МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

ИКБ направление «Кибербразведка и противодействие угрозам с применением технологий искусственного интеллекта» 10.04.01

Кафедра КБ-4 «Интеллектуальные системы информационной безопасности»

Лабораторная работа №1

по дисциплине

«Анализ защищенности систем искусственного интеллекта»

Группа:
ББМО-01-22
Выполнил:
Воронов А.И.

Проверил:
Спирин А.А.

Москва 2023

Номер в таблице: 15.

Сменим среду выполнения на T4 GPU и выполним клонирование репозитория с гита.

Сразу меняем среду выполенения на T4 GPU

```
[ ]  # клонируем репозиторий
     !git clone https://github.com/ewatson2/EEL6812_DeepFool_Project

     Cloning into 'EEL6812_DeepFool_Project'...
     remote: Enumerating objects: 96, done.
     remote: Counting objects: 100% (3/3), done.
     remote: Compressing objects: 100% (2/2), done.
     remote: Total 96 (delta 2), reused 1 (delta 1), pack-reused 93
     Receiving objects: 100% (96/96), 33.99 MiB | 15.14 MiB/s, done.
     Resolving deltas: 100% (27/27), done.
```

Проверим, что репозиторий загружен и переходим в новую директорию.

```
[ ]  # проверяем, что репозиторий загружен
     !ls

     EEL6812_DeepFool_Project  sample_data

[ ]  # переходим в директорию "EEL6812_DeepFool_Project"
     %cd EEL6812_DeepFool_Project

     /content/EEL6812_DeepFool_Project
```

Импортируем необходимые библиотеки и присвоим параметру rand_seed номер студента (15) из таблицы.

```
[ ]  # импортируем библиотеки
     import numpy as np
     import json
     import torch
     from torch.utils.data import DataLoader, random_split
     from torchvision import datasets, models
     from torchvision.transforms import transforms

[ ]  from models.project_models import FC_500_150, LeNet_CIFAR, LeNet_MNIST, Net
     from utils.project_utils import get_clip_bounds, evaluate_attack, display_attack

[ ]  # номер в таблице 16 - 1 за шапку таблицы = 15
     rand_seed = 15
```

Выполним загрузку датасета MNIST.

```
[ ] # Загрузка датасета MNIST
    mnist_mean = 0.5
    mnist_std = 0.5
    mnist_dim = 28

    from utils.project_utils import get_clip_bounds
    mnist_min, mnist_max = get_clip_bounds(mnist_mean, mnist_std, mnist_dim)

    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    mnist_min = mnist_min.to(device)
    mnist_max = mnist_max.to(device)

    mnist_tf = transforms.Compose([
        transforms.ToTensor(),
        transforms.Normalize(mean=mnist_mean, std=mnist_std)
    ])

    mnist_tf_train = transforms.Compose([
        transforms.RandomHorizontalFlip(),
        transforms.ToTensor(),
        transforms.Normalize(mean=mnist_mean, std=mnist_std)
    ])

    mnist_tf_inv = transforms.Compose([
        transforms.Normalize(mean=0.0, std=np.divide(1.0, mnist_std)),
        transforms.Normalize(mean=np.multiply(-1.0, mnist_std), std=1.0)
    ])

    mnist_temp = datasets.MNIST(root='datasets/mnist', train=True, download=True, transform=mnist_tf_train)
    mnist_train, mnist_val = random_split(mnist_temp, [50000, 10000])

    mnist_test = datasets.MNIST(root='datasets/mnist', train=False, download=True, transform=mnist_tf)
```

```
Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz to datasets/mnist/MNIST/raw/train-images-idx3-ubyte.gz
100%|██████████| 9912422/9912422 [00:00<00:00, 116367632.14it/s]Extracting datasets/mnist/MNIST/raw/train-images-idx3-ubyte.gz to datasets/mnist/MNIST/raw


Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz to datasets/mnist/MNIST/raw/train-labels-idx1-ubyte.gz
100%|██████████| 28881/28881 [00:00<00:00, 112162679.47it/s]
Extracting datasets/mnist/MNIST/raw/train-labels-idx1-ubyte.gz to datasets/mnist/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz to datasets/mnist/MNIST/raw/t10k-images-idx3-ubyte.gz
100%|██████████| 1648877/1648877 [00:00<00:00, 38459009.57it/s]Extracting datasets/mnist/MNIST/raw/t10k-images-idx3-ubyte.gz to datasets/mnist/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz

Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz to datasets/mnist/MNIST/raw/t10k-labels-idx1-ubyte.gz
100%|██████████| 4542/4542 [00:00<00:00, 21381064.84it/s]
Extracting datasets/mnist/MNIST/raw/t10k-labels-idx1-ubyte.gz to datasets/mnist/MNIST/raw
```

