```
[8] import torch
    import torch.nn as nn
    import torch.optim as optim
    import torchvision
    import torchvision.transforms as transforms
    import matplotlib.pyplot as plt
    import numpy as np
```

```
# Трансформации для нормализации данных
transform = transforms.Compose([
    transforms.ToTensor(),
])

# Загрузка датасета MNIST
trainset = torchvision.datasets.MNIST(root='./data', train=True,
                                      download=True, transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=64,
                                          shuffle=True)

testset = torchvision.datasets.MNIST(root='./data', train=False,
                                     download=True, transform=transform)
testloader = torch.utils.data.DataLoader(testset, batch_size=1,
                                         shuffle=False)
```

```
Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz
Failed to download (trying next):
<urlopen error [Errno 111] Connection refused>

Downloading https://ossci-datasets.s3.amazonaws.com/mnist/train-images-idx3-ubyte.gz
Downloading https://ossci-datasets.s3.amazonaws.com/mnist/train-images-idx3-ubyte.gz to ./data/MNIST/raw/train-images-idx3-ubyte.gz
100%|██████████| 9.91M/9.91M [00:00<00:00, 17.9MB/s]
Extracting ./data/MNIST/raw/train-images-idx3-ubyte.gz to ./data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz
Failed to download (trying next):
<urlopen error [Errno 111] Connection refused>

Downloading https://ossci-datasets.s3.amazonaws.com/mnist/train-labels-idx1-ubyte.gz
Downloading https://ossci-datasets.s3.amazonaws.com/mnist/train-labels-idx1-ubyte.gz to ./data/MNIST/raw/train-labels-idx1-ubyte.gz
100%|██████████| 28.9k/28.9k [00:00<00:00, 501kB/s]
Extracting ./data/MNIST/raw/train-labels-idx1-ubyte.gz to ./data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz
Failed to download (trying next):
<urlopen error [Errno 111] Connection refused>

Downloading https://ossci-datasets.s3.amazonaws.com/mnist/t10k-images-idx3-ubyte.gz
Downloading https://ossci-datasets.s3.amazonaws.com/mnist/t10k-images-idx3-ubyte.gz to ./data/MNIST/raw/t10k-images-idx3-ubyte.gz
100%|██████████| 1.65M/1.65M [00:00<00:00, 4.55MB/s]
Extracting ./data/MNIST/raw/t10k-images-idx3-ubyte.gz to ./data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz
Failed to download (trying next):
<urlopen error [Errno 111] Connection refused>

Downloading https://ossci-datasets.s3.amazonaws.com/mnist/t10k-labels-idx1-ubyte.gz
Downloading https://ossci-datasets.s3.amazonaws.com/mnist/t10k-labels-idx1-ubyte.gz to ./data/MNIST/raw/t10k-labels-idx1-ubyte.gz
100%|██████████| 4.54k/4.54k [00:00<00:00, 2.31MB/s]
Extracting ./data/MNIST/raw/t10k-labels-idx1-ubyte.gz to ./data/MNIST/raw
```

```python
class SimpleFCNN(nn.Module):
    def __init__(self):
        super(SimpleFCNN, self).__init__()
        self.fc1 = nn.Linear(28*28, 256)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(256, 10)

    def forward(self, x):
        x = x.view(-1, 28*28)  # Развертывание изображения в вектор
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x

# Инициализация модели, функции потерь и оптимизатора
model = SimpleFCNN()
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)
```

```python
# Функция обучения
def train(model, device, trainloader, optimizer, criterion, epochs=5):
    model.train()
    for epoch in range(epochs):
        running_loss = 0.0
        for data, target in trainloader:
            data, target = data.to(device), target.to(device)

            optimizer.zero_grad()
            outputs = model(data)
            loss = criterion(outputs, target)
            loss.backward()
            optimizer.step()

            running_loss += loss.item()
        print(f'Epoch {epoch+1}, Loss: {running_loss/len(trainloader)}')
```

```python
# Проверка устройства (CPU или GPU)

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

# Обучение модели
train(model, device, trainloader, optimizer, criterion, epochs=5)
```

```
Epoch 1, Loss: 0.29758511738244026
Epoch 2, Loss: 0.12378769978853081
Epoch 3, Loss: 0.08300007835947978
Epoch 4, Loss: 0.06007767748633729
Epoch 5, Loss: 0.04502707581519326
```

```python
def fgsm_attack(data, epsilon, data_grad):
    # Получение знака градиента
    sign_data_grad = data_grad.sign()
    # Создание адверсариального примера путем добавления возмущения
    perturbed_data = data + epsilon * sign_data_grad
    # Клинговка значений в допустимом диапазоне
    perturbed_data = torch.clamp(perturbed_data, 0, 1)
    return perturbed_data
```

