For this assignment, two networks, LeNet and 3C+2FC were used. The details of the networ ks are shown below.

Details of models:

LeNet

```
(conv1): Conv2d(1, 20, kernel_size=(5, 5), stride=(1, 1))
(conv2): Conv2d(20, 50, kernel_size=(5, 5), stride=(1, 1))
(avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
(fc1): Linear(in_features=50, out_features=500, bias=True)
(fc2): Linear(in_features=500, out_features=10, bias=True)
(relu): ReLU()
```

3C+2FC

```
(conv1): Conv2d(1, 20, kernel_size=(5, 5), stride=(1, 1))
(bn1): BatchNorm2d(20, eps=1e-05, momentum=0.1, affine=True,    track_r
unning_stats=True)
(conv2): Conv2d(20, 50, kernel_size=(5, 5), stride=(1, 1))
(bn2): BatchNorm2d(50, eps=1e-05, momentum=0.1, affine=True, track_runn
ing_stats=True)
(conv3): Conv2d(50, 100, kernel_size=(3, 3), stride=(1, 1))
(bn3): BatchNorm2d(100, eps=1e-05, momentum=0.1, affine=True, track_run
ning_stats=True)
(fc1): Linear(in_features=100, out_features=50, bias=True)
(fc2): Linear(in_features=50, out_features=10, bias=True)
(relu): ReLU()
```

Details of loss function and optimizer:

```
loss_func = nn.CrossEntropyLoss()
optimizer = optim.SGD(net.parameters(), lr=0.001, momentum=0.9)
```

Each net is trained for two different datasets: MNIST and USPS. MNIST is a digit dataset fro m National Institute of Standards and Technology, while USPS is scanned from envelopes b y the U.S. Postal Service.

Then, some adversarial examples are made using the Fast Gradient Signed Method (FGSM) attack. The accuracy of each model before and after the attack is recorded. Also, the accura cy of each model after the attack using other models' adversarial examples is recorded. Note that here epsilon is set as 0.6.

	Without attack	LeNet (MNIST)	LeNet (USPS)	3C+2FC (MNIST)	3C+2FC (USPS)
LeNet (MNIST)	98%	15%	24%	93%	91%
LeNet (USPS)	93%	29%	20%	88%	91%
3C+2FC (MNIST)	99%	70%	74%	39%	48%
3C+2FC (USPS)	96%	71%	73%	44%	25%

As can be seen, each model has a very good accuracy before the attack. Each model is the weakest when attacked using adversarial examples made from itself. LeNet models are relatively strong when attacked using examples made from 3C+2FC models, while being weaker when attacked using examples made from LeNet models. Conversely, 3C+2FC models are relatively strong when attacked using examples made from LeNet models, while being weaker when attacked using examples made from 3C+2FC models. It can be concluded an adversarial example is the most effective when it is used against the model which it was produced.

Also, I tried to vary the epsilon when attacking LeNet (MNIST) using adversarial examples made from itself. The accuracy record is shown below.

Epsilon	Accuracy
0.05	79%
0.1	78%
0.2	74%
0.3	71%
0.4	67%
0.5	35%
0.6	15%

As can be seen, the higher the epsilon, the lower the accuracy. As epsilon increases, it becomes easier to fool the network. However, this comes as a trade-off which results in the perturbations becoming more identifiable.