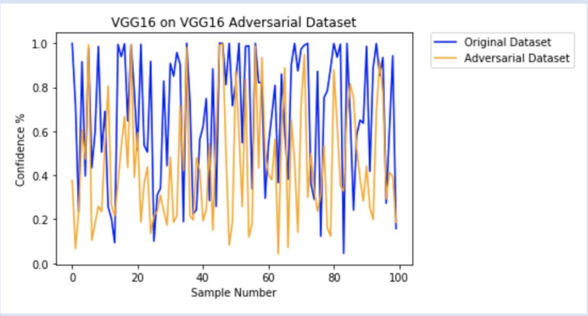
(a) DNNs produce same output (b) DNNs produce different output

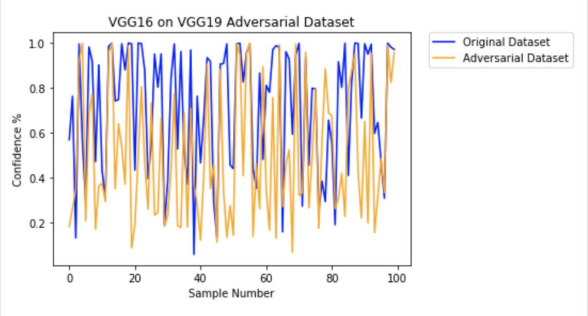
Figure 6: Inputs inducing different behaviors in two similar DNNs.

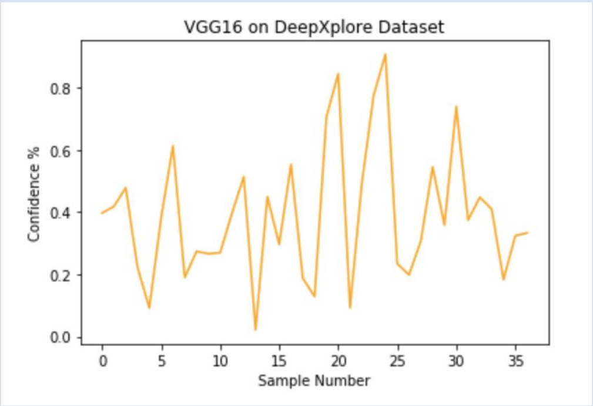
Procedure

1) Utilized the cleverhans github repository to generate adversarial examples for VGG-16, VGG-19, and ResNet50.
2) Used the previously generated adversarial examples as test data for each of the neural networks and calculated accuracy.
3) Calculated the predictive accuracy of each of the neural networks on the DeepXplore-generated data.

Progress and Preliminary Results







These are "confidence percentage" graphs, which depict the percentage of confidence that the neural network has on that particular sample. As can be seen, the confidence in an accurate prediction is *low*, whereas the confidence in an inaccurate prediction is *high.*

Through my analysis of the sample attack generation algorithms, I found that the adversarial datasets typically caused a lower predictive accuracy than the DeepXplore datasets, thereby demonstrating the transferability of the FGSM-generated attacks.

Skills Gained

Explored the cleverhans github repository & worked with tensorflow to modify the code to run on different pretrained neural nets from keras.

Used matplotlib to plot the "confidence percentage" of the neural net on each sample.

Future Work

1) Calculate the neuron coverage for each of the adversarial datasets and the DeepXplore dataset.
2) Try to develop a comprehensive mechanism to see what constitutes as a "better" test case dataset.

Acknowledgments

Thank you to my mentor, Wei Yang, for giving me this opportunity and helping me learn more about adversarial machine learning.