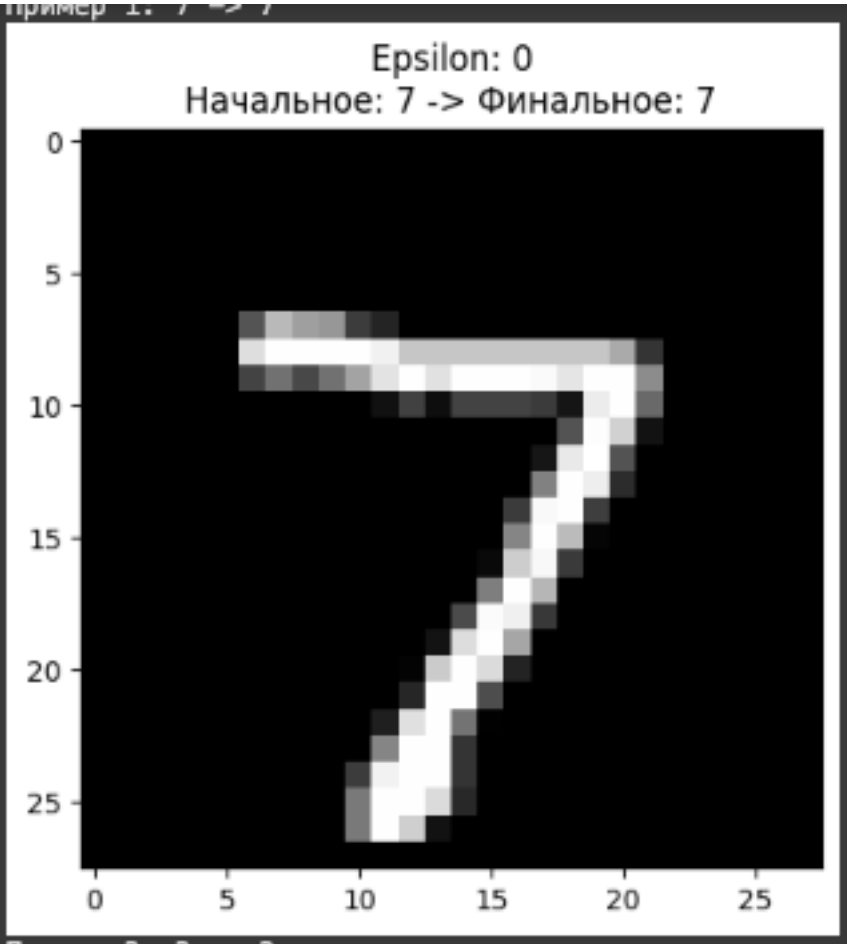
```python
def test(model, device, testloader, epsilon):
    # Установка модели в режим оценки
    model.eval()
    correct = 0
    adv_examples = []

    for data, target in testloader:
        data, target = data.to(device), target.to(device)

        # Включение градиентов для входных данных
        data.requires_grad = True

        # Прямой проход
        output = model(data)
        init_pred = output.max(1, keepdim=True)[1]

        # Если предсказание неверно, пропустить пример
        if init_pred.item() != target.item():
            continue

        # Вычисление функции потерь
        loss = criterion(output, target)

        # Обратный проход
        model.zero_grad()
        loss.backward()
        data_grad = data.grad.data

        # Создание адверсариального примера
        perturbed_data = fgsm_attack(data, epsilon, data_grad)

        # Повторный проход
        output = model(perturbed_data)
        final_pred = output.max(1, keepdim=True)[1]
```

```
        # Сравнение предсказаний
        if final_pred.item() == target.item():
            correct += 1
            # Сохранение некоторых примеров
            if len(adv_examples) < 5:
                adv_ex = perturbed_data.squeeze().detach().cpu().numpy()
                adv_examples.append((init_pred.item(), final_pred.item(), adv_ex))
        else:
            # Сохранение некоторых примеров
            if len(adv_examples) < 5:
                adv_ex = perturbed_data.squeeze().detach().cpu().numpy()
                adv_examples.append((init_pred.item(), final_pred.item(), adv_ex))

    # Подсчет точности
    final_acc = correct / len(testloader)
    print(f'Epsilon: {epsilon}\tTest Accuracy = {final_acc * 100:.2f}%')

    return final_acc, adv_examples

# Тестирование модели на различных значениях epsilon
epsilons = [0, 0.001, 0.02, 0.5, 0.9, 10]
accuracies = []
examples = []

for eps in epsilons:
    acc, ex = test(model, device, testloader, eps)
    accuracies.append(acc)
    examples.append(ex)
```

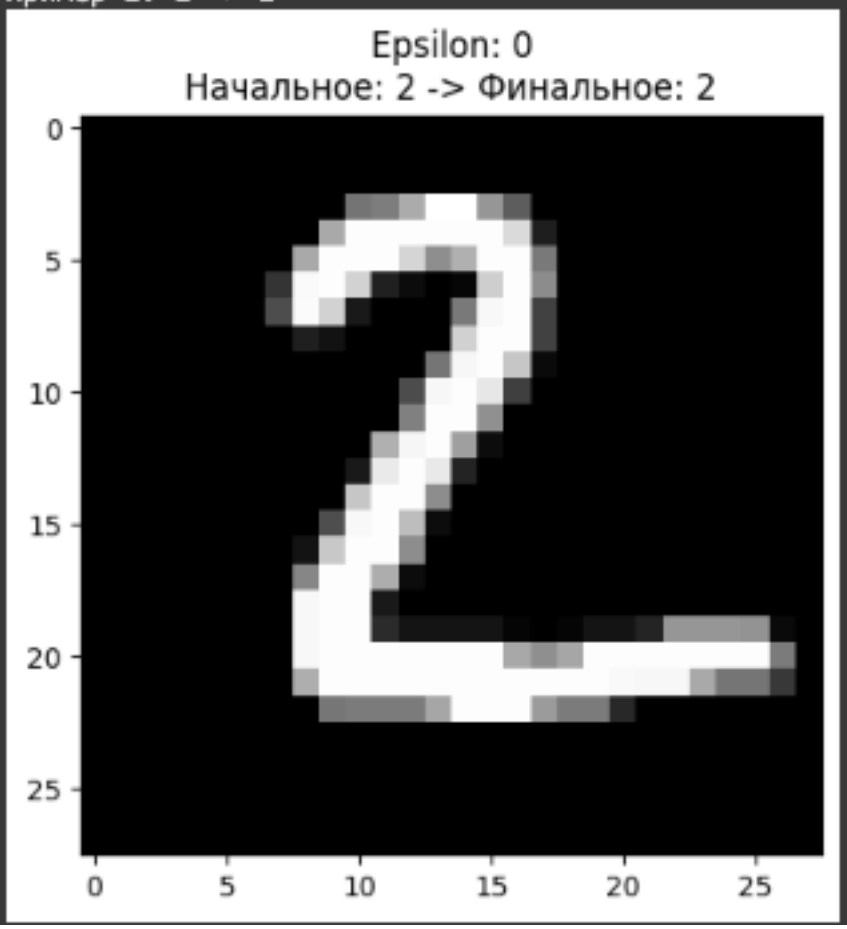
```
Epsilon: 0      Test Accuracy = 97.77%
Epsilon: 0.001  Test Accuracy = 97.62%
Epsilon: 0.02   Test Accuracy = 92.46%
Epsilon: 0.5    Test Accuracy = 0.02%
Epsilon: 0.9    Test Accuracy = 0.00%
Epsilon: 10     Test Accuracy = 0.00%
```

