

|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт кибербезопасности и цифровых технологий Кафедра КБ-4 «Интеллектуальные системы информационной  безопасности» |

# Отчёт по лабораторной работе №1

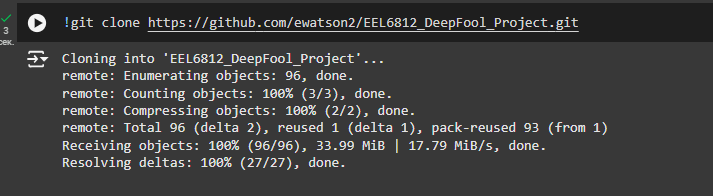
**по дисциплине «Анализ защищенности систем искусственного**

# интеллекта»

Выполнил: Мелкумян А.А

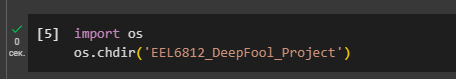
Группа: ББМО-02-23

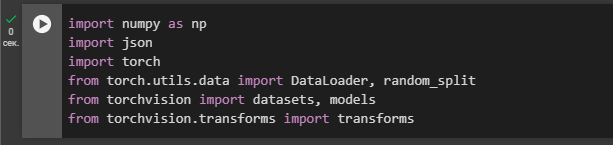
Москва - 2024

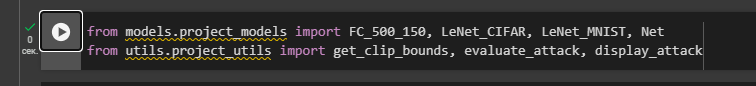


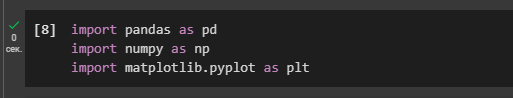
Копирую гитхаб

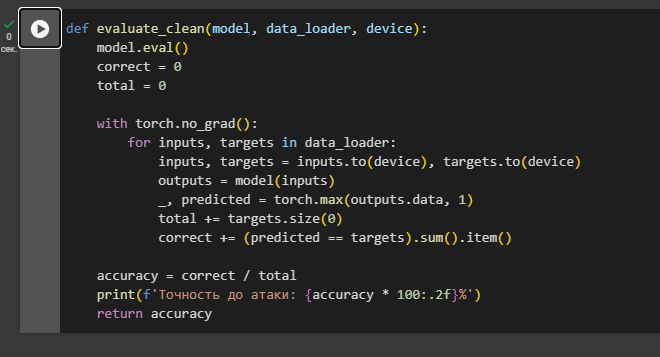
**Смена директории и импорт библиотек**







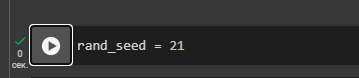


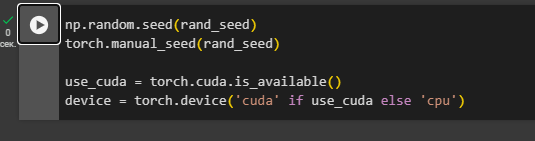


**Функция evaluate\_clean(model, data\_loader, device)**

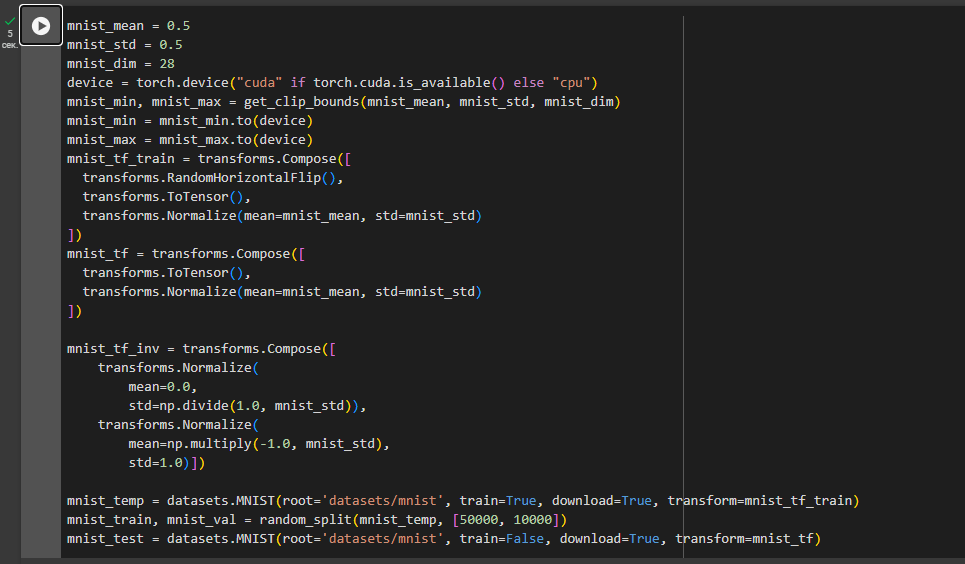
Эта функция оценивает точность модели на чистых (не атакованных) данных, используя заданный загрузчик данных (data loader).

