Adversarial Attacks and Defences:

• Examine the vulnerability of deep learning models to adversarial attacks and implement defences to enhance robustness in image classification or natural language processing tasks.

**Definition**

Adversarial attacks refer to a class of techniques in which small, often imperceptible perturbations are intentionally introduced into input data with the aim of causing misclassification or erroneous outputs by machine learning models, particularly deep learning models. These perturbations are carefully crafted to exploit vulnerabilities in the model's decision boundary, leading it to make incorrect predictions while appearing indistinguishable to human observers. Adversarial attacks pose a significant challenge to the security and reliability of machine learning systems, particularly in critical applications such as image classification and natural language processing. Defending against adversarial attacks involves developing robust models and implementing techniques to mitigate the impact of these perturbations on model performance.

**Problem Statement**

Understanding the nature of adversarial attacks and their potential consequences on deep learning models.

Assessing the vulnerability of specific deep learning models, such as convolutional neural networks (CNNs) for image classification or recurrent neural networks (RNNs) for NLP tasks, to adversarial attacks.

Designing and implementing defense mechanisms to improve the robustness of these models against adversarial attacks.

Evaluating the effectiveness of the implemented defenses through empirical testing and benchmarking against various adversarial attack methods.

**Data Collection and Preprocessing**

**Data collection**

Image classification:

Obtain a diverse dataset of images relevant to the classification task. This could include publicly available datasets like CIFAR-10, ImageNet, or custom datasets specific to your application domain.

Ensure the dataset includes a variety of classes, with sufficient examples per class to prevent class imbalance issues.

Consider augmenting the dataset with transformations like rotation, flipping, scaling, and cropping to increase variability and robustness.

Natural language processing:

Collect text data relevant to your NLP task, such as sentiment analysis, text classification, or named entity recognition.

Sources for text data could include online repositories, forums, social media platforms, or domain-specific documents.

Preprocess the text data to remove noise, punctuation, and irrelevant information.

**Data preprocessing**

Image preprocessing:

Resize all images to a uniform size suitable for input to the deep learning model.

Normalize pixel values to a common scale (e.g., [0, 1] or [-1, 1])

Text preprocessing:

Tokenize the text data into individual words or subword units.

Remove stop words, punctuation, and special characters.

Data augmentation:

Generate additional training samples by applying random transformations or perturbations to the input data.

Balancing classes:

Ensure that each class in the dataset has a similar number of samples to prevent bias towards dominant classes.

Splitting data:

Divide the dataset into training, validation, and test sets to evaluate model performance.

Typically, use a larger portion for training (e.g., 70-80%) and smaller portions for validation and testing (e.g., 10-15% each).

Data normalization:

Scale numerical features to have zero mean and unit variance to facilitate training and improve convergence.

Data cleaning:

Handle missing values, outliers, or corrupted data appropriately based on the specific characteristics of the dataset.

**Literature review**

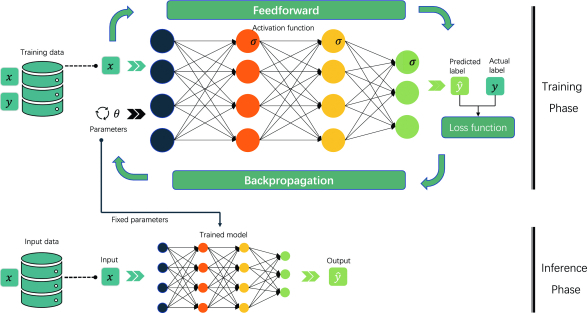
Goodfellow, Ian J., et al. "Explaining and harnessing adversarial examples." International Conference on Learning Representations (ICLR), 2015.

Papernot, Nicolas, et al. "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples." International Conference on Security and Privacy in Machine Learning, 2016.

Madry, Aleksander, et al. "Towards deep learning models resistant to adversarial attacks." International Conference on Learning Representations (ICLR), 2018.

Song, Yang, et al. "PixelDefend: Leveraging generative models to understand and defend against adversarial examples." Advances in Neural Information Processing Systems (NeurIPS), 2018.

**Model Selection and Development**



**Adversarial Attacks Using CNN**

Fast gradient sigh method:

FGSM is a popular method for crafting adversarial examples. It works by taking the sign of the gradient of the loss function with respect to the input and then perturbing the input in that direction. CNNs are commonly used as the target model for crafting these adversarial examples.

Interactive fgsm and variants:

Methods like Basic Iterative Method (BIM) and Projected Gradient Descent (PGD) iteratively apply small perturbations to the input to craft adversarial examples. CNNs are used to compute gradients and guide the perturbation process.