Выполним загрузку CIFAR-10.

```
[ ] # загрузка датасета CIFAR-10
    cifar_mean = [0.491, 0.482, 0.447]
    cifar_std = [0.202, 0.199, 0.201]
    cifar_dim = 32

    from utils.project_utils import get_clip_bounds
    cifar_min, cifar_max = get_clip_bounds(cifar_mean, cifar_std, cifar_dim)

    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    cifar_min = cifar_min.to(device)
    cifar_max = cifar_max.to(device)

    cifar_tf = transforms.Compose([
        transforms.ToTensor(),
        transforms.Normalize(mean=cifar_mean, std=cifar_std)
    ])

    cifar_tf_train = transforms.Compose([
        transforms.RandomCrop(size=cifar_dim, padding=4),
        transforms.RandomHorizontalFlip(),
        transforms.ToTensor(),
        transforms.Normalize(mean=cifar_mean, std=cifar_std)
    ])

    cifar_tf_inv = transforms.Compose([
        transforms.Normalize(mean=[0.0, 0.0, 0.0], std=np.divide(1.0, cifar_std)),
        transforms.Normalize(mean=np.multiply(-1.0, cifar_mean), std=[1.0, 1.0, 1.0])
    ])

    cifar_temp = datasets.CIFAR10(root='datasets/cifar-10', train=True, download=True, transform=cifar_tf_train)
    cifar_train, cifar_val = random_split(cifar_temp, [40000, 10000])

    cifar_test = datasets.CIFAR10(root='datasets/cifar-10', train=False, download=True, transform=cifar_tf)

    cifar_classes = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']
```

```
Downloading https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz to datasets/cifar-10/cifar-10-python.tar.gz
100%|██████████| 170498071/170498071 [00:20<00:00, 8288831.08it/s]
Extracting datasets/cifar-10/cifar-10-python.tar.gz to datasets/cifar-10
Files already downloaded and verified
```

Выполним загрузку и настройку DataLoader.

```
[ ]  # Выполняем настройку и загрузку DataLoader
     batch_size = 64
     workers = 4

     mnist_loader_train = DataLoader(mnist_train, batch_size=batch_size, shuffle=True, num_workers=workers)
     mnist_loader_val = DataLoader(mnist_val, batch_size=batch_size, shuffle=False, num_workers=workers)
     mnist_loader_test = DataLoader(mnist_test, batch_size=batch_size, shuffle=False, num_workers=workers)

     cifar_loader_train = DataLoader(cifar_train, batch_size=batch_size, shuffle=True, num_workers=workers)
     cifar_loader_val = DataLoader(cifar_val, batch_size=batch_size, shuffle=False, num_workers=workers)
     cifar_loader_test = DataLoader(cifar_test, batch_size=batch_size, shuffle=False, num_workers=workers)

     /usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560: UserWarning: This DataLoader will create 4 w
       warnings.warn(_create_warning_msg(
```

Далее загружаем и оцениваем стойкость модели Network-In-Network к DeepFool атакам.

```
[ ]  # Загрузка и оценка стойкости модели Network-In-Network Model к FGSM и DeepFool атакам
     fgsm_eps = 0.2

     model = Net().to(device)
     model.load_state_dict(torch.load('weights/clean/cifar_nin.pth', map_location=device))

     evaluate_attack('cifar_nin_fgsm.csv', 'results', device, model, cifar_loader_test, cifar_min, cifar_max, fgsm_eps, is_fgsm=True)
     print('')

     deep_args = [64, 10, 0.02, 100]
     evaluate_attack('cifar_nin_deepfool.csv', 'results', device, model, cifar_loader_test, cifar_min, cifar_max, deep_args, is_fgsm=False)

     if device.type == 'cuda': torch.cuda.empty_cache()

     FGSM Test Error : 81.29%
     FGSM Robustness : 1.77e-01
     FGSM Time (All Images) : 0.67 s
     FGSM Time (Per Image) : 67.07 us

     DeepFool Test Error : 93.76%
     DeepFool Robustness : 2.12e-02
     DeepFool Time (All Images) : 185.12 s
     DeepFool Time (Per Image) : 18.51 ms
```

Модель Network-In-Network на основе проведенных оценок не проявляет высокую стойкость к FGSM и DeepFool атакам. Высокий процент ошибок после атак свидетельствует о существенном влиянии этих атак на производительность модели.