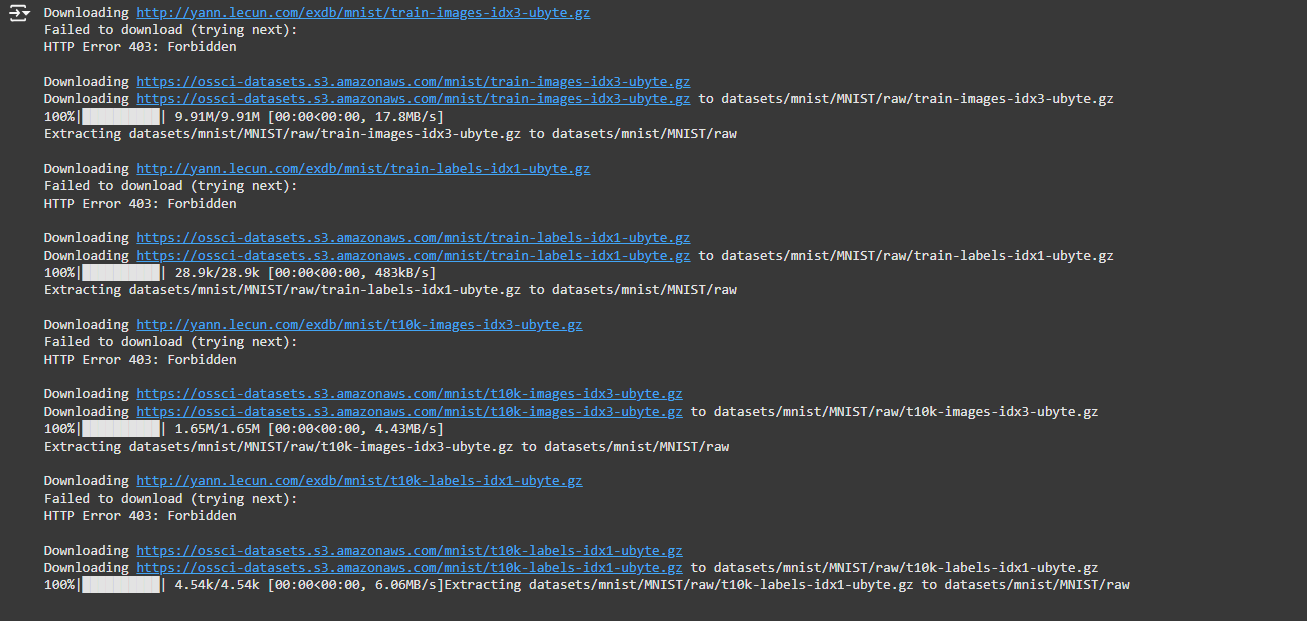
**Установка случайного значения - номер в списке группы "21"(я в журнале)**



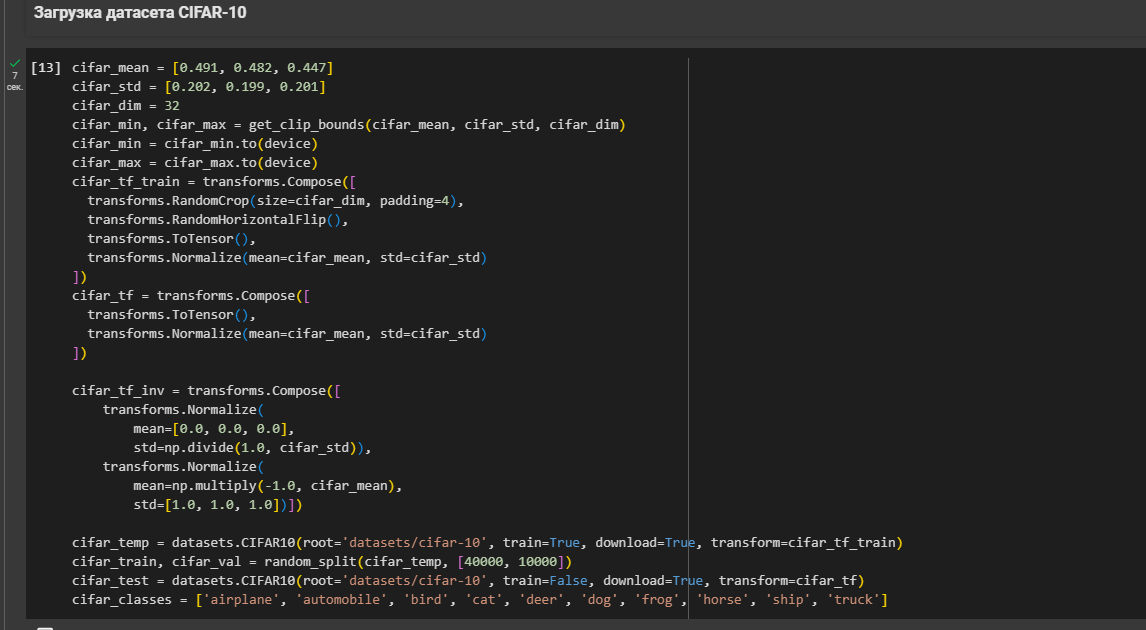


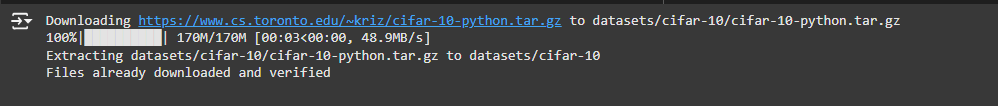
**Загрузка датасета MNIST**



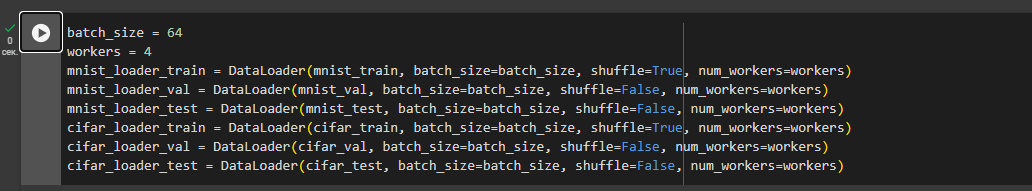


**Загрузка датасета CIFAR-10**





**Настройка DataLoader**



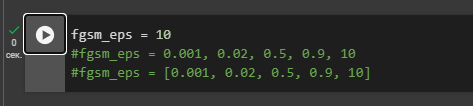
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: UserWarning: This DataLoader will create 4 worker processes in total. Our suggested max number of worker in current system is 2, which is smaller than what this DataLoader is going to create. Please be aware that excessive worker creation might get DataLoader running slow or even freeze, lower the worker number to avoid potential slowness/freeze if necessary.