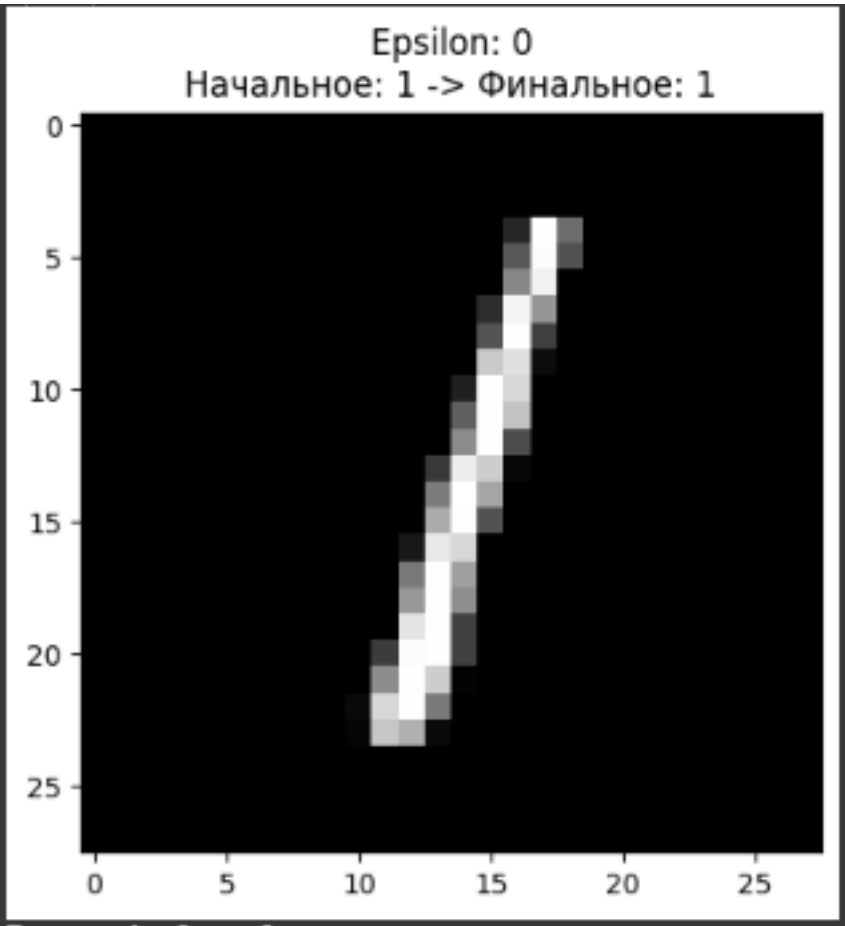
```
def imshow(img, title):
    npimg = img.numpy()
    plt.imshow(np.transpose(npimg, (1, 2, 0)), interpolation='nearest')
    plt.title(title)
    plt.show()

# Пример отображения адверсариальных примеров
for i, eps in enumerate(epsilons):
    print(f'Epsilon: {eps}\n')
    for j, (init_pred, final_pred, ex) in enumerate(examples[i]):
        print(f'Пример {j+1}: {init_pred} -> {final_pred}')
        plt.imshow(ex, cmap='gray')
        plt.title(f'Epsilon: {eps}\nНачальное: {init_pred} -> Финальное: {final_pred}')
        plt.show()
```

