

AI 보안 과제 #3 - Adversarial Training for git hub report

<과제 개요 및 목표>

학습된 MNIST 이미지 분류 모델에 대한 Adversarial Training이며, 성능을 개선하여 Test Accuracy를 90%이상 올리는 것이 목표입니다.

<데이터셋 설명>

MNIST 데이터셋은 손으로 쓴 숫자 이미지로 구성된 대표적인 이미지 분류 데이터셋입니다. 이 데이터셋은 주로 이미지 분류 및 딥러닝 모델 학습에 사용됩니다. MNIST 데이터셋은 60000개의 훈련 이미지와 10000개의 테스트 이미지로 구성되어 있으며, 각 이미지는 28x28 픽셀 크기의 흑백 이미지입니다. 이를 통해 Adversarial Training을 적용하여 모델의 강인함을 향상 시키고, Adversarial Example에 대해 높은 정확도를 달성하는 것이 목표입니다.

<모델 설명>

CNN모델을 선택하여 MNIST 손글씨 이미지 분류 작업을 수행합니다. CNN은 이미지 데이터에 대하여 특화된 구조를 가지고 있으며, 이미지의 공간적 특징을 효과적으로 학습할 수 있는 강력한 모델입니다. 각 컨볼루션 층은 이미지의 특징을 추출하고, 풀링 층은 중요한 정보만 남겨서 차원을 축소하여 계산량을 줄이고 효율적인 학습을 지원합니다.

<수정된 모델 설계 및 이유>

이미지 분류 작업에 적합한 모델로 CNN을 선택하였습니다. CNN은 이미지의 공간적 특징을 효과적으로 학습할 수 있는 구조이기 때문에, 손 글씨 숫자 분류와 같은 이미지 인식 문제에 매우 유용합니다. 특히, MNIST와 같은 간단한 이미지 분류 문제에서는 CNN이 매우 뛰어난 성능을 보입니다.

또한, 본 프로젝트에서는 Adversarial Training을 적용하고 FGSM 기법을 사용하여 모델의 강인함을 평가합니다. 테스트 시 FGSM 공격에서 epsilon 값은 0.1로 설정되었으며, 이 값은 실험적으로 기본 CNN 모델에서도 충분히 좋은 성능을 낼 수 있을 것으로 예상되어 모델 구조의 큰 변경 없이 진행하였습니다. 즉, 기존 코드에서 제공된 구조를 그대로 사용하고, 추가적인 레이어 수정이나 복잡한 구조 변경 없이 실험을 진행하였습니다.

<학습 과정 및 성능 비교>

모델을 학습하는 과정에서 FGSM 공격의 엡실론은 0.5로 설정하여 강인한 특징을 학습하도록 하였고, FGSM의 엡실론 값은 모델의 학습에 영향을 미치고, 공격 강도를 설정합니다.

엡실론 값이 크거나 낮으면 과도하게 민감해져 제대로 학습 못하는 경우가 있기 때문에, 0.5로도 충분히 강한 공격을 하면서도, 과적합을 방지하도록 하였습니다.

```
torch.save(model.state_dict(), model_path)
```

```
Epoch [1/10], Loss: 0.1669, Accuracy: 26.76%  
Epoch [2/10], Loss: 0.0405, Accuracy: 30.73%  
Epoch [3/10], Loss: 0.0262, Accuracy: 30.34%  
Epoch [4/10], Loss: 0.0189, Accuracy: 32.45%  
Epoch [5/10], Loss: 0.0129, Accuracy: 34.43%  
Epoch [6/10], Loss: 0.0101, Accuracy: 37.27%  
Epoch [7/10], Loss: 0.0085, Accuracy: 40.56%  
Epoch [8/10], Loss: 0.0058, Accuracy: 42.94%  
Epoch [9/10], Loss: 0.0050, Accuracy: 46.72%  
Epoch [10/10], Loss: 0.0039, Accuracy: 50.55%
```

FGSM 값이 0.5일 때, 학습률이 50%라는 낮은 기록이 나왔지만, 테스트 과정에서는 0.1로 테스트하기 때문에 학습이 끝나고 검증 하기 위한 값으로 엡실론 값을 0.1로 낮추었습니다.

```

3초 # 학습 모델 검증
evaluate(model, test_loader, criterion, epsilon=0.1, is_train=True)

Training Accuracy on adversarial examples: 94.56%
94.56

```

학습한 모델을 엡실론 0.1로 검증을 하였을 때, 정확도가 은 94.56%로 준수한 성능이 나왔으며, 이 모델을 저장하고 테스트 환경에 모델을 로드하여 FSGM의 엡실론 값이 0.1일 때 테스트하였습니다.

```

<ipython-input-36-61627adc469e>:4: FutureWarning: You are using `torch.load` with `weights_only=False` (the current
RE_model.load_state_dict(torch.load(model_path))

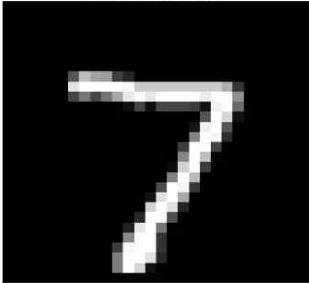
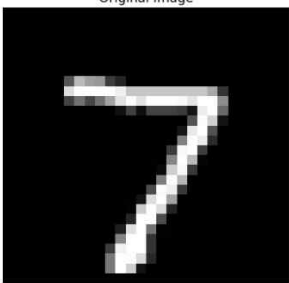
# Adversarial Examples에 대한 정확도 평가 (테스트)
adversarial_accuracy = evaluate(RE_model, test_loader, nn.CrossEntropyLoss(), epsilon=0.1, is_train=False)

Test Accuracy on adversarial examples: 94.56%

```

로드한 가중치 모델로 FSGM의 엡실론 값을 0.1로 맞추고 테스트를 해본 결과 검증 테스트와 같은 94.56%로 높은 정확도가 나와 목표한 수치를 달성했다는 것을 알 수 있습니다.

<결과 요약 및 분석>

Train image	Test image
	

학습 과정에서 엡실론 값 0.5를 사용하여 공격에 대한 모델의 내성을 학습하였으며, 테스트 과정에서 FSGM 공격의 엡실론 0.1을 설정하여 실제 상황에서의 모델을 테스트 했습니다 학습 정확도는 40~50%대로 다소 낮지만, 과적합을 방지하면서, 모델의 내성을 높였고, 테스트 환경에서 엡실론 값을 0.1로 설정하여 평가한 결과. 처음 목표한 정확도인 90%를 초과하여, 모델이 Adversarial 공격에 잘 대응했다는 걸 알 수 있습니다. 하지만 엡실론 값은 공격 강도를 정하는 파라미터로 테스트 과정에서 0.3, 0.5등 다른 값을 하면 모델의 성능이 떨어질 거라는 예측도 가능합니다. 그리하여 적절한 엡실론 값을 선택하는 것이 가장 중요합니다.