warnings.warn(

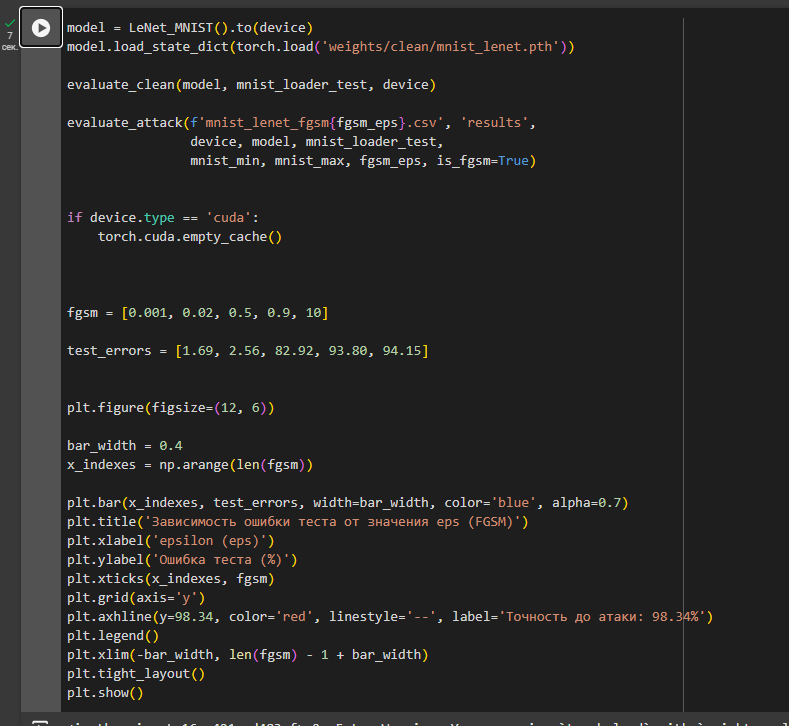
инструмент в PyTorch для пакетной загрузки данных из набора данных Он автоматически разделяет данные на пакеты и позволяет эффективно загружать данные для обучения или тестирования.

**FGSM Атака**

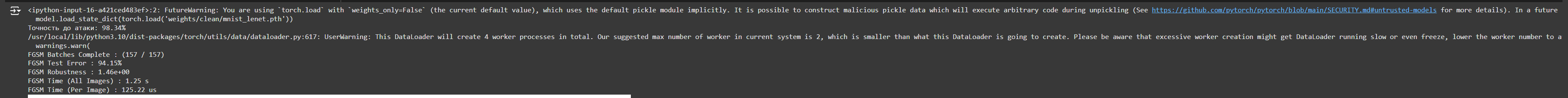


это метод атаки, используемый для создания примеров, которые могут сбить с толку обученную модель машинного обучения. В рамках FGSM мы добавляем небольшие, но целенаправленные изменения (в зависимости от **eps**) к изображению, чтобы изменить его так, чтобы модель неправильно классифицировала его

**Стойкость к атакам модели LeNet на датасете MNIST**



Этот код выполняет несколько ключевых операций, связанных с обученной моделью **LeNet** для классификации набора данных **MNIST**. В нем также выполняется атака FGSM с различными значениями параметра **eps**, а затем строится график зависимости ошибки от этого параметра



<ipython-input-16-a421ced483ef>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/mnist\_lenet.pth'))

Точность до атаки: 98.34%

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: 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(

FGSM Batches Complete : (157 / 157)

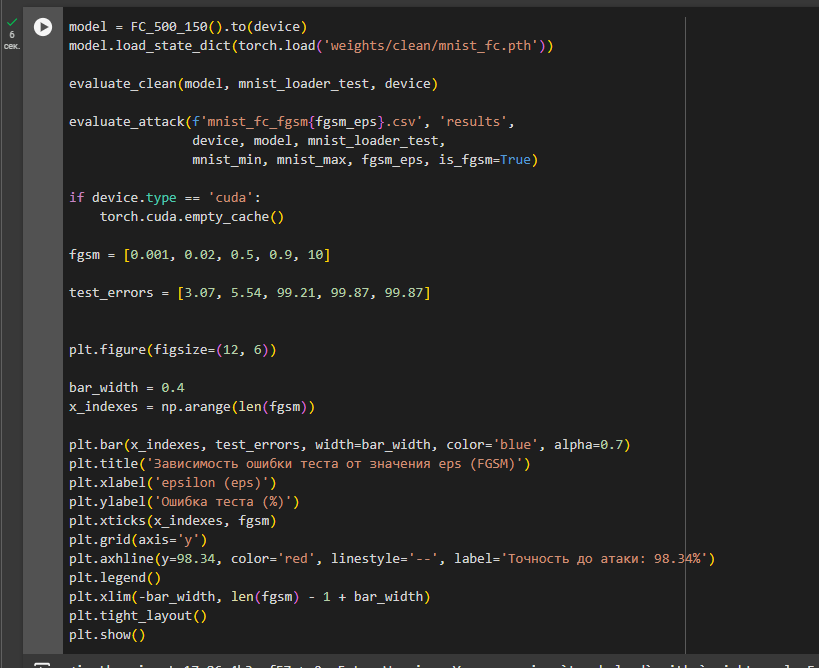
FGSM Test Error : 94.15%

FGSM Robustness : 1.46e+00

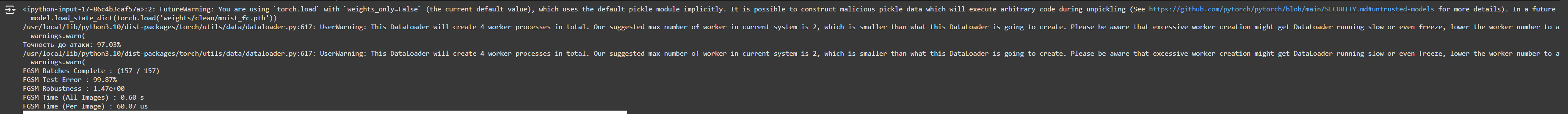
FGSM Time (All Images) : 1.25 s

FGSM Time (Per Image) : 125.22 us

**Стойкость к атакам модели FC на датасете MNIST**



Этот код выполняет аналогичные операции с моделью **FC\_500\_150**, но для другого типа нейронной сети (возможно, полносвязной сети с 500 и 150 нейронами на слоях).