Пример 1: 7 -> 7

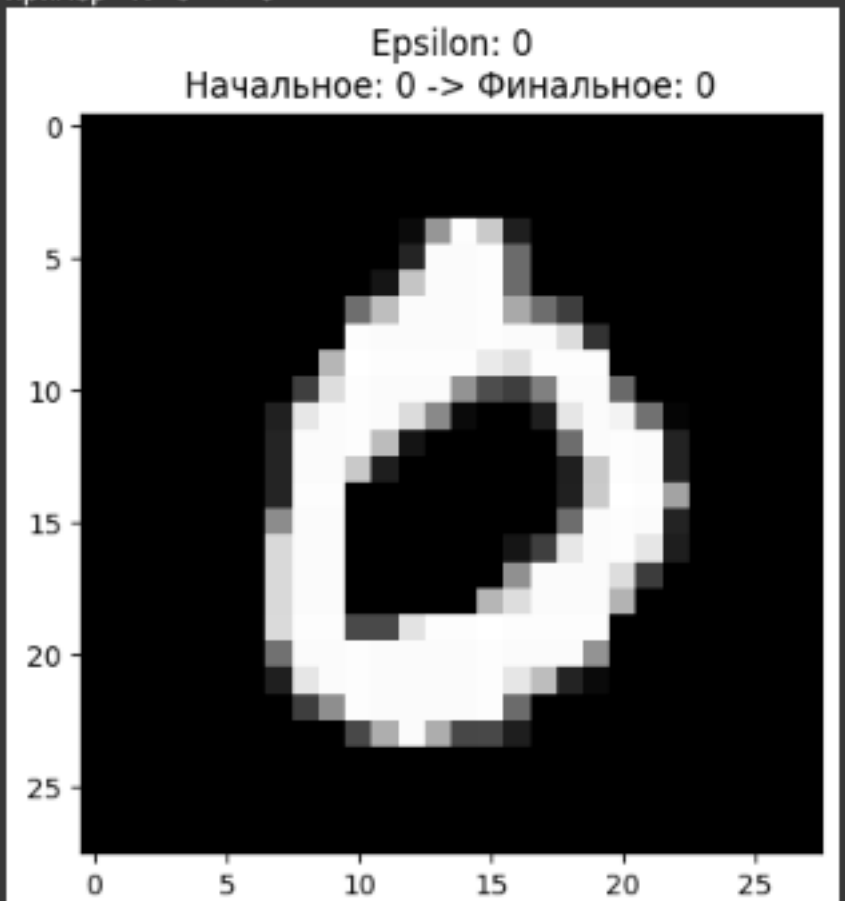


Пример 2: 2 -> 2

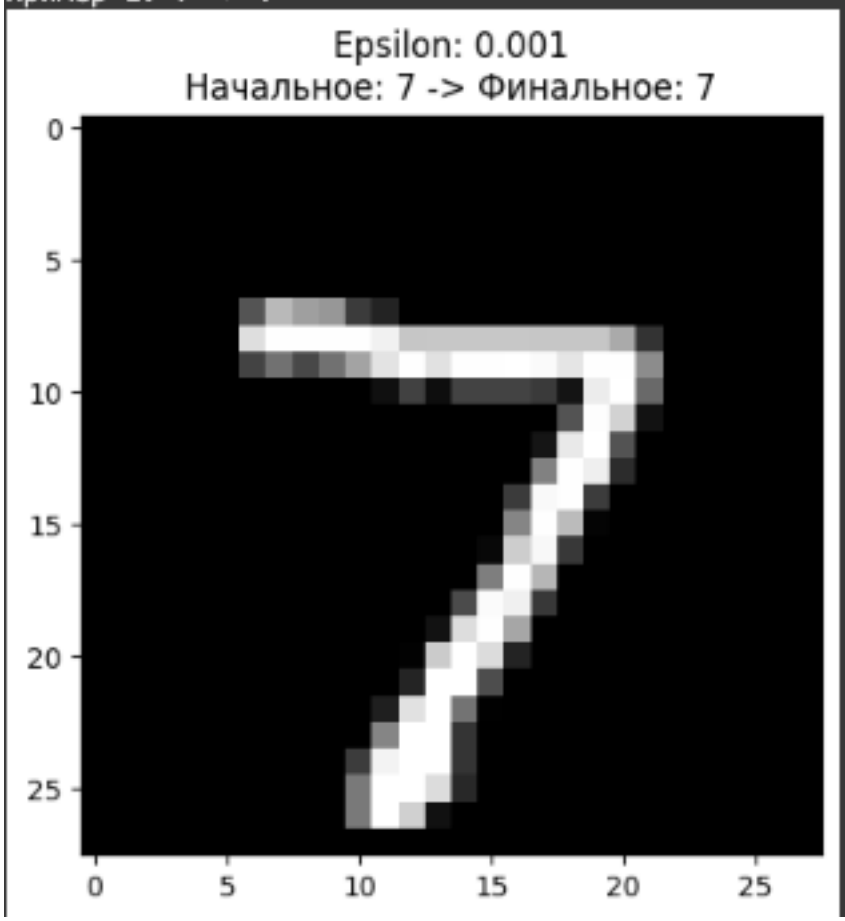


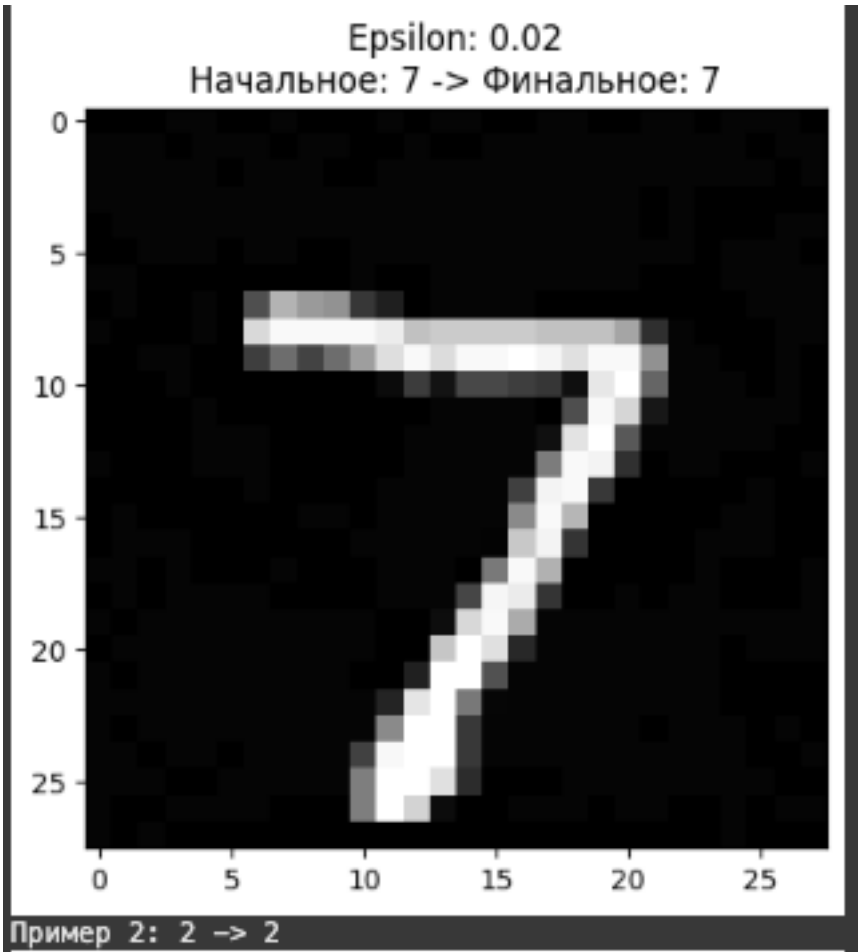

Epsilon: 0
Начальное: 1 -> Финальное: 1
0
5
10
15
20
25
0
5
10
15
20
25

```
Пример 4: 0 -> 0
```