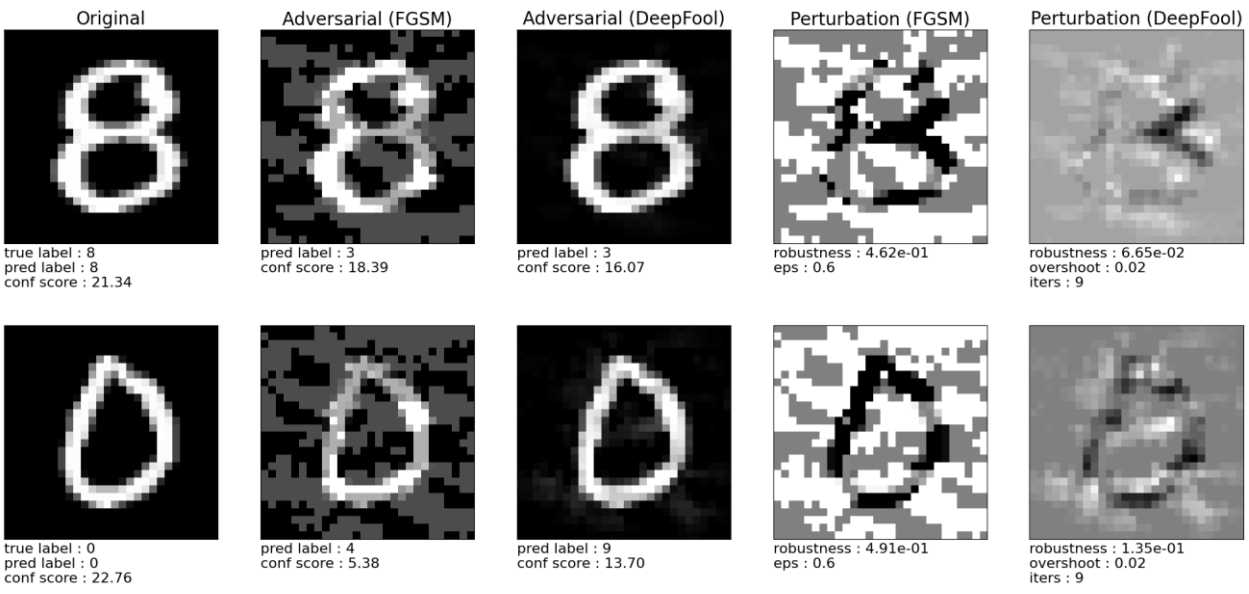
Загружаем модель LeNet и оцениваем ее стойкость к FGSM и DeepFool атакам.

