# Adversarial Machine Learning Lab

A graph with black lines and white text

Description automatically generated

Adversarial attack refers to a technique used to fool models through malicious inputs. Such techniques may be particularly relevant in computer vision, natural language processing and other domain where ML models are implemented. Through the process, original inputs are modified to lead to incorrect outputs by the model, while the modifications remain imperceptible to human eyes. Adversarial attacks can be broadly categorized into two types:

White box attacks- The attacker has complete knowledge of the model including its architecture, parameters and training data, that is used to alter the inputs to the model to produce incorrect guesses. Fast Gradient Sign Method (FGSM) is an example.

Black box attacks- The attacker has no knowledge of the model’s internals and only attempts to alter inputs by hit and trial to distort the outputs.

Adversarial attacks aim to achieve at least one of the following goals:

Misclassification- making the model to classify the input incorrectly.

Source/target misclassification- making the model misclassify inputs from one class to another predetermined class.

Confidence reduction – Reducing the model’s efficiency and confidence in correct classification, without necessarily misclassification.

1. **Briefly explain FGSM.**

FGSM is a popular white box attack technique due to its simplicity and efficiency. Basic ideas behind FGSM is to take an original input image and apply small modifications to it, to maximize the loss of a neural network on that input. The modified input however looks identical to original to humans, but is classified incorrectly by the model.

It computes the gradient of the loss w.r.t input image, which is used to modify the pixels of the original input image. Accompanied by an ‘Epsilon’ value used to scale the alterations to ensure it is small yet significant, forming the ‘Adversarial image’.

Adversarial Image=Original Image+*ϵ*⋅sign(∇Image​Loss)

where

∇ImageLoss∇Image​Loss: gradient of the loss with respect to the image,

sign (⋅) : the sign function applied to the gradient,

Epsilon parameter (ϵ): controls the magnitude of the perturbation.

1. **Briefly explain the steps involved in the notebook. You do not need to explain code sections or any concepts that involve neural networks - just a few bullet points about what each code section does (this does not require screenshots of code sections, just discuss the concepts used in each section**

* Set up: Importing the Necessary Libraries- TensorFlow, matplotlib and pre-trained CNN model.
* Selecting an image from the images database (ImageNet)- ‘Yellow Labrador’ image is taken an example.
* Preprocess an image of Labrador -(just resizing and scaling the pixel values) in order to make it suitable for classification by the model.
* Model is fed with the preprocessed original image to be classified correctly to establish a baseline for comparisons.
* Creating Adversarial images:
* Perturbations are created (modified alterations in the original image) barely visible to human eyes.
* Impact of Epsilon-Determining the magnitude of change to be applied to original image.
* The Model then classifies the adversarial image showing the change in model’s predictions.
* Misclassified Images would display and look identical to original image for human eye but would be classified to entirely different object/breed by AI model.

1. **What does the original image look like?**

The original image here is of a famous dog breed ‘Labrador’(yellow color), which would be correctly classified by AI model when provided as input.

**4. What do the perturbations look like?**

Perturbations are imperceptible changes or modifications to the original image that are meant to deceive AI into making wrong predictions. And the calculation is based on gradient of the loss with respect to input image. These perturbations look like a pattern of noise overlaid on the image/distortions of the original pixels.

1. **How does the value of Epsilon impact "fooling" the model?**

Value of epsilon determined the magnitude of perturbations applied to in the input image.

Effect on model accuracy with Confidence and perceptibility of perturbations is observed in the notebook by varying epsilon.

Whereas a small ε means less noticeable changes to the image, which might not fool the model effectively. A larger ε increases the changes, making it more likely to fool the model into misclassification and more likely that the changes can be detected by human observers or possibly exceed the natural image range, leading to obvious artifacts.

The goal is to find the balance where epsilon is large enough to cause the model to misclassify the input but small enough that the changes are not easily noticeable.

**6. What do the misclassified images look like (i.e. show a picture of each type of dog that the model misclassified)?**

Misclassified images would look almost identical to the original image but have been just altered enough for AI’s wrong prediction. Almost all the misclassified images here resembled the same yellow Labrador, but fooled AI with varied confidence. With increasing value of Epsilon, the Model’s conviction gets better at wrong prediction of the Object image (yellow Labrador) to be something else.

A dog standing on grass

Description automatically generated

A dog standing on grass

Description automatically generatedA dog standing on grass

Description automatically generatedA dog standing on grass

Description automatically generated

*Illustrated the effects of different levels of Adversarial perturbation (generated by FGSM) on image recognition model’s predictions.*

\*Confidence Score- numerical value that represents the probability assigned by the model to its prediction. It reflects model’s learned patterns from training data and how well those patterns match the input image.

**Top left:** is Original Unperturbed image of Labrador with the label ‘Labrador Retriever’.

In this case, a confidence score of 41.82% is relatively low, suggesting that the model is not very certain about its classification. This could be due to several factors such as the quality of the image, the position or posture of the dog, the training data used to train the model, or the complexity of the model itself.

**Other three:** Withincreasing Epsilon 0.01,0.1 and 0.15 for each of the three images, the confidence level drops significantly, model wrongly predicts the dog to be of other breed types -Saluki and Weimaraner. The progression of these images demonstrates how increasing the intensity of adversarial perturbations can systematically degrade a model's ability to make correct prediction. However all the three images resemble the original image of ‘Yellow colored Labrador’ to human eyes.

Few of the Applications of Fast Gradient Sign Method (FGSM):

**Improving AI Security**: In Facial recognition systems used for security purposed, it can help identify weaknesses that attackers might exploit to bypass the system. Developers can address those vulnerabilities making the system more secure against unauthorized access.

**Enhancing Privacy:** Quite relevant in the context of social media where individuals might wish to safeguard their privacy from automated tracking.

**Automotive Safety:** Manufacturer of autonomous vehicles can test their perception systems against potential visual manipulations (altered road safety signs) and improve safe decision making even when faced with deceptive inputs.

**Medical Imaging analysis:** There is vulnerability in Computer algorithms used in medical imaging diagnosis to small changes introduced by methods like the Fast Gradient Sign Method (FGSM). By applying FGSM to alter just one or a few pixels in medical images, researchers tested the robustness of these diagnostic programs. The results indicated that even minor, carefully crafted disruptions can cause the algorithms to misread the images, which could potentially lead to false medical diagnoses.

References:

<https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/generative/adversarial_fgsm.ipynb#scrollTo=3DA8g-Zp69J4>

<https://neptune.ai/blog/adversarial-attacks-on-neural-networks-exploring-the-fast-gradient-sign-method>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10487122/>

<https://chat.openai.com/>