<ipython-input-17-86c4b3caf57a>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/mnist\_fc.pth'))

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: 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(

Точность до атаки: 97.03%

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: 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(

FGSM Batches Complete : (157 / 157)

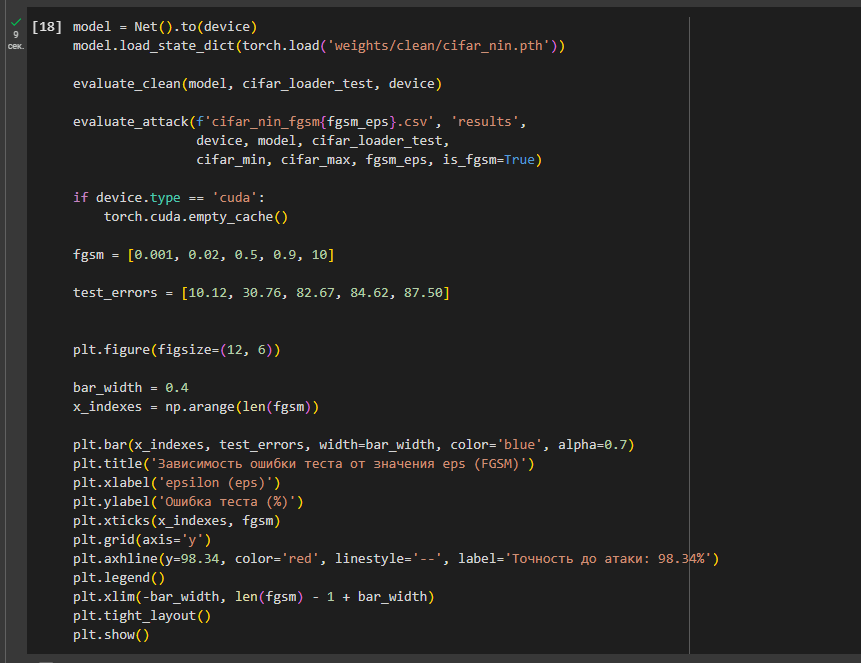
FGSM Test Error : 99.87%

FGSM Robustness : 1.47e+00

FGSM Time (All Images) : 0.60 s

FGSM Time (Per Image) : 60.07 us

**Стойкость к атакам модели Network-In-Network на датасете CIFAR-10**



представляет собой сеть **Network-in-Network (NiN)** для классификации изображений из набора данных **CIFAR-10**

Код оценивает модель сначала на тестовых данных без атак, а затем применяет атаку **FGSM** с различными значениями **eps**, измеряя влияние атаки на точность модели.

<ipython-input-18-412f7b94b9eb>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/cifar\_nin.pth'))

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: 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(

Точность до атаки: 90.72%

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: 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(

FGSM Batches Complete : (157 / 157)

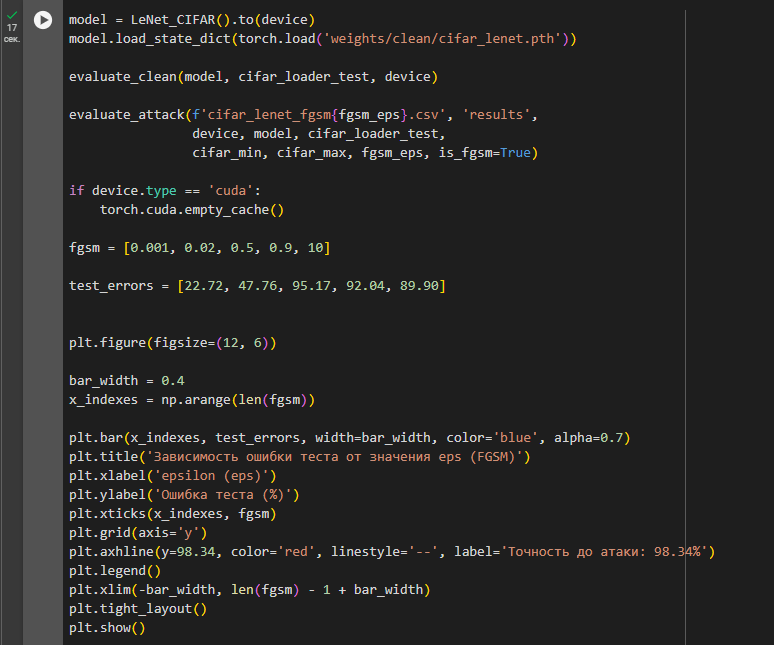
FGSM Test Error : 87.50%

FGSM Robustness : 2.46e+00

FGSM Time (All Images) : 1.49 s

FGSM Time (Per Image) : 148.83 us

**Стойкость к атакам модели LeNet на датасете CIFAR-10**



теперь для модели **LeNet** на наборе данных **CIFAR-10**

Сначала оценивается точность модели на тестовом наборе данных **CIFAR-10** без применения атак. Затем применяется атака **FGSM** с различными значениями **epsilon (eps)**, и результаты (ошибки модели) сохраняются в CSV-файл

<ipython-input-19-fe2d187d7de7>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/cifar\_lenet.pth'))

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: 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(

Точность до атаки: 78.66%

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: 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(

FGSM Batches Complete : (157 / 157)