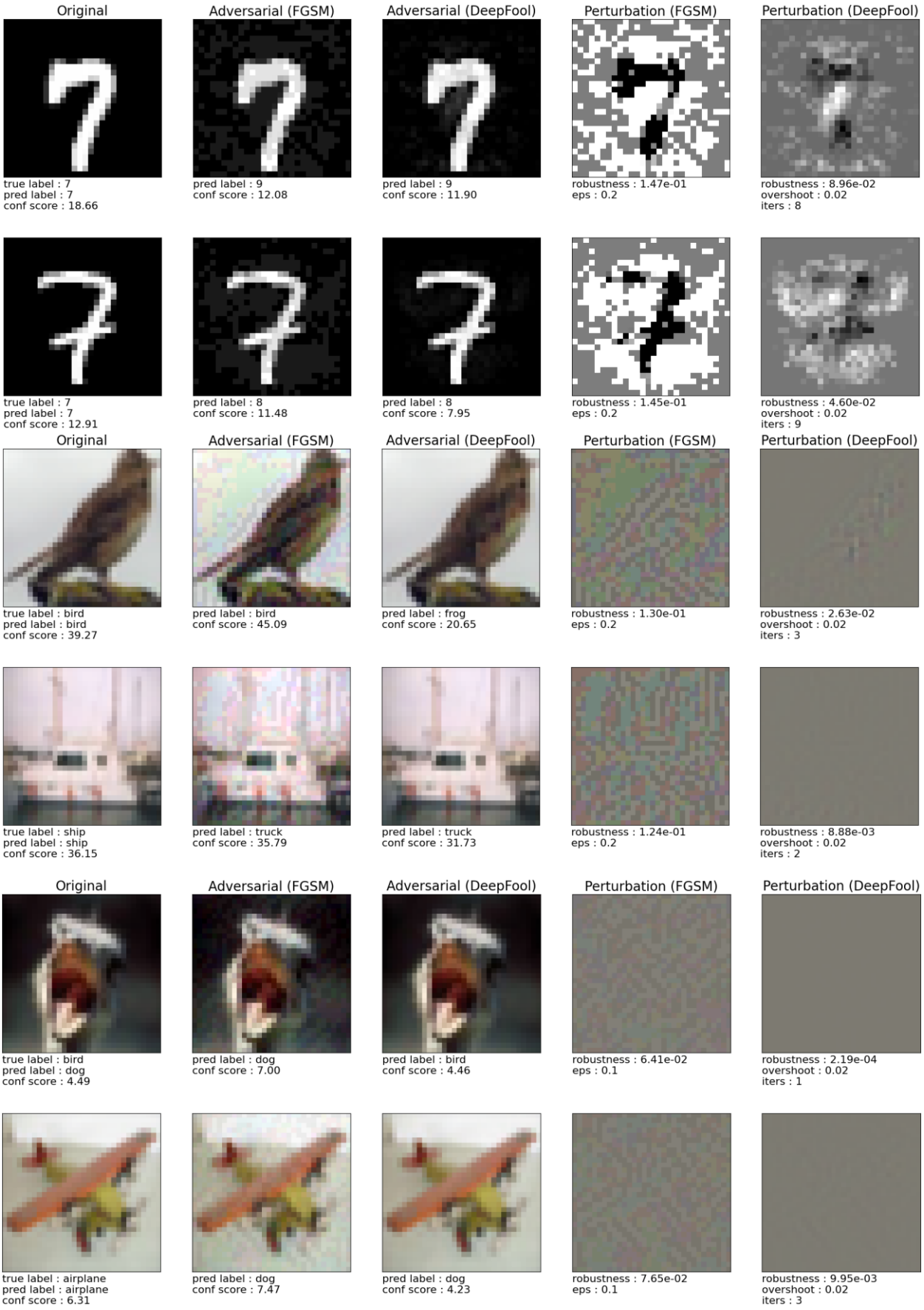
```
# Загрузка и оценка стойкости модели LeNet к FGSM и DeepFool атакам
fgsm_eps = 0.1

model = LeNet_CIFAR().to(device)
model.load_state_dict(torch.load('weights/clean/cifar_lenet.pth', map_location=device))

evaluate_attack('cifar_lenet_fgsm.csv', 'results', device, model, cifar_loader_test, cifar_min, cifar_max, fgsm_eps, is_fgsm=True)
print('')

deep_args = [64, 10, 0.02, 100]
evaluate_attack('cifar_lenet_deepfool.csv', 'results', device, model, cifar_loader_test, cifar_min, cifar_max, deep_args, is_fgsm=False)

if device.type == 'cuda': torch.cuda.empty_cache()

FGSM Test Error : 91.71%
FGSM Robustness : 8.90e-02
FGSM Time (All Images) : 0.40 s
FGSM Time (Per Image) : 40.08 us

DeepFool Test Error : 87.81%
DeepFool Robustness : 1.78e-02
DeepFool Time (All Images) : 73.27 s
DeepFool Time (Per Image) : 7.33 ms
```

Модель LeNet на основе проведенных оценок также не проявляет высокую стойкость к FGSM и DeepFool атакам. Оба вида атак оказывают существенное воздействие на производительность модели.

