В среде Google Colab реализовать атаку Deepfool на датасет MNIST.

a. Загрузить необходимые библиотеки и установить пакет (!pip install foolbox):

import numpy as np

import tensorflow as tf

import foolbox

import eagerpy as ep

import matplotlib.pyplot as plt

from tensorflow.keras import datasets, layers, models

b. Загрузить датасет:

(train\_images, train\_labels), (test\_images, test\_labels) = datasets.mnist.load\_data()

c. Выполнить нормализацию данных:

train\_images, test\_images = train\_images / 255.0, test\_images / 255.0

d. Зададим архитектуру нейронной сети:

model = models.Sequential([

    layers.Flatten(input\_shape=(28, 28)),

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

    layers.Dense(10)

])

e. Компиляция модели:

model.compile(optimizer='adam',

              loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True),

              metrics=['accuracy'])

f. Обучить модель:

model.fit(train\_images, train\_labels, epochs=5)

g. Создать модель foolbox:

fmodel = foolbox.models.TensorFlowModel(model, bounds=(0, 1))

h. Выбрать случайное тестовое изображение:

idx = np.random.randint(0, len(test\_images))

image, label = test\_images[idx].astype(np.float32), test\_labels[idx]

i. Преобразовать метку в тип данных, который совместим с EagerPy:

label = np.array([label], dtype=np.int64)

j. Преобразовать изображение в тензор TensorFlow:

image\_tensor = tf.convert\_to\_tensor(image.reshape((1, 28, 28, 1)))

k. Создать атаку Deepfool:

attack = foolbox.attacks.L2DeepFoolAttack()

L. Запустить атаку на изображение с указанием epsilons:

epsilons = [0.01]  # Пример значения, может потребоваться настройка

adversarial = attack(fmodel, image\_tensor, label, epsilons=epsilons)

M. Визуализировать изображения:

plt.figure()

plt.subplot(1, 3, 1)

plt.title("Original")

plt.imshow(image.squeeze(), cmap="gray")

plt.subplot(1, 3, 2)

plt.title("Adversarial")

plt.imshow(adversarial[0][0].numpy().squeeze(), cmap="gray")

plt.subplot(1, 3, 3)

plt.title("Difference")

plt.imshow(adversarial[0][0].numpy().squeeze() - image.squeeze(), cmap="gray")

plt.show()

1. Объяснить что происходит в каждой ячейке выполненного кода.

2. Предоставить отчет в формате pdf.