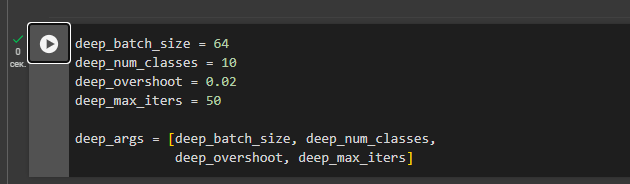
FGSM Test Error : 89.90%

FGSM Robustness : 2.47e+00

FGSM Time (All Images) : 1.45 s

FGSM Time (Per Image) : 144.77 us

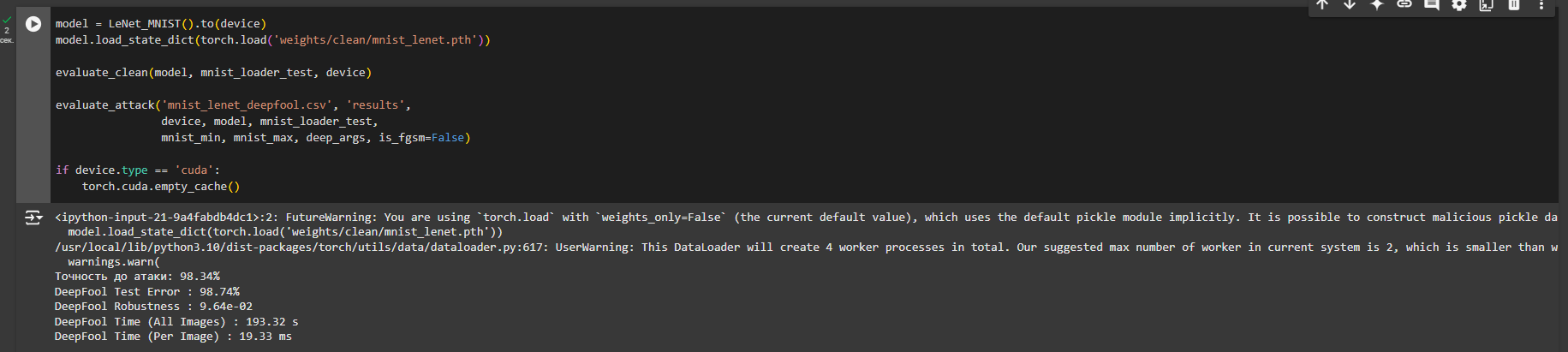
**DeepFool атака**



Этот код определяет несколько гиперпараметров и создает список **deep\_args**, содержащий их, который может быть использован позже в модели или в процессе обучения/оценки.

**DeepFool** — это метод, предназначенный для вычисления минимальных изменений (perturbations), которые могут привести к неправильной классификации изображения. В отличие от **FGSM**, который является относительно простой атакой с фиксированным размером шага, **DeepFool** использует более сложный и адаптивный подход.

**Стойкость к атакам модели LeNet на датасете MNIST**



оценки модели **LeNet** на наборе данных **MNIST** с применением атаки **DeepFool**.

Сначала модель **LeNet\_MNIST** загружается и оценивается на тестовом наборе данных MNIST без атаки.Затем применяется атака **DeepFool** с заданными параметрами, чтобы проверить, как модель справляется с малозаметными изменениями, приводящими к ошибкам в классификации.Результаты атаки сохраняются в файл, и в конце освобождается память, если используется **GPU**.

<ipython-input-21-9a4fabdb4dc1>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/mnist\_lenet.pth'))

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: 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(

Точность до атаки: 98.34%

DeepFool Test Error : 98.74%

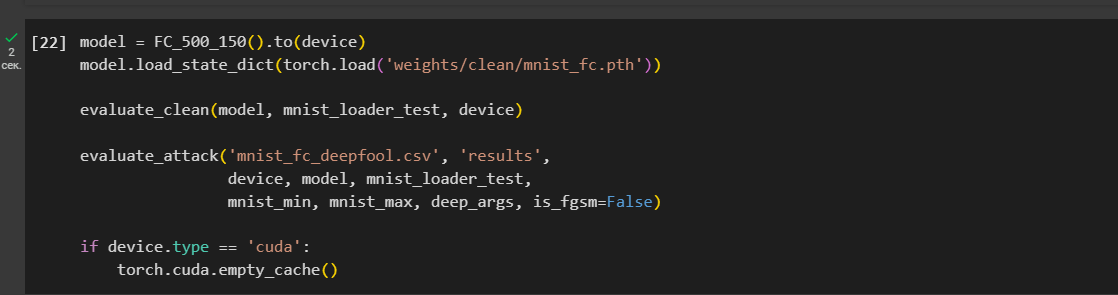
DeepFool Robustness : 9.64e-02

DeepFool Time (All Images) : 193.32 s

DeepFool Time (Per Image) : 19.33 ms

**Стойкость к атакам модели FC на датасете MNIST**

**FC\_500\_150** а также применяется атака **DeepFool**.

1.   
   **Модель** загружается и оценивается на чистых данных для получения точности.
2. **Атака DeepFool** применяется к модели, чтобы оценить, как она справляется с малозаметными изменениями на изображениях, предназначенных для введения модели в ошибку.
3. Результаты атаки сохраняются в **mnist\_fc\_deepfool.csv**, чтобы можно было проанализировать производительность модели после атаки.

После завершения работы с **GPU** память очищается для предотвращения утечек

<ipython-input-22-f4287413aeee>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/mnist\_fc.pth'))

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: 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(

Точность до атаки: 97.03%

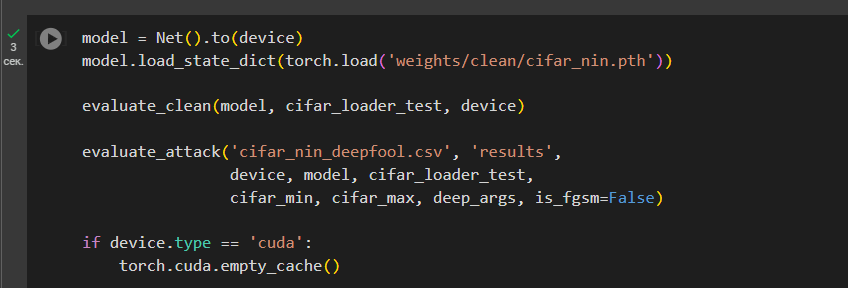
DeepFool Test Error : 97.92%

DeepFool Robustness : 6.78e-02

DeepFool Time (All Images) : 141.81 s

DeepFool Time (Per Image) : 14.18 ms

**Стойкость к атакам модели Network-In-Network на датасете CIFAR-10**



<ipython-input-23-d39c82e071ac>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/cifar\_nin.pth'))