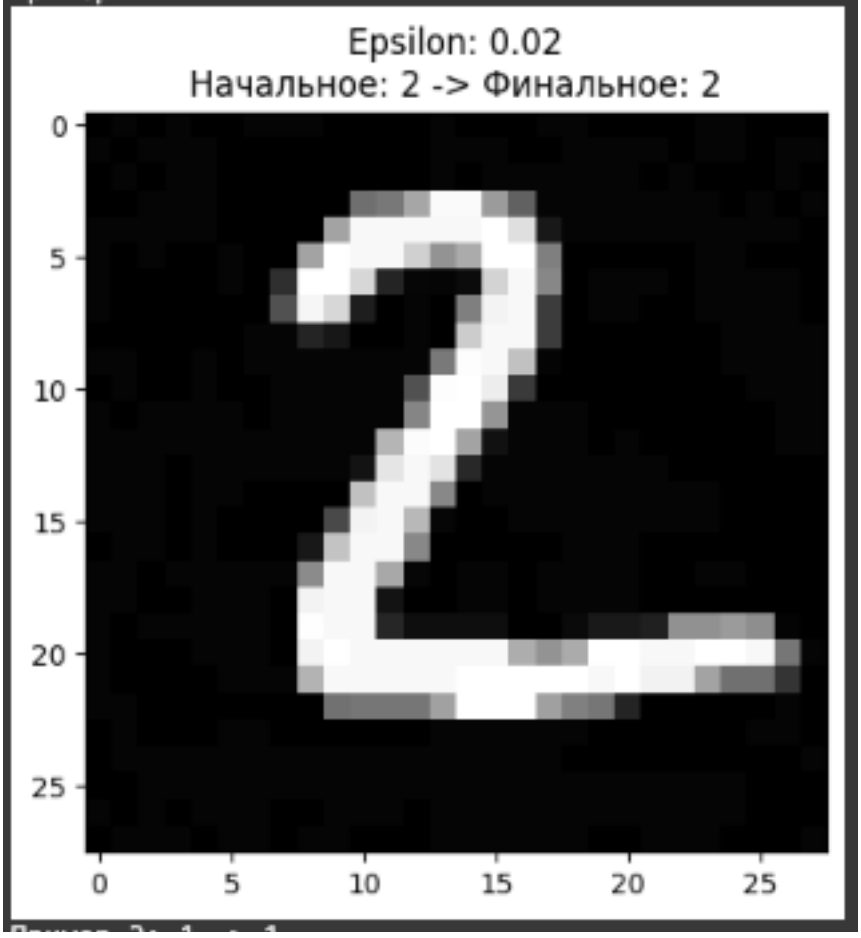

Epsilon: 0
Начальное: 0 -> Финальное: 0
0
5
10
15
20
25
0
5
10
15
20
25

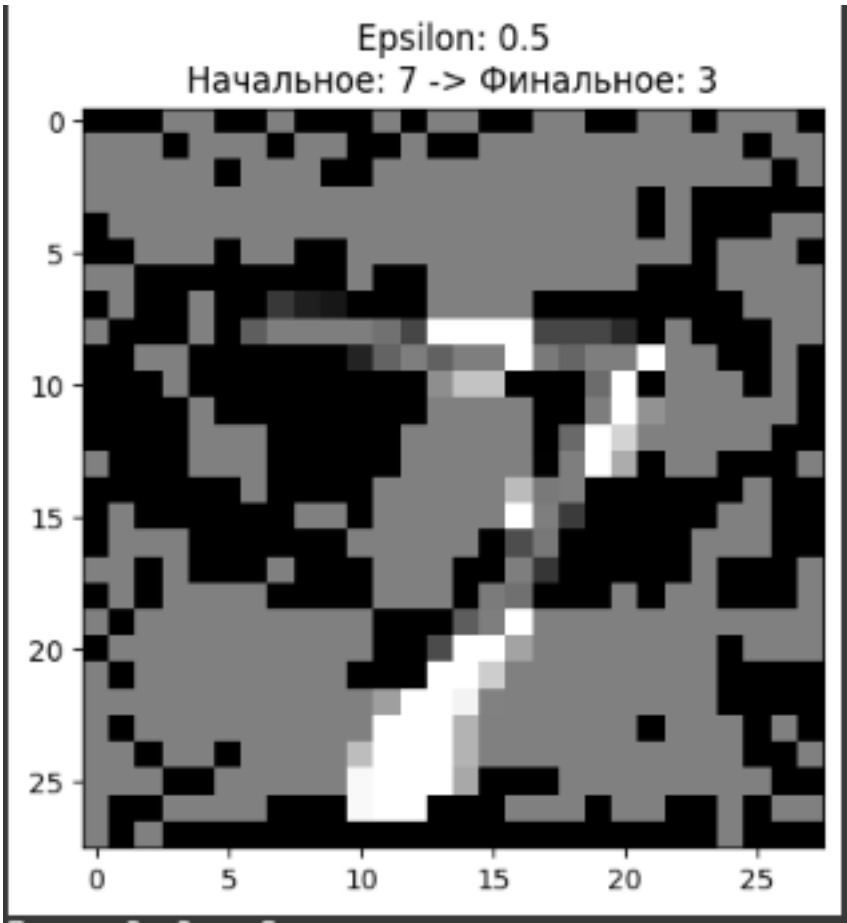

```
Epsilon: 0.001

Пример 1: 7 -> 7
```



