AI 보안 과제#1 보고서

1) 과제 개요 및 목표

<개요>

AI 보안 과제 1번은 코랩 및 로컬에서 GPU or CPU를 사용하여 CIFAR-10 데이터 셋을 통한 이미지 분류 모델을 개발하고 모델을 수정하거나 개선하여 CIFAR-10 데이터 셋에 대한 모델의 정확도를 개선합니다.

<목표>

LeNet-5 모델의 구조적 한계를 극복하기 위해 ResNet, AlexNet, VGGNet등 다양한 모델을 적용하여 개선하는 것이 목적입니다, 저는 ResNet9 모델로 통해 정확도 90%이상을 달성하는 것이 목표입니다. 이를 통하여, LeNet-5와 ResNet9 모델 간의 성능 차이 비교 및 분석하고, 각 모델이 CIFAR-10 데이터셋에서 이미지를 분류하면서 보이는 학습 속도, 로스률, 정확도를 확인하는 것입니다.

2) 데이터 셋 및 기존 모델 설명

<CIFAR-10 설명>

CIFAR-10은 머신 러닝 및 컴퓨터 비전 연구에서 많이 사용되는 이미지 데이터 셋으로 비행기, 자동차, 고양이, 새, 사슴, 개, 개구리, 말, 배, 트럭라는 10개의 클래스가 있으며, 총 60,000개의 32 * 32 픽셀 이미지를 포함합니다, 각 클래스는 6,000개의 이미지로 구성되어 있으며, 50,000개의 훈련 이미지와 10,000개의 테스트 이미지로 나뉩니다. 또한 RGB색 공간에서 컬러 이미지로 제공되어, 다양한 배경과 객체가 포함되어 있습니다. 그렇기 때문에, 간단한 이미지 분류 작업 위한 기준 데이터 셋으로서, 다양한 모델의 성능을 비교 및 개선하기 위한 기준으로 사용되고 있습니다.

<LeNet-5 설명>

LeNet-5는 초기 CNN의 모델 중 하나로, 이미지 인식 작업을 위해 설계된 모델입니다. 보통은 MNIST 분류 작업에 사용되고 있습니다. 구성 요소로 입력 이미지의 크기는 32x32 흑백 이미지를 보통 사용하며, 구조는 첫 번째 레이어 계층에서 6 개의 5x5 필터로 특징을 추출합니다. 평균 폴링 계층을 통해 차원을 줄여서 연산량을 낮추고, 위치에 대한 불변성을 제공합니다. 그런 다음 두 번째 레이어 계층에서는 16개의 5x5 필터로 더 복잡한 특징을 추출하고 다시 차원을 축소합니다. 세 번째 레이어 계층은 120개의 필터로 마지막 특징을 추출한 후, 120개에서 84개의 뉴런으로 축소 시킨 후에 클래스의 수에 맞게 뉴런 수를 조절합니다. 이러한 특징으로 LeNet-5모델은 간단한 구조와 적은 매개변수로 인해, 작은 데이터 셋 분류 모델 및 학습 초기 단계의 연구에 적합한 모델입니다.

3) 수정된 모델 설계 및 이유

<ResNet-9 설명>

ResNet-9는 Residual Network 구조를 기반으로 한 심층 신경망입니다. ResNet 모델은 딥러닝에서 흔히 발생하는 기울기 소실 문제를 해결하기 위해 설계되었습니다. 주요한 특징으로, Skip Connection 또는 잔차 연결을 사용하여 층을 건너뛰는 방식으로, 네트워크가 깊어질수록 발생하는 학습 어려움을 완화시켜줍니다.

<ResNet-9 설계 이유>

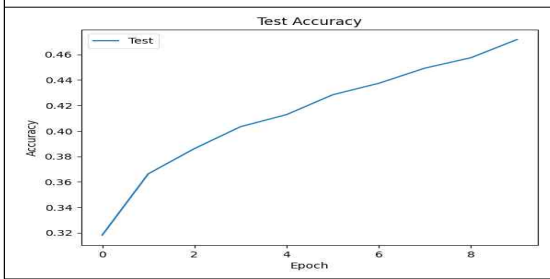
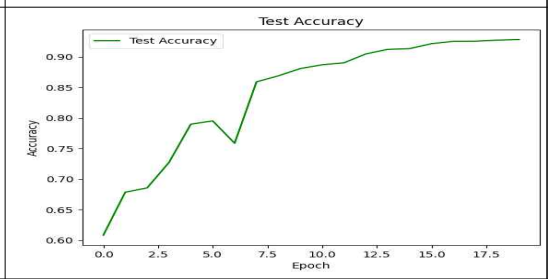
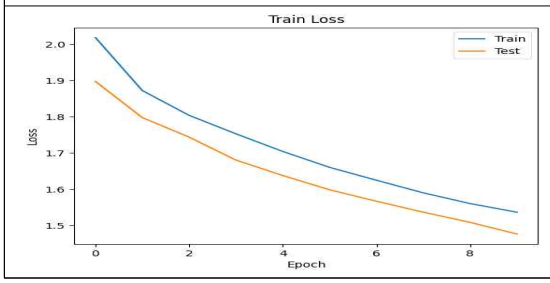
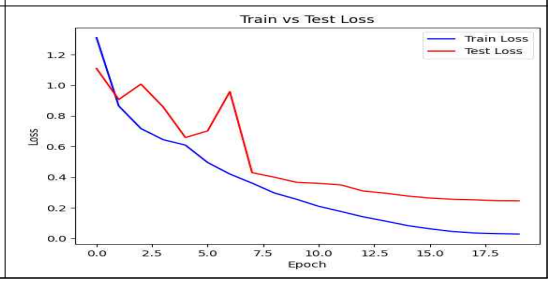
LeNet-5는 간단한 구조로 적은 매개변수이기 때문에. 간단한 작업에 적합하지만, CIFAR-10같은 복잡한 이미지 데이터 셋에는 구조상의 한계가 있어서 적합하지 않다고 판단하였습니다. ResNet-9는 더 깊은 구조로, 4개의 컨볼루션 블록과 2개의 잔차 블록이 포함되어 있고, 배치 정규화와 ReLU 활성화 함수, 잔차 연결을 통해 더 깊은 네트워크를 학습할 수 있어서, CIFAR-10와 같은 복잡한 데이터 셋에도 효율적으로 학습할 수 있다고 판단하여 설계하게 되었습니다.

