

**딥러닝 hw4 과제**

**과목명 딥러닝**

**담당교수 장익범교수님**

**제출일 20231214**

**전공 컴퓨터전자시스템**

**학번 201904458**

**이름 이준용**

1. **각 채널 별 평균과 표준편차를 넣어서 normalization을 한 이유와 장점 서술**

이미지를 정규화 할 때 보통 생각하면 평균을 빼고 그 값에 표준 편차로 나누어 정규화하는 것을 생각해 볼 수 있습니다.

첫번째 장점으로 표준편차로 나누면 데이터의 크기가 조정되고, 정규화된 데이터는 더 작은 범위에 속하므로 최척화를 더욱 안정적으로 만들고 활성화 함수의 포화상태를 방지할 수 있습니다.

두번째로 정규화를 통해 모델이 입력 분포의 변화에 덜 민감해지며, 보이지 않는 데이터에 대한 일반화가 향상됩니다.

세번째로 평균을 빼면 데이터가 0을 중심으로 중앙에 배치되어 수렴하여 훈련됩니다. 빠른 수렴이 가능 해집니다.

결론은 수행한 채널별 평균 및 표준 편차 정규화는 각 색상 채널(빨간색, 녹색, 파란색)을 독립적으로 처리하는 데 도움이 됩니다. 채널마다 값 범위가 다를 수 있으며 정규화는 이러한 범위를 보다 균일하게 만드는 데 도움이 되기 때문에 중요하다고 생각합니다. 그러므로 신경망이 특정 채널의 강도에 편향되지 않고 효과적인 학습을 기대할 수 있습니다.

1. **Hyperparameter를 바꾸며 도출된 성능 그래프를 첨부하고,**

**자신의 코드를 예시로 그래프의 결과 설명하기.**

* optimizer\_18 = optim.SGD(resnet18.parameters(), lr=0.01, momentum=0.9)

텍스트, 스크린샷, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명텍스트, 그래프, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

훈련이 잘 안되었다고 생각합니다. 정확도는 50%이상이지만 Validation loss의 그래프를 안따라가고 loss 차이가 점점 심해지는 모습을 볼 수 있습니다.

* optimizer\_50 = optim.SGD(resnet50.parameters(), lr=0.001, momentum=0.85)

텍스트, 스크린샷, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

훈련은12.5에폭에서 점점 loss 차이가 커지는 모습을 보이지만 그래도 훈련이 되었다고 판단합니다.

학습률을 resnet 18보다 0.1더 낮춰서 진행하였습니다.

* optimizer\_152 = optim.SGD(resnet152.parameters(), lr=0.0001, momentum=0.9,weight\_decay=0.0001)

텍스트, 스크린샷, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

Loss는 잘 줄어들고 있지만 정확도가 11%로 너무 낮아 훈련이 잘 안되었다고 생각합니다.

1. **각 모델 별 출력 성능에 어떤 차이가 있으며, 성능이 왜 그렇게 출력된 것 같은지 본인의 생각 서술.**

Resnet 18이 정확도가 제일 높아 성능이 좋았다. 이후로 resnet50과 resnet152는 신경망이 복잡해지고 뉴런 수가 많아져 fc layer까지 도달하지 못해 출력을 못하여 정확도 측면에서 낮아졌다고 생각합니다.

1. **첨부된 파일을(resnetScratch.ipynb) 토대로 resnet의 구조의 특징에 대해 서술**

Resnet의 경우 conv layer을 여러 층을 쌓아서 깊어짐에 따라 발생하는 과적합과 기울기 소멸을 skip connection을 사용하는 Residual Block를 제안했기 때문에 이와 같은 이론 개념을 resnet을 만들어보는데 적용하였습니다.

기초적인 block를 다음과 같이 작성하였습니다. Input data는 처음으로 Conv1 layer를 통과하고 학습을 빠르게 하기위해 batch normalization(BN)을 통과하게 만들었습니다. 이후로 같은 conv2 layer와 BN을 하나 더 초기화 layer로 생성하였습니다. 이후에 forward에 relu와 다운샘플링을 통해 결과를 도출합니다.

다음으로는 resnet클래스를 만들어서 input 채널 64로 들어와서 conv2d(kernel\_size=7, stride=2, padding=3) -> BN -> relu -> maxpooling(kernel\_size=3, stride=2, padding=1)를 통해 resnet block를 생성하였습니다.

위와 같은 block을 총 4개의 layer를 stride=2조건으로 통과하도록하였습니다.

최종적으로 fc layer에서는 avgpooling를 사용하고 flatten하여 결과를 도출합니다.

1. **과제를 진행하며 어려웠던 점 서술 (어려워던 부분이 없다면, 본인이 이해한 내용을 기반으로 3줄 요약해서 작성)**

직접 만든 resent 모델을 summary()함수를 통해 어떻게 이미지가 통과하는지 한 줄 한 줄 보며 이해하는데 시간이 걸렸습니다. 그리고 좋은 결과의 훈련과 정확도를 어떻게 높여야 될지 하이퍼파라미터 조정하는 부분에서 어려웠습니다.

1. **과제를 진행하며 배운 점과 느낀 점 서술 ('r5번'의 3줄 요약과 중복도 가능하며 중복된 내용이라도 작성하시길 바랍니다. 또한, 느낀 점은 반드시 다른 내용으로 작성해야합니다.)**

이론으로 배운 resnet을 직접 작성해보면서 input 이미지가 resnet을 어떻게 통과하는지 알게되었습니다. 이를 통해