

CNN을 활용한 흑색종과 일반적인 점을 판별하는 모델 개발

Hyun_min_KIM

김 현 민

Abstract

흑색종은 일종의 피부암으로 일반 점과 구분하기 어렵다. 따라서 나는 인공지능 모델을 학습시켜 일반 점과 흑색종의 점의 특성으로 구분하여 학습시켜 구분할 수 있는 모델을 만들고 싶어 이 모델을 생각했다. 우선 모델을 학습시키기 위한 흑색종과 일반 점의 데이터셋을 구하여 약 1,000장의 사진으로 모델을 총 2개의 컨볼루션 계층과 2개의 맥스풀링 그리고 3개의 fully connected를 사용하여 학습시켰다. 학습시키고 난 후 테스트 데이터로 확인해본 결과 약 70퍼센트의 정확도를 보여주었다. 따라서 이 모델에게 흑색종 사진을 주었을 때 70퍼센트의 확률로 흑색종을 진단 내릴 수 있는 모델이다.

1. 서론

흑색종은 일반 점과 비교하기 힘들다. 따라서 컴퓨터에 미리 학습시켜놓고 판단을 할 수 있는 모델을 만들고 싶어서 흑색종인 점 사진과 일반적인 점 사진으로 모델을 학습시켜 흑색종인지 일반 점인지 판별하는 모델을 만들어 사람들이 이 모델을 통해 더 쉽게 흑색종을 판별할 수 있는 모델을 만들기 위해 실험한 것이다.

2. 본론

2-1. 데이터 세트

케 글에서 흑색종과 일반 점이라는 데이터 세트를 가져와서 사용했다. test 파일에 흑색종,일반점 각각 150개씩 train 파일에 각각 500장씩 가져왔다. 사진은 300,300으로 이미지 크기를 지정하였고 256 크기로 이미지를 무작위로 자르고 크기를 조정했다.

2-2.수행환경

파이썬을 기반으로 한 파이 토치를 사용했고, 코 랩으로 실험하였음.측정 목적으로는 단순히 흑색종과 정상적인 점을 구분 지으는 것이다. 알고리즘 CNN 구조는 밑에 사진과 같다.

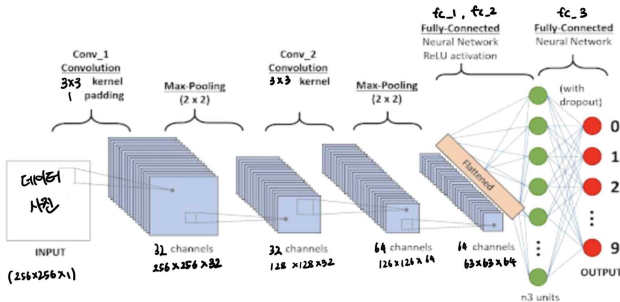


그림 1 fashionCNN 알고리즘 구조

하이퍼 파라미터는 우선 배치 크기는 16 ,에폭 10,학습률은 0.0001로 설정, 첫 in_features 값은 254016이고 out_features 값은 800으로 설정하였고 마지막 fc3의 out_features 값은 2 개의 클래스를 분류하는 것이므로 2로 지정하였다.

3. 결과

Epoch 0/9	Loss: 3.0238 Acc: 0.8140	Acc: 0.6200
Epoch 1/9	Loss: 1.6669 Acc: 0.8720	Loading model archive/01.pth Acc: 0.7233
Epoch 2/9	Loss: 1.6211 Acc: 0.8490	Loading model archive/02.pth Acc: 0.6833
Epoch 3/9	Loss: 0.9585 Acc: 0.8820	Loading model archive/03.pth Acc: 0.5800
Epoch 4/9	Loss: 0.7357 Acc: 0.8660	Loading model archive/04.pth Acc: 0.6333
Epoch 5/9	Loss: 0.5344 Acc: 0.8920	Loading model archive/05.pth Acc: 0.6933
Epoch 6/9	Loss: 0.5246 Acc: 0.8740	Loading model archive/06.pth Acc: 0.6367
Epoch 7/9	Loss: 0.3521 Acc: 0.8930	Loading model archive/07.pth Acc: 0.7300
Epoch 8/9	Loss: 0.3495 Acc: 0.8840	Loading model archive/08.pth Acc: 0.6767
Epoch 9/9	Loss: 0.3311 Acc: 0.8920	Loading model archive/09.pth Acc: 0.7500
Training complete in 2m 4s Best Acc: 0.893000		Validation complete in 0m 51s Best Acc: 0.750000

그림 1 훈련,테스트 데이터 정확도 숫자 표현

위 알고리즘 구조로 CNN 모델 학습 시 결과 값으로 처음 손실 값 3.0238 정확도:0.8140 나왔고 중간 5번째 에폭때는 손실 값 0.5344 정확도 0.8920 나왔고 마지막 9번째 에폭때는 손실 값 0.3311 정확도 0.8920 나왔다. 훈련 데이터의 최고정확도는 0.8930이고 테스트 데이터의 최고 정확도는 0.7500이다.