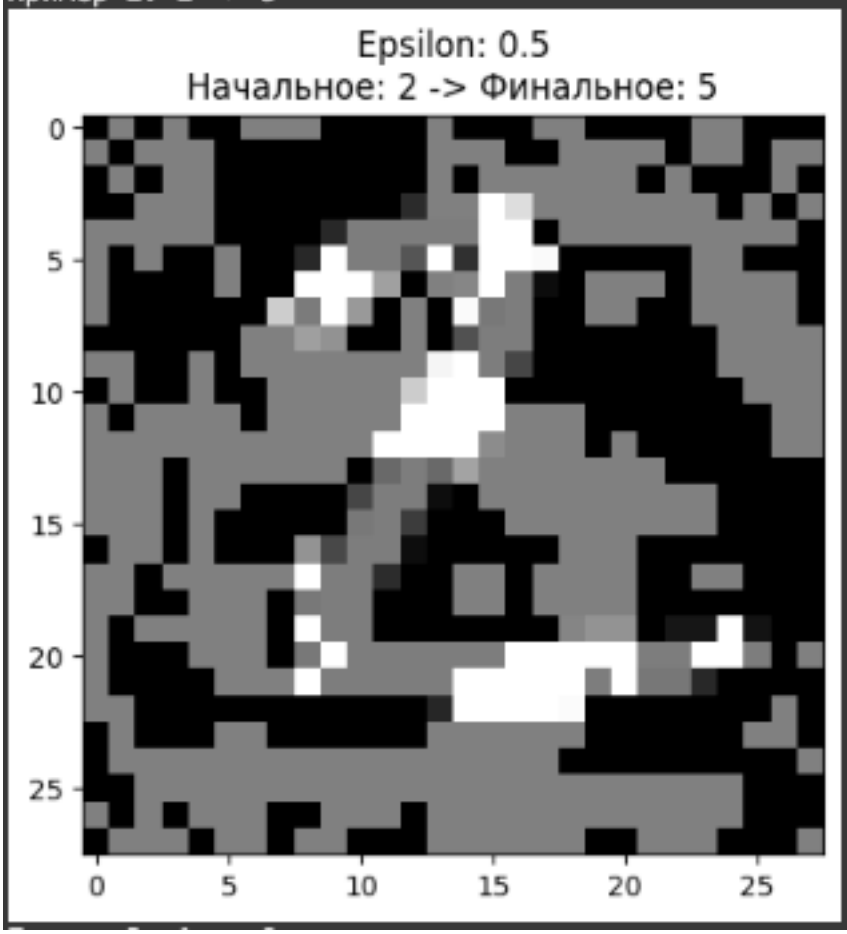

Epsilon: 0.02
Начальное: 7 -> Финальное: 7
0
5
10
15
20
25
0
5
10
15
20
25

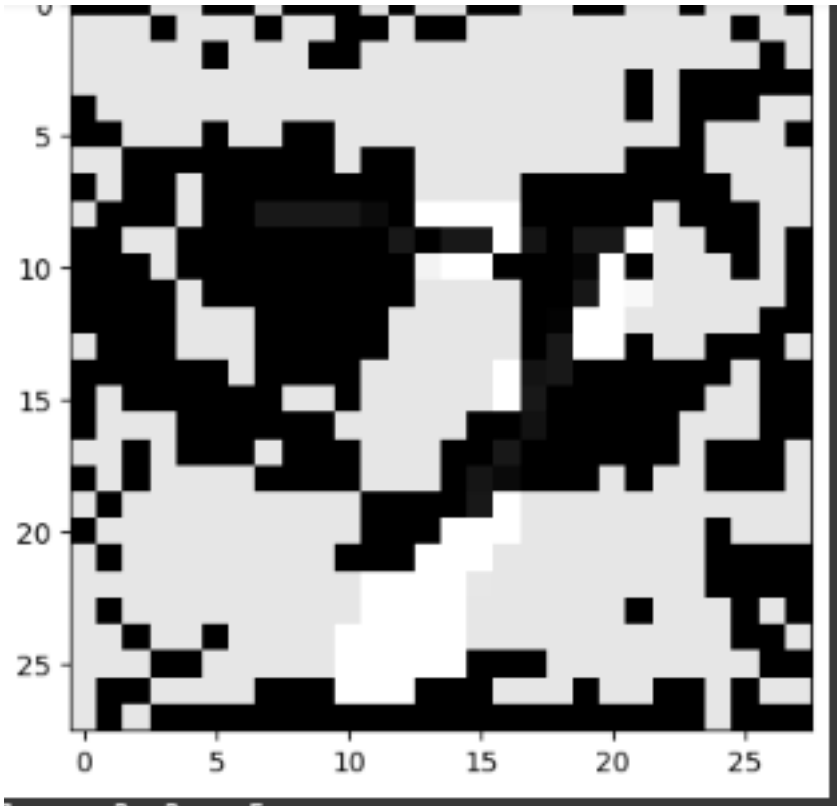
Пример 2: 2 -> 2


Epsilon: 0.02
Начальное: 2 -> Финальное: 2
0
5
10
15
20
25
0
5
10
15
20
25

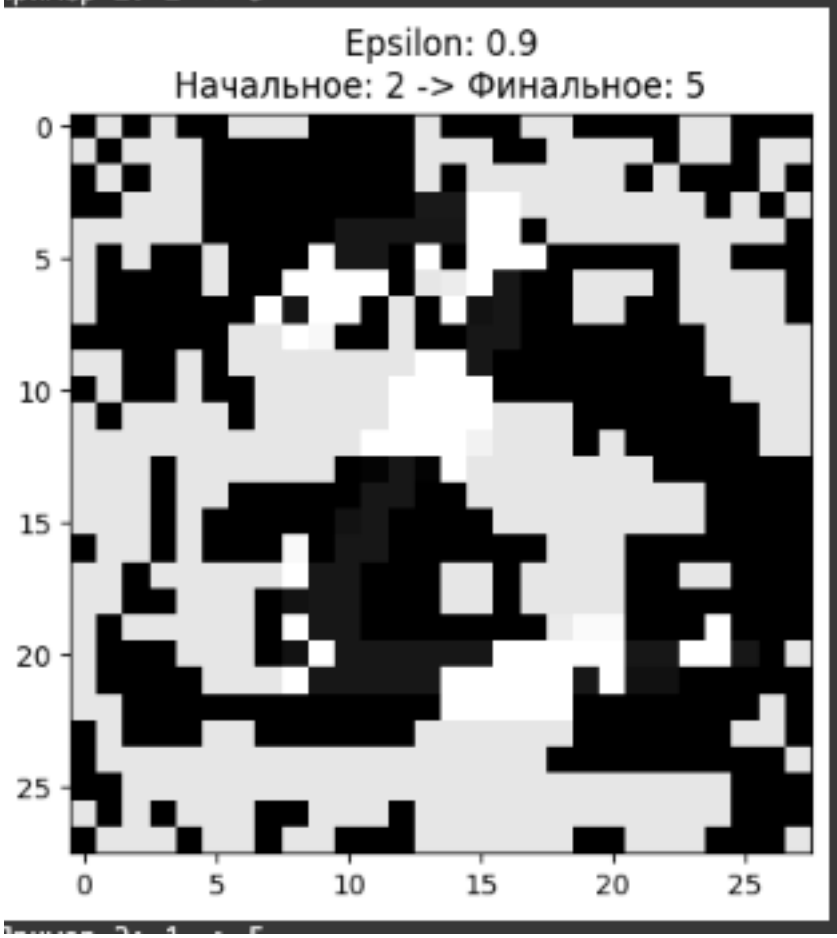

Epsilon: 0.5
Начальное: 7 -> Финальное: 3
0
5
10
15
20
25
0
5
10
15
20
25

Пример 2: 2 -> 5


Epsilon: 0.5
Начальное: 2 -> Финальное: 5
0
5
10
15
20
25
0
5
10
15
20
25

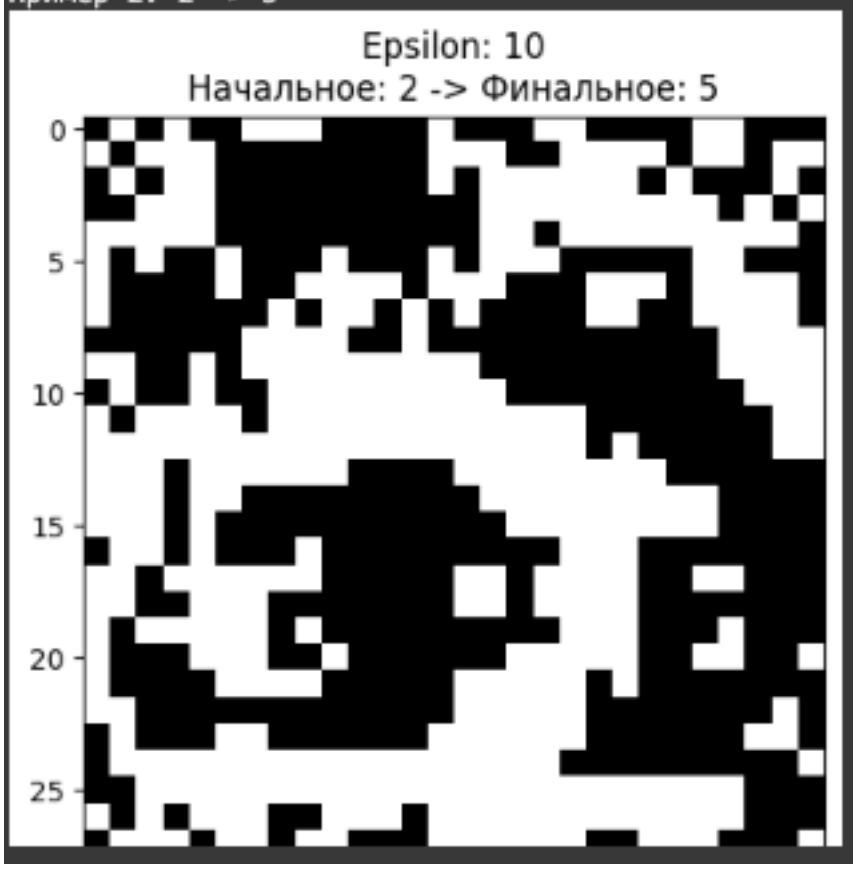

0
5
10
15
20
25
0
5
10
15
20
25

Пример 2: 2 -> 5


Epsilon: 0.9
Начальное: 2 -> Финальное: 5
0
5
10
15
20
25
0
5
10
15
20
25


Epsilon: 10
Начальное: 7 -> Финальное: 3
0
5
10
15
20
25
0
5
10
15
20
25

```
Пример 2: 2 -> 5
```


Epsilon: 10
Начальное: 2 -> Финальное: 5
0
5
10
15
20
25

```python
import torch
import torch.nn as nn
import torch.optim as optim
import torchvision
import torchvision.transforms as transforms
import matplotlib.pyplot as plt
import numpy as np
```

```python
# Трансформации для нормализации данных
transform = transforms.Compose([
    transforms.ToTensor(),
])

# Загрузка датасета MNIST
trainset = torchvision.datasets.MNIST(root='./data', train=True,
                                      download=True, transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=64,
                                          shuffle=True)

testset = torchvision.datasets.MNIST(root='./data', train=False,
                                     download=True, transform=transform)
testloader = torch.utils.data.DataLoader(testset, batch_size=1,
                                         shuffle=False)
```

```python
class SubstituteFCNN(nn.Module):
    def __init__(self):
        super(SubstituteFCNN, self).__init__()
        self.fc1 = nn.Linear(28*28, 256)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(256, 10)

    def forward(self, x):
        x = x.view(-1, 28*28)  # Развертывание изображения в вектор
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x

# Инициализация модели, функции потерь и оптимизатора
substitute_model = SubstituteFCNN()
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(substitute_model.parameters(), lr=0.001)
