**ADVERSARIAL MACHINE LEARNING**

**LIBRARIES**

* Tensorflow
* Keras
* Numpy

**MODEL**

# Define the model architecture

model = tf.keras.Sequential([

    tf.keras.layers.Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(28, 28, 1)),

    tf.keras.layers.MaxPooling2D(pool\_size=(2, 2)),

    tf.keras.layers.Flatten(),

    tf.keras.layers.Dense(128, activation='relu'),

    tf.keras.layers.Dense(10, activation='softmax')

])

# Results

Test loss: 0.0525

Test accuracy: 98.42%

**DATASET USED**

**MNIST**

The MNIST dataset is a widely used dataset in the field of machine learning and computer vision. MNIST stands for Modified National Institute of Standards and Technology, which is the organization that originally collected and processed the dataset. It consists of a large collection of handwritten digits, from 0 to 9, each represented as a 28x28 grayscale image.

The MNIST dataset is often used as a benchmark for evaluating and comparing various machine learning algorithms, particularly in the context of image classification tasks. It has become a standard dataset for beginners to practice and learn about machine learning and deep learning techniques.

The dataset contains 60,000 training examples and 10,000 testing examples. The training set is typically used to train machine learning models, while the testing set is used to evaluate the performance of the trained models on unseen data.

Each image in the MNIST dataset is labeled with the corresponding digit it represents, allowing for supervised learning approaches. The labels are represented as integers from 0 to 9. The goal is to train a model that can correctly classify new, unseen images into the correct digit category.

Over the years, the MNIST dataset has served as a starting point for many researchers and has helped pave the way for advancements in machine learning and computer vision. It has been used to develop and benchmark various algorithms, including traditional machine learning methods as well as deep learning models such as convolutional neural networks (CNNs).

While the MNIST dataset has been valuable for research and education purposes, it is worth noting that it is a relatively simple dataset compared to real-world problems. As a result, achieving high accuracy on MNIST does not necessarily guarantee good performance on more complex and diverse datasets. Nonetheless, it remains an important resource for understanding and experimenting with machine learning techniques.

**ATTACKS**

1. **Fast Gradient Sine Method**
2. **Progressive Gradient Descent**
3. **Basic Iteration Method**
4. **Deep Fool**
5. **Carlini & Wagner**

**MITIGATION TECHNIQUES**

1. **Adversarial Training**
2. **Median Filtering**

**OBSERVATION**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attacks** | **FGSM** | **PGD** | **BIM** | **Deep Fool** | **C&W** |
| **After attack** | 64 | 14.38 | 9.66 | 10 | 6 |
| **After Adv Training** | 99 | 97.5 |  |  | NA |

*Table: Accuracy of model after attack (in %)*

|  |  |  |  |
| --- | --- | --- | --- |
| **eps** | **eps\_step** | **After attack** | **After Adv Training** |
| 0.1 | 0.01 | 5 | 98 |
| 0.1 | 0.1 | 17.5 | 97.3 |

*Table: PGD - Accuracy of model with varied eps & eps\_step (in %)*

**RESULTS**

**WORK IN PROGRESS**

* Accuracy variation after Adversarial Training on BIM & Deep Fool
* Using alternative dataset ( other than MNIST )
* Implement of pre-existing models like ResNet-50, EfficientNet, MobineNet, etc & trying on

some lighter variants of Vision Transformer, EfficentFormer, etc.

* Exploring Deep Robust Toolkit, in place of ART ( <https://github.com/DSE-MSU/DeepRobust> )
* Trying other attacks and mitigation techniques

PENDING WORK

* Mitigation Technique - Median filtering