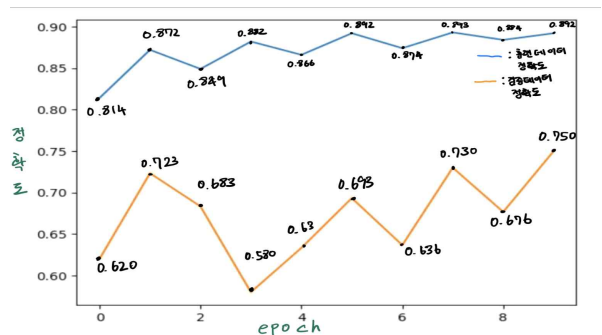


그림 2 훈련,테스트 데이터의 정확도 그래프

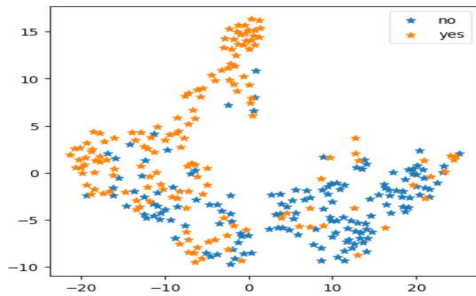


그림 3 훈련,테스트 데이터 산점도

참고문헌

- 1) 캐글(데이터 세트)
<https://www.kaggle.com/>
- 2) 알고리즘 구조 확인
<https://sonsnotation.blogspot.com/2020/11/7-convolutional-neural-networkcnn.html>
- 3) 수업 ppt자료

4. 고 찰

처음 모델을 만들 때 많은 수의 에폭을 사용하면 광범위한 학습과 상당히 높은 정확도를 얻을 수 있다고 믿었다. 훈련 데이터에 대한 광범위한 훈련 때문에 과적합의 발생하더라도 드롭아웃을 통합하여 완화할 수 있다고 가정했다. 그러나 모델을 광범위하게 훈련했음에도 불구하고 정확도는 특정 지점 이상으로 계속 증가하지 않았다. 다음 문제로는 더 많은 특성으로 구분해야겠다고 판단하여 사진의 크기와 특성값을 증가시키는 것과 fully connected 증가시켜 모델을 학습시키는 도 중 많은 계산으로 코 랩 환경의 ram의 최대 용량인 15기가를 넘기는 상황이 발생하여 15기가를 넘기지 않는 선에서 많은 특성을 추출할 수 있는 알고리즘 구조를 찾아냈다. 미니배치 크기를 16으로 하여 메모리의 효율성을 늘렸고 학습률을 조정할 때 학습률 0.0005~0.001 사이의 숫자들을 넣어 에폭 10 기준으로 손실 값이 급격하게 증가하는 부분이 있었다. 따라서 낮은 0.0001로 진행하여 기울기가 급격하게 하강하는 걸 예방하였다.

또 다른 문제로 데이터를 훈련데이터 1000개가 아닌 100개로 하였는데 성능이 최적화 되지 않아 더 많은 데이터를 수집하여 성능을 최적화 시켰다. 따라서 나는 이 실험을 하면서 데이터,알고리즘,알고리즘 튜닝, 하이퍼 파라미터를 이용하여 성능 최적화를 시켰다. 그리고 아쉬웠던 점으로 전이 학습을 이용하여 ResNet의 알고리즘을 선택하여 실행해 봤다면 더 좋은 결과가 나올 것이라고 보고서를 다 작성하고 나서 생각했다.

5. 결 론

우선 큰 궁금증은 모델이 많은 사진으로 훈련을 시킨다고 하였을 때 잘 구분해 있는지 궁금했고, 많은 데이터와 많은 에폭과 적당한 학습률이 있으면 매우 정확한 모델을 만들 수 있다고 생각하고 실험을 해보았지만 정확하지 않은 검증 데이터, 높지 않은 훈련데이터 기능 손실 등으로 실패했다. 많은 하이퍼파라미터의 조정이 필요하고 정규화 전처리 등 많은 방법이 필요한 것을 알았고 데이터를 전처리하고 드롭아웃과 하이퍼파라미터를 조정해가면서 약 70퍼센트의 정확도를 보여주는 만족스러운 결과를 보여줬다.