4) 학습 과정 및 성능 비교

<학습 과정>

CIFAR-10 데이터 셋을 다운로드 하고 데이터 증강 및 정규화 등으로 전처리를 수행하고, 데이터 로더를 통해 배치 단위로 모델에 GPU를 통해 공급될 수 있도록 데이터를 준비합니다. ResNet-9은 Conv Block과 Residual Block을 사용하여 설계되었으며, 합성곱 블록은 3x3 필터, 정규화, ReLU활성화 함수 및 2x2 맥스 폴링을 사용하였습니다. Residual Block에서는 입력 값에 블록의 출력을 더하는 스킵 연결을 기울기 소실 문제를 완화하며 성능을 향상시킵니다. 최종적으로 맥스 폴링과 Flatten을 거쳐 완전 연결 레이어에서 10개의 클래스 중 하나로 분류 합니다. OneCyleLR학습률 스케줄링을 통해 1번의 사이클로 조정하는 방식으로 초기에는 학습률을 높이고, 점진적 학습률을 감소시켜, 과적합을 방지하고 짧은 시간동안 학습하여도 좋은 성능을 나올 수 있도록 하였습니다. 옵티마이저는 Adam 옵티마이저를 사용하여, 평균과 분산을 추적하여 학습률을 자동으로 조정하며, 하이퍼파라미터에 민감하지 않게 학습을 안정적으로 진행시킵니다. 이런 방식을 통해, ResNet-9 모델이 적은 시간에도 효율적으로 학습을 하고 CIFAR-10 데이터 셋에서도 높은 성능을 달성할 수 있었습니다. 또한, 데이터 증강을 통해 모델이 다양한 패턴을 학습하도록 유도하여 성능을 향상시켰습니다. 학습 중 기울기 클리핑 기법을 사용하여 과도한 기울기 폭발을 방지하였습니다.

<성능 비교>

<table>
<tr><th>LeNet-5</th><th>ResNet-9</th></tr>
<tr><td>Number of Train Data : 50000
Number of Test Data : 10000
Number of Classes : 10
1/ 10: Train Loss [2.02] / Test Loss [1.90], Test Accuracy [31.81%]
2/ 10: Train Loss [1.87] / Test Loss [1.80], Test Accuracy [36.63%]
3/ 10: Train Loss [1.80] / Test Loss [1.74], Test Accuracy [38.61%]
4/ 10: Train Loss [1.75] / Test Loss [1.68], Test Accuracy [40.33%]
5/ 10: Train Loss [1.70] / Test Loss [1.64], Test Accuracy [41.28%]
6/ 10: Train Loss [1.66] / Test Loss [1.60], Test Accuracy [42.84%]
7/ 10: Train Loss [1.62] / Test Loss [1.57], Test Accuracy [43.74%]
8/ 10: Train Loss [1.59] / Test Loss [1.54], Test Accuracy [44.93%]
9/ 10: Train Loss [1.56] / Test Loss [1.51], Test Accuracy [45.75%]
10/ 10: Train Loss [1.54] / Test Loss [1.48], Test Accuracy [47.18%]</td><td>12/ 20: Train Loss [0.18] / Test Loss [0.35], Test Accuracy [89.01%]
13/ 20: Train Loss [0.14] / Test Loss [0.31], Test Accuracy [90.46%]
14/ 20: Train Loss [0.11] / Test Loss [0.30], Test Accuracy [91.21%]
15/ 20: Train Loss [0.08] / Test Loss [0.28], Test Accuracy [91.33%]
16/ 20: Train Loss [0.06] / Test Loss [0.26], Test Accuracy [92.13%]
17/ 20: Train Loss [0.05] / Test Loss [0.26], Test Accuracy [92.52%]
18/ 20: Train Loss [0.04] / Test Loss [0.25], Test Accuracy [92.54%]
19/ 20: Train Loss [0.03] / Test Loss [0.25], Test Accuracy [92.71%]
20/ 20: Train Loss [0.03] / Test Loss [0.25], Test Accuracy [92.82%]</td></tr>
<tr><td>  </td><td>  </td></tr>
<tr><td>  </td><td>  </td></tr>
</table>

5) 결과 요약 및 분석

<결과>

LeNet는 단순한 구조이기 때문에 에포크를 10번 반복하더라도 유의미한 변화 없이 조금씩 성능이 올라갔지만, 결국에는 47.18%에서 훈련을 멈췄습니다. 반면에 ResNet-9는 초기 에포크에서는 60% 후반의 성능을 보여주다가 반복 학습을 통해 점점 성능이 나아지고 결국에 13에포크에서는 90%가 돌파하였고, 최종적으로 20에포크에서 92.82%라는 유의미한 결과가 나왔습니다. 이를 통해 ResNet-9를 통한 CIFAR-10 이미지 분류 작업이 성공적으로 마쳤다는걸 알 수 있습니다. 그리하여, 단순한 이미지 분류 작업에서는 LeNet-5 모델이 적합하지만, 복잡한 이미지 데이터 셋 경우에는 ResNet같은 깊은 네트워크가 필요하다는 걸 알 수 있었습니다.