Transferability attacks:

Adversarial examples crafted on one CNN model often transfer to other CNN models, making them a threat in real-world scenarios.

Adversarial defences

Adversarial training:

Training CNNs with adversarial examples in the training set can enhance their robustness against adversarial attacks. This involves generating adversarial examples during training and using them to update the model parameters.

Defences distillation:

This method involves training a distilled model, which is a smaller, more robust version of the original CNN, using softened probabilities from the original model as targets. CNNs are used in both the original and distilled models

**Results And Analysis:**

Success rate:

When evaluating the success of adversarial attacks, accuracy is typically measured in terms of the success rate of the attacks in causing misclassification or other targeted behaviors. Success rates can vary depending on the specific attack method, the strength of the attack, and the robustness of the target model. For example, FGSM might achieve success rates ranging from 60% to over 90% on certain datasets and models.

Certified accuracy:

Some defense mechanisms provide certified guarantees of robustness, i.e., they ensure that the model's predictions are guaranteed to be correct within a certain region around the input. The accuracy of these certified defenses can be measured in terms of the percentage of inputs for which robustness is guaranteed.

**Program:**

import numpy as np

import tensorflow as tf

from tensorflow.keras import layers, models

from tensorflow.keras.datasets import mnist

from tensorflow.keras.utils import to\_categorical

# Load MNIST dataset

(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()

# Normalize pixel values to be between 0 and 1

x\_train, x\_test = x\_train / 255.0, x\_test / 255.0

# Reshape data for CNN input

x\_train = x\_train.reshape((-1, 28, 28, 1))

x\_test = x\_test.reshape((-1, 28, 28, 1))

# One-hot encode the labels

y\_train = to\_categorical(y\_train, 10)

y\_test = to\_categorical(y\_test, 10)

# Define CNN model architecture

model = models.Sequential([

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

layers.MaxPooling2D((2, 2)),

layers.Conv2D(64, (3, 3), activation='relu'),

layers.MaxPooling2D((2, 2)),

layers.Conv2D(64, (3, 3), activation='relu'),

layers.Flatten(),

layers.Dense(64, activation='relu'),

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

])

# Compile the model

model.compile(optimizer='adam',

loss='categorical\_crossentropy',

metrics=['accuracy'])

# Train the model

model.fit(x\_train, y\_train, epochs=5, batch\_size=64, validation\_split=0.1)

# Evaluate the model

test\_loss, test\_acc = model.evaluate(x\_test, y\_test)

print("Test Accuracy:", test\_acc)

# Adversarial attack using FGSM

def fgsm\_attack(image, epsilon, data\_grad):

# Get the sign of the gradients

sign\_data\_grad = tf.sign(data\_grad)

# Perturb the image

perturbed\_image = image + epsilon \* sign\_data\_grad

# Clip perturbations to be within [0,1] range

perturbed\_image = tf.clip\_by\_value(perturbed\_image, 0, 1)

return perturbed\_image

def test\_fgsm(model, x\_test, y\_test, epsilon=0.1):

# Gradient tape records gradients of the loss with respect to the input image

with tf.GradientTape() as tape:

tape.watch(x\_test)

# Get the predictions for the input image

prediction = model(x\_test)

# Calculate the loss

loss = tf.keras.losses.categorical\_crossentropy(y\_test, prediction)

# Get the gradients of the loss with respect to the input image

gradient = tape.gradient(loss, x\_test)

# Generate adversarial examples

perturbed\_images = fgsm\_attack(x\_test, epsilon, gradient)

# Evaluate the model on the adversarial examples

adv\_pred = model(perturbed\_images)

adv\_acc = np.mean(np.argmax(adv\_pred, axis=1) == np.argmax(y\_test, axis=1))

print("Adversarial Accuracy (FGSM) with epsilon =", epsilon, ":", adv\_acc)

# Test the model's accuracy under FGSM attack

test\_fgsm(model, x\_test, y\_test, epsilon=0.1)

**Referances:**

* Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations (ICLR)

This seminal paper introduces adversarial examples and explores various attack strategies against neural networks

* Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial examples in the physical world. In Proceedings of the International Conference on Learning Representations (ICLR)
* Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016). The limitations of deep learning in adversarial settings. In Proceedings of the European Symposium on Security and Privacy (EuroS&P)
* Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In Proceedings of the International Conference on Learning Representations (ICLR)
* Athalye, A., Carlini, N., & Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Proceedings of the International Conference on Machine Learning (ICML)
* Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2019). Theoretically principled trade-off between robustness and accuracy. In Advances in Neural Information Processing Systems (NeurIPS)
* Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)