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: 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(

Точность до атаки: 90.72%

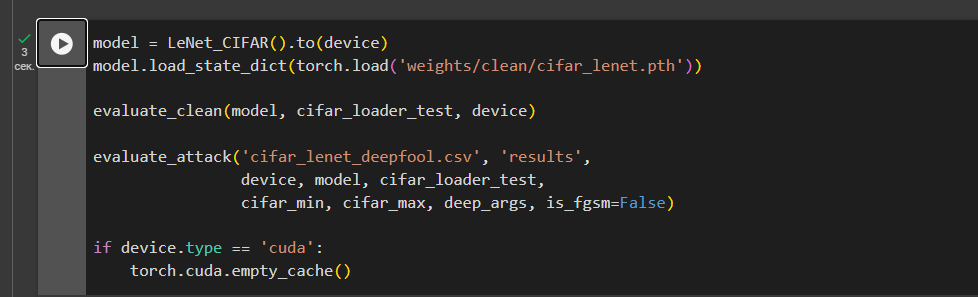
DeepFool Test Error : 93.76%

DeepFool Robustness : 2.12e-02

DeepFool Time (All Images) : 185.12 s

DeepFool Time (Per Image) : 18.51 ms

**Стойкость к атакам модели LeNet на датасете CIFAR-10**



1. **Модель** загружается и оценивается на чистых данных **CIFAR-10** для получения точности без атак.
2. На модель **LeNet\_CIFAR** применяется **атака DeepFool** для проверки устойчивости к малозаметным изменениям на изображениях, которые могут привести к неправильной классификации.
3. Результаты атаки сохраняются в **cifar\_lenet\_deepfool.csv** для анализа.
4. После выполнения работы с **GPU** очищается память для предотвращения утечек.

<ipython-input-24-71a3964ca979>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/cifar\_lenet.pth'))

/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: 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(

