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?

### Purpose

- **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
    - \( \eta = \text{esign}(\nabla_x J(\theta, x, y)) \)
    - Overlay the computed value as a “perturbation” to the original training image

- **Neuron Coverage:** Measures how much of a neural network’s logic is exercised
  - \# of activated neurons / total \# of neurons

- **DeepXplore:** Generates test inputs that maximize neuron coverage and differential behaviors

### 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
    - \( \eta = \text{esign}(\nabla_x J(\theta, x, y)) \)
    - Overlay the computed value as a “perturbation” to the original training image

- **Neuron Coverage:** Measures how much of a neural network’s logic is exercised
  - \# of activated neurons / total \# of neurons

- **DeepXplore:** Generates test inputs that maximize neuron coverage and differential behaviors

### Progress and Preliminary Results

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.