```

```python
# Проверка устройства (CPU или GPU)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
substitute_model.to(device)

# Функция обучения
def train_substitute(model, device, trainloader, optimizer, criterion, epochs=5):
    model.train()
    for epoch in range(epochs):
        running_loss = 0.0
        for data, target in trainloader:
            data, target = data.to(device), target.to(device)

            optimizer.zero_grad()
            outputs = model(data)
            loss = criterion(outputs, target)
            loss.backward()
            optimizer.step()

            running_loss += loss.item()
        print(f'[Substitute Model] Epoch {epoch+1}, Loss: {running_loss/len(trainloader)}')

# Обучение модели
train_substitute(substitute_model, device, trainloader, optimizer, criterion, epochs=5)
```

```
[Substitute Model] Epoch 1, Loss: 0.30178928797814386
[Substitute Model] Epoch 2, Loss: 0.1289351133893786
[Substitute Model] Epoch 3, Loss: 0.08606629233771582
[Substitute Model] Epoch 4, Loss: 0.0612714388581521
[Substitute Model] Epoch 5, Loss: 0.04727650730948109
```

```python
def create_adversarial_examples(model, device, testloader, epsilon):
    model.eval()
    adv_examples = []

    for data, target in testloader:
        data, target = data.to(device), target.to(device)
        data.requires_grad = True

        output = model(data)
        init_pred = output.max(1, keepdim=True)[1]

        if init_pred.item() != target.item():
            continue

        loss = criterion(output, target)
        model.zero_grad()
        loss.backward()
        data_grad = data.grad.data

        perturbed_data = fgsm_attack(data, epsilon, data_grad)
        adv_examples.append((perturbed_data, target))

    return adv_examples
```

```python
class TargetFCNN(nn.Module):
    def __init__(self):
        super(TargetFCNN, self).__init__()
        self.fc1 = nn.Linear(28*28, 128)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(128, 10)

    def forward(self, x):
        x = x.view(-1, 28*28)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