Точность до атаки: 78.66%

DeepFool Test Error : 87.81%

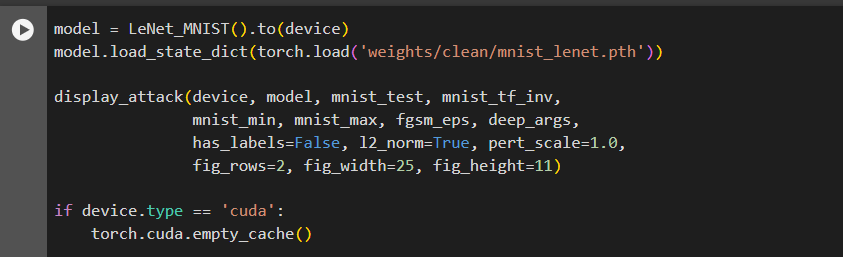
DeepFool Robustness : 1.78e-02

DeepFool Time (All Images) : 73.27 s

DeepFool Time (Per Image) : 7.33 ms

**Графическое представление атакованных примеров**

**LeNet MNIST**

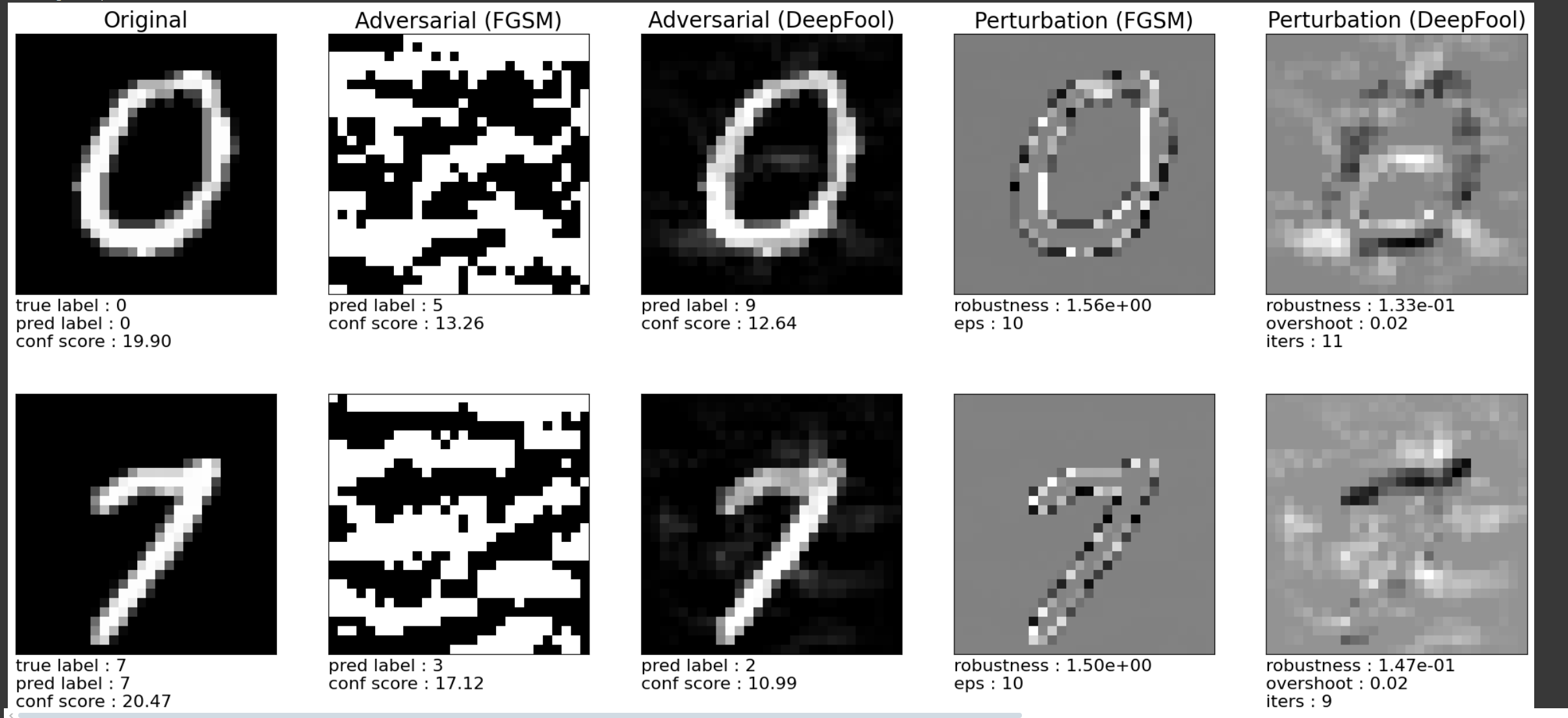


<ipython-input-25-59500bac25fc>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/mnist\_lenet.pth'))

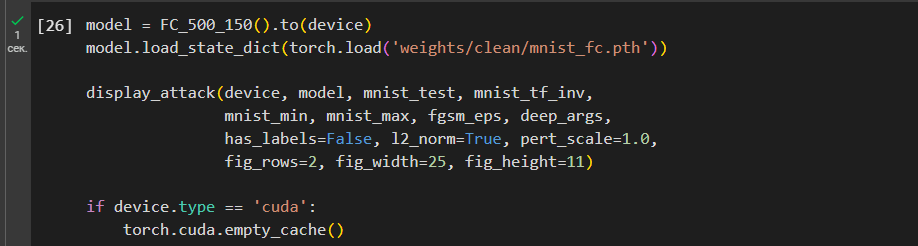
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:617: UserWarning: This DataLoader will create 4 worker processes in total. Our suggested max number of worker in current system is 2, which is smaller than what this DataLoader is going to create. Please be aware that excessive worker creation might get DataLoader running slow or even freeze, lower the worker number to avoid potential slowness/freeze if necessary.

warnings.warn(



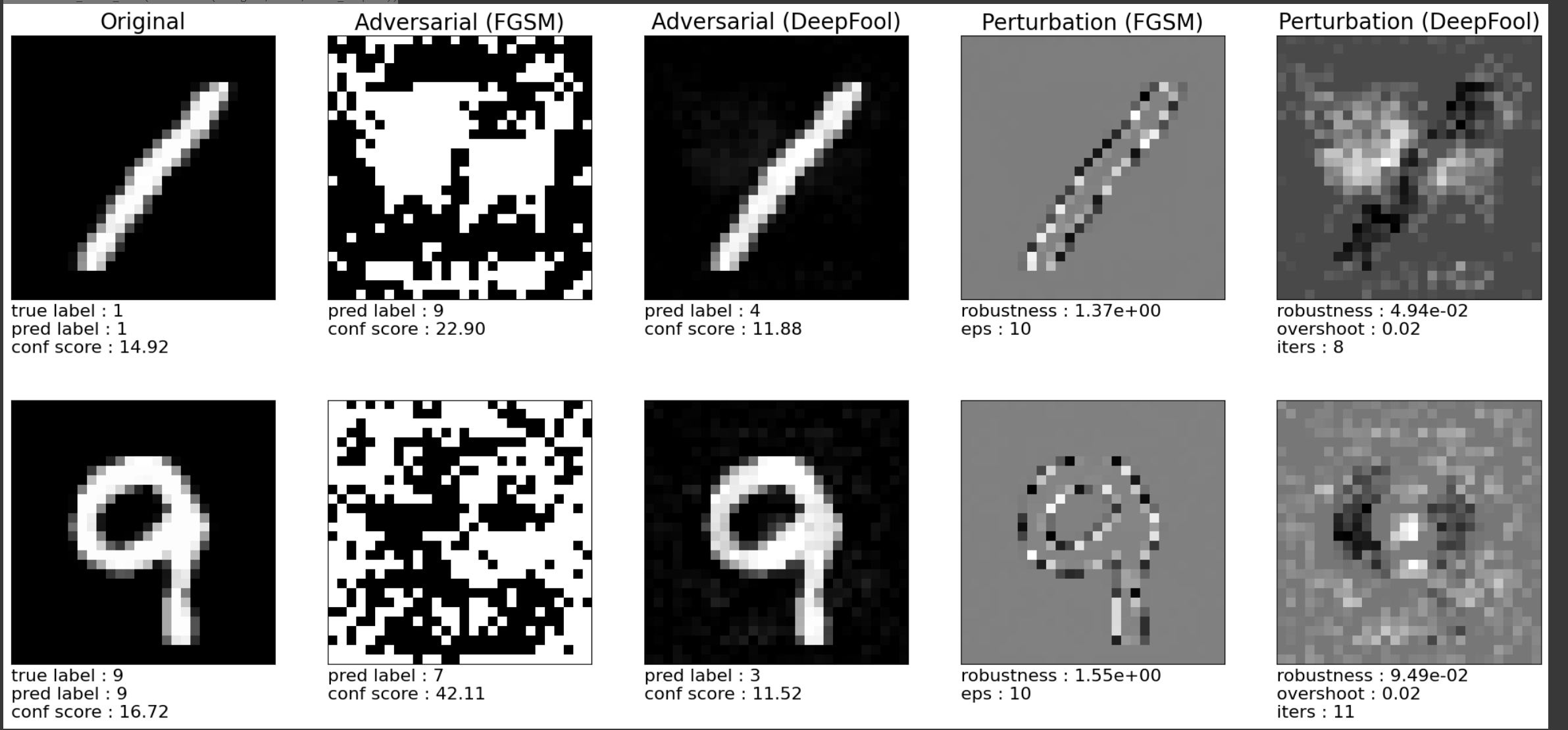
В приведенном коде используется модель **LeNet\_MNIST** для работы с набором данных **MNIST**, с применением атаки.