Проведем атаки с использованием FGSM и DeepFool на нескольких моделях, работающих с наборами данных MNIST и CIFAR-10.

```
# LeNet на датасете MNIST
fgsm_eps = 0.6
model = LeNet_MNIST().to(device)
model.load_state_dict(torch.load('weights/clean/mnist_lenet.pth'))
display_attack(device, model, mnist_test, mnist_tf_inv, mnist_min, mnist_max, fgsm_eps, deep_args,
               has_labels=False, l2_norm=True, pert_scale=1.0, fig_rows=2, fig_width=25, fig_height=11)

if device.type == 'cuda': torch.cuda.empty_cache()

# FCNet на датасете MNIST
fgsm_eps = 0.2
model = FC_500_150().to(device)
model.load_state_dict(torch.load('weights/clean/mnist_fc.pth'))
display_attack(device, model, mnist_test, mnist_tf_inv, mnist_min, mnist_max, fgsm_eps, deep_args,
               has_labels=False, l2_norm=True, pert_scale=1.0, fig_rows=2, fig_width=25, fig_height=11)

if device.type == 'cuda': torch.cuda.empty_cache()

# Network-in-Network на датасете CIFAR-10
fgsm_eps = 0.2
model = Net().to(device)
model.load_state_dict(torch.load('weights/clean/cifar_nin.pth'))
display_attack(device, model, cifar_test, cifar_tf_inv, cifar_min, cifar_max, fgsm_eps, deep_args,
               has_labels=False, l2_norm=True, pert_scale=1.0, fig_rows=2, fig_width=25, fig_height=11,
               label_map=cifar_classes)

if device.type == 'cuda': torch.cuda.empty_cache()

# LeNet на датасете CIFAR-10
fgsm_eps = 0.1
model = LeNet_CIFAR().to(device)
model.load_state_dict(torch.load('weights/clean/cifar_lenet.pth'))
display_attack(device, model, cifar_test, cifar_tf_inv, cifar_min, cifar_max, fgsm_eps, deep_args,
               has_labels=False, l2_norm=True, pert_scale=1.0, fig_rows=2, fig_width=25, fig_height=11,
               label_map=cifar_classes)


if device.type == 'cuda': torch.cuda.empty_cache()
```




Original
Adversarial (FGSM)
Adversarial (DeepFool)
Perturbation (FGSM)
Perturbation (DeepFool)
true label : 7
pred label : 7
conf score : 18.66
pred label : 9
conf score : 12.08
pred label : 9
conf score : 11.90
robustness : 1.47e-01
eps : 0.2
robustness : 8.96e-02
overshoot : 0.02
iters : 8
true label : 7
pred label : 7
conf score : 12.91
pred label : 8
conf score : 11.48
pred label : 8
conf score : 7.95
robustness : 1.45e-01
eps : 0.2
robustness : 4.60e-02
overshoot : 0.02
iters : 9
Original
Adversarial (FGSM)
Adversarial (DeepFool)
Perturbation (FGSM)
Perturbation (DeepFool)
true label : bird
pred label : bird
conf score : 39.27
pred label : bird
conf score : 45.09
pred label : frog
conf score : 20.65
robustness : 1.30e-01
eps : 0.2
robustness : 2.63e-02
overshoot : 0.02
iters : 3
true label : ship
pred label : ship
conf score : 36.15
pred label : truck
conf score : 35.79
pred label : truck
conf score : 31.73
robustness : 1.24e-01
eps : 0.2
robustness : 8.88e-03
overshoot : 0.02
iters : 2
Original
Adversarial (FGSM)
Adversarial (DeepFool)
Perturbation (FGSM)
Perturbation (DeepFool)
true label : bird
pred label : dog
conf score : 4.49
pred label : dog
conf score : 7.00
pred label : bird
conf score : 4.46
robustness : 6.41e-02
eps : 0.1
robustness : 2.19e-04
overshoot : 0.02
iters : 1
true label : airplane
pred label : airplane
conf score : 6.31
pred label : dog
conf score : 7.47
pred label : dog
conf score : 4.23
robustness : 7.65e-02
eps : 0.1
robustness : 9.95e-03
overshoot : 0.02
iters : 3