```python
# Инициализация целевой модели, функции потерь и оптимизатора
target_model = TargetFCNN()
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(target_model.parameters(), lr=0.001)

# Обучение целевой модели
def train_target(model, device, trainloader, optimizer, criterion, epochs=5):
    model.train()
    for epoch in range(epochs):
        running_loss = 0.0
        for data, target in trainloader:
            data, target = data.to(device), target.to(device)

            optimizer.zero_grad()
            outputs = model(data)
            loss = criterion(outputs, target)
            loss.backward()
            optimizer.step()

            running_loss += loss.item()
        print(f'[Target Model] Epoch {epoch+1}, Loss: {running_loss/len(trainloader)}')

# Обучение целевой модели
train_target(target_model, device, trainloader, optimizer, criterion, epochs=5)
```

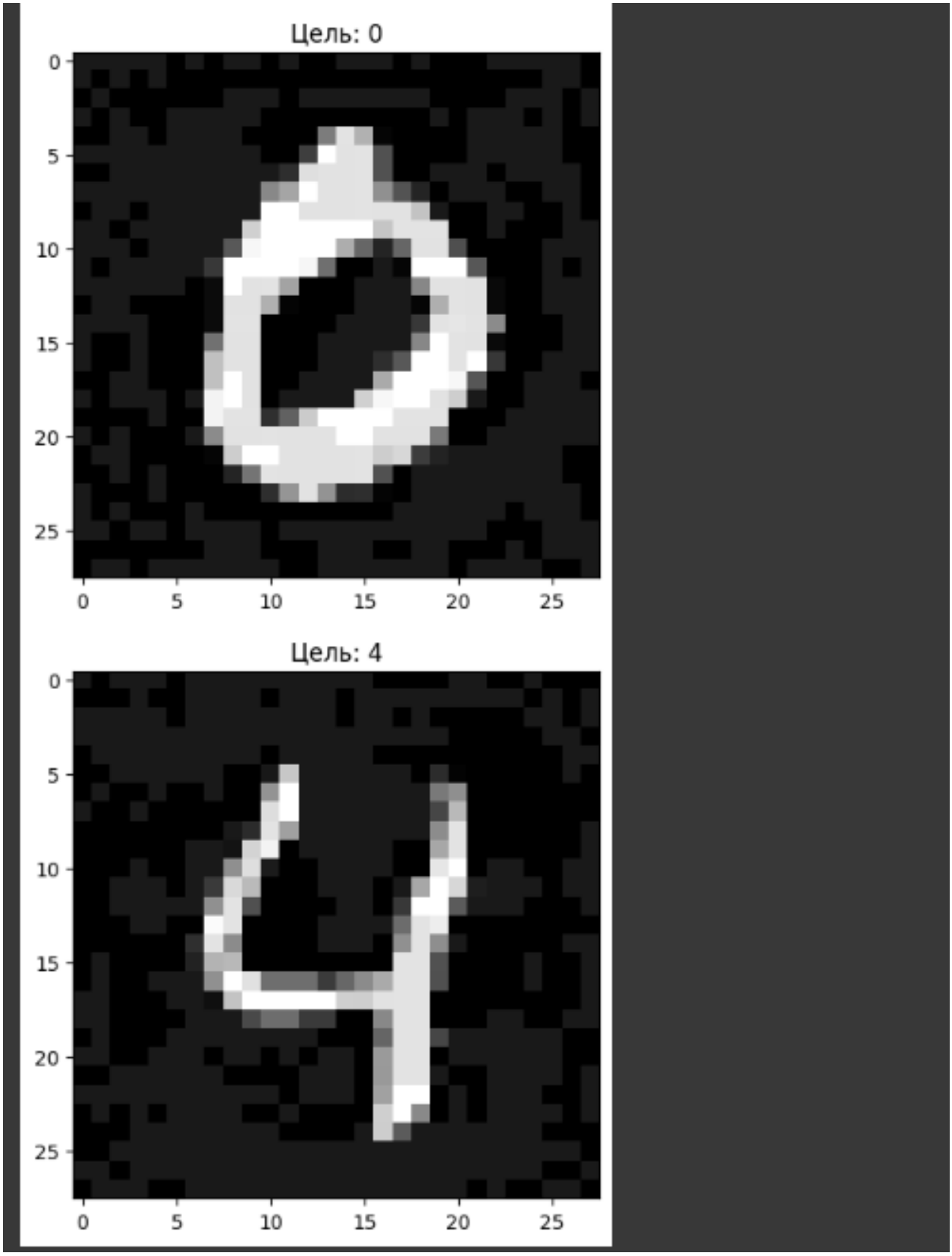
```
[Target Model] Epoch 1, Loss: 0.3455870131821012
[Target Model] Epoch 2, Loss: 0.15820899675649874
[Target Model] Epoch 3, Loss: 0.1104910367144657
[Target Model] Epoch 4, Loss: 0.08266963377030992
[Target Model] Epoch 5, Loss: 0.0642885443906405
```

```python
def test_target_on_adversarial(model, device, adv_examples):
    model.eval()
    correct = 0
    for perturbed_data, target in adv_examples:
        output = model(perturbed_data)
        final_pred = output.max(1, keepdim=True)[1]
        if final_pred.item() == target.item():
            correct += 1
    final_acc = correct / len(adv_examples)
    print(f'[Target Model] Accuracy on Adversarial Examples: {final_acc * 100:.2f}%')
```

```
epsilon = 0.1  # Выбранное значение epsilon
adv_examples = create_adversarial_examples(substitute_model, device, testloader, epsilon)
test_target_on_adversarial(target_model, device, adv_examples)
```

```
[Target Model] Accuracy on Adversarial Examples: 20.22%
```

```
# Функция для отображения адверсариальных примеров
def visualize_adversarial_examples(adv_examples, epsilon):
    for i, (perturbed_data, target) in enumerate(adv_examples):
        plt.figure()
        plt.imshow(perturbed_data.squeeze().detach().cpu().numpy(), cmap='gray')
        plt.title(f'Цель: {target.item()}')
        plt.show()
        if i == 4:  # Отображаем первые 5 примеров
            break

visualize_adversarial_examples(adv_examples, epsilon)
```


Цель: 0
Цель: 4