**FC MNIST**

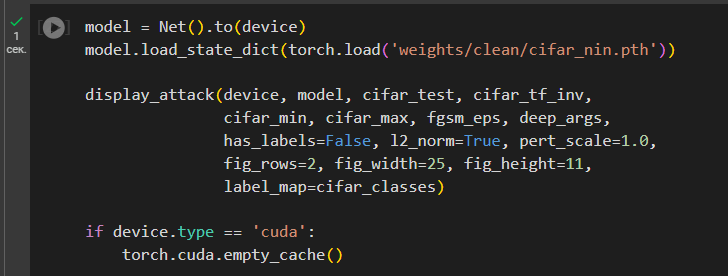


<ipython-input-26-7168114d0e38>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/mnist\_fc.pth'))

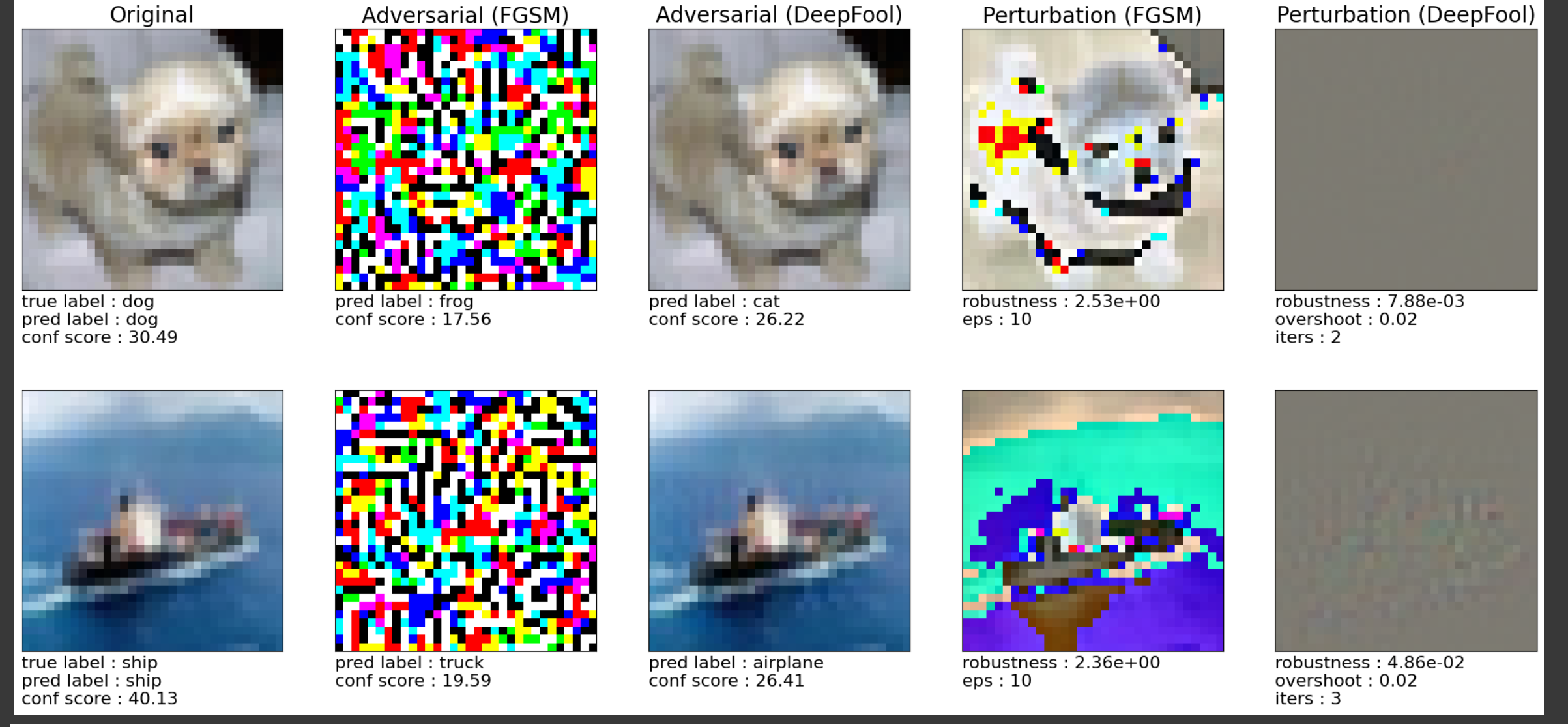


**Network-In-Network Model CIFAR-10**

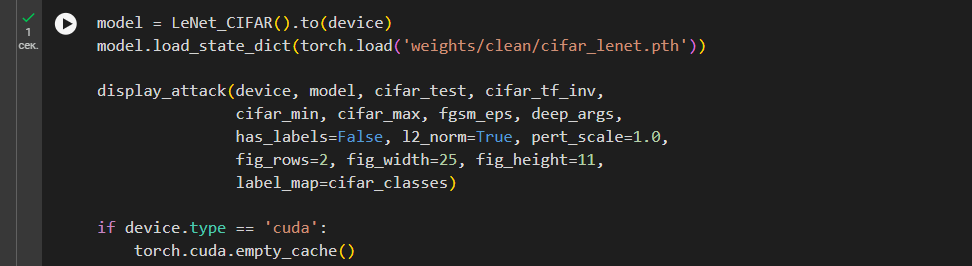


<ipython-input-27-e97f89c1935b>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/cifar\_nin.pth'))



**LeNet CIFAR-10**



<ipython-input-28-c2525624b6c5>:2: FutureWarning: You are using `torch.load` with `weights\_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See <https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models> for more details). In a future release, the default value for `weights\_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add\_safe\_globals`. We recommend you start setting `weights\_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.

model.load\_state\_dict(torch.load('weights/clean/cifar\_lenet.pth'))