Далее проводим атаку FGSM с различными значениями eps на двух моделях и датасетах. Так мы можем оценить совйкость обоих моделей к атаке при различных значениях eps.

```
Evaluating FGSM Attack with eps=0.001...
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560:
UserWarning: This DataLoader will create 4 worker processes in total. Our
suggested max number of worker in current system is 2, which is smaller than what
this DataLoader is going to create. Please be aware that excessive worker creation
might get DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 3.07%
FGSM Robustness : 8.08e-04
FGSM Time (All Images) : 0.76 s
FGSM Time (Per Image) : 76.04 us
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560:
UserWarning: This DataLoader will create 4 worker processes in total. Our
suggested max number of worker in current system is 2, which is smaller than what
this DataLoader is going to create. Please be aware that excessive worker creation
might get DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 10.12%
FGSM Robustness : 8.92e-04
FGSM Time (All Images) : 1.15 s
FGSM Time (Per Image) : 115.46 us
Evaluating FGSM Attack with eps=0.02...
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560:
UserWarning: This DataLoader will create 4 worker processes in total. Our
suggested max number of worker in current system is 2, which is smaller than what
this DataLoader is going to create. Please be aware that excessive worker creation
might get DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 5.54%
FGSM Robustness : 1.60e-02
FGSM Time (All Images) : 0.45 s
FGSM Time (Per Image) : 44.66 us
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560:
UserWarning: This DataLoader will create 4 worker processes in total. Our
```

```
suggested max number of worker in current system is 2, which is smaller than what this DataLoader is going to create. Please be aware that excessive worker creation might get DataLoader running slow or even freeze, lower the worker number to avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 30.76%
FGSM Robustness : 1.78e-02
FGSM Time (All Images) : 1.38 s
FGSM Time (Per Image) : 138.22 us
Evaluating FGSM Attack with eps=0.5...
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560:
UserWarning: This DataLoader will create 4 worker processes in total. Our
suggested max number of worker in current system is 2, which is smaller than what
this DataLoader is going to create. Please be aware that excessive worker creation
might get DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 99.21%
FGSM Robustness : 3.86e-01
FGSM Time (All Images) : 0.55 s
FGSM Time (Per Image) : 54.78 us
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560:
UserWarning: This DataLoader will create 4 worker processes in total. Our
suggested max number of worker in current system is 2, which is smaller than what
this DataLoader is going to create. Please be aware that excessive worker creation
might get DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 82.67%
FGSM Robustness : 4.40e-01
FGSM Time (All Images) : 1.14 s
FGSM Time (Per Image) : 114.04 us
Evaluating FGSM Attack with eps=0.9...
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560:
UserWarning: This DataLoader will create 4 worker processes in total. Our
suggested max number of worker in current system is 2, which is smaller than what
this DataLoader is going to create. Please be aware that excessive worker creation
might get DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
```

```
FGSM Test Error : 99.87%
FGSM Robustness : 6.86e-01
FGSM Time (All Images) : 0.53 s
FGSM Time (Per Image) : 53.29 us
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560:
UserWarning: This DataLoader will create 4 worker processes in total. Our
suggested max number of worker in current system is 2, which is smaller than what
this DataLoader is going to create. Please be aware that excessive worker creation
might get DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 84.62%
FGSM Robustness : 7.79e-01
FGSM Time (All Images) : 1.07 s
FGSM Time (Per Image) : 106.74 us
Evaluating FGSM Attack with eps=10...
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560:
UserWarning: This DataLoader will create 4 worker processes in total. Our
suggested max number of worker in current system is 2, which is smaller than what
this DataLoader is going to create. Please be aware that excessive worker creation
might get DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 99.87%
FGSM Robustness : 1.47e+00
FGSM Time (All Images) : 0.53 s
FGSM Time (Per Image) : 52.56 us
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:560:
UserWarning: This DataLoader will create 4 worker processes in total. Our
suggested max number of worker in current system is 2, which is smaller than what
this DataLoader is going to create. Please be aware that excessive worker creation
might get DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
FGSM Batches Complete : (157 / 157)
FGSM Test Error : 87.50%
FGSM Robustness : 2.46e+00
FGSM Time (All Images) : 1.11 s
FGSM Time (Per Image) : 110.72 us
```

При малых значениях eps (например, 0.001 и 0.02) модель показывает низкую ошибку после атаки, но при этом и низкую стойкость к FGSM атаке.

С увеличением eps процент ошибок растет, а стойкость увеличивается. Однако, при очень больших значениях eps (10), процент ошибок также растет, что может говорить о насыщении атаки.