|  |  |  |  |
| --- | --- | --- | --- |
| **HW#02** | **딥러닝 과제보고서** | **학번: 2018741035** | **이름: 한준호** |
| #1. 과제 목표 | | | |
| Convolution Neural Network를 이용하여 개와 고양이의 종을 분류하는 모델인, Pet 분류기를 만들어본다. 만든 Pet 분류기가 Test set에서 높은 성능을 보여줄 수 있도록 Data augmentation, Validation set 생성, 모델의 레이어 구조, 학습 파라미터 등을 고려하여 CNN모델을 개선하여 test set에 대해서 일반화된 모델을 만들어본다. | | | |
| #2. 코드 수행 결과 (사진 첨부)  1. Validation Set 생성  텍스트이(가) 표시된 사진  자동 생성된 설명  Train data을 이용하여 모델을 학습시키고 학습시킨 모델의 성능을 확인하기위해, 이미지의 픽셀 밝기를 0에서 1사이로 정규화(Normalize)하여 전처리한 ‘train\_images’와 One-hot Encoding을 진행한 ‘train\_labels\_onehot’을 train\_test\_split() 함수를 이용하여 Data를 Train data 와 Validation data로 분할해주었다. 이 때, 효과적인 학습을 위해 Data를 분할할 때 x\_data와 y\_data를 동일 비율로 Suffle 해주었다.  2. Data Augmentaion  텍스트이(가) 표시된 사진  자동 생성된 설명  Augmentation한 Data 이미지의 결과를 확인하기 위한 함수를 만들어주었다.  2-1. Translation텍스트이(가) 표시된 사진  자동 생성된 설명 | | | |
| 텍스트이(가) 표시된 사진  자동 생성된 설명  Data Augmentation 기법 중 Translation을 진행하였다. 분할한 초기의 2000개의 Train Data에 상, 하, 좌, 우, 대각으로 Translation 진행하여 14000개의 Train Data로 Augmentation된 것을 확인하였다.  2-2. Flip    텍스트이(가) 표시된 사진  자동 생성된 설명  Data Augmentation 기법 중 Flip을 진행하였다. 분할한 초기의 2000개의 Train Data에 좌/우, 상/하로 Flip 진행하여 18000개의 Train Data로 Augmentation된 것을 확인하였다.  2-3. Rotate  텍스트이(가) 표시된 사진  자동 생성된 설명    텍스트이(가) 표시된 사진  자동 생성된 설명  Data Augmentation 기법 중 Rotation을 진행하였다. 분할한 초기의 2000개의 Train Data에 30, 60, 90, -45도로 Rotate 진행하여 26000개의 Train Data로 Augmentation된 것을 확인하였다.  2-4. Random Noise  텍스트이(가) 표시된 사진  자동 생성된 설명  텍스트이(가) 표시된 사진  자동 생성된 설명  Data Augmentation 기법 중 Random Noise를 추가하는 기법을 진행하였다. 분할한 초기의 2000개의 Train Data에 랜덤으로 노이즈를 추가하여 30000개의 Train Data로 Augmentation된 것을 확인하였다.  2-5. Distortion  텍스트이(가) 표시된 사진  자동 생성된 설명  텍스트, 화이트, 스크린샷이(가) 표시된 사진  자동 생성된 설명  Data Augmentation 기법 중 Distortion을 진행하였다. 분할한 초기의 2000개의 Train Data에 의도적으로 왜곡을 진행하여 36000개의 Train Data로 Augmentation된 것을 확인하였다.  2-6. Change Color  텍스트이(가) 표시된 사진  자동 생성된 설명  텍스트이(가) 표시된 사진  자동 생성된 설명  Data Augmentation 기법 중 Change Color을 진행하였다. 분할한 초기의 2000개의 Train Data에 의도적으로 색상을 변경하여 42000개의 Train Data로 Augmentation된 것을 확인하였다.  3. CNN Layer  텍스트이(가) 표시된 사진  자동 생성된 설명 테이블이(가) 표시된 사진  자동 생성된 설명  학습 성능을 높이기 위해 CNN 모델의 레이어 구조를 변경해주었다. BatchNormalization( ) 함수를 이용해 Batch Normalization을 진행하였고, Convolution 과정에서 이미지 데이터의 손실을 막기위해 Padding을 추가해 Convolution 과정에서 이미지 데이터의 크기가 줄어들지 않게 설정해주었다. 또한 학습을 여러 번 진행하면서 Dropout( ) 함수의 파라미터인 Drop rate를 0.6으로 설정해주었다.  4. 학습 파라미터  텍스트이(가) 표시된 사진  자동 생성된 설명  학습을 여러 번 진행하면서 학습이 완료된 모델을 Validation data를 이용해 성능을 확인하고, 이를 비교하면서 학습 파라미터를 설정해주었다. 옵티마이저(Optimizer)를 RMSprop와 Momentum 두 가지를 합친 듯한 방법인 ‘Adam’으로 설정하였고, ‘batch\_size’는 128, ‘epochs’는 60번으로 설정해주었다.  테이블이(가) 표시된 사진  자동 생성된 설명  설정한 학습 파라미터로 학습을 진행한 결과, Training Data에 대하여 loss는 0.0958, accuracy는 0.9750로 Training Data에 대한 모델의 성능을 확인하였다.  텍스트이(가) 표시된 사진  자동 생성된 설명  분할해뒀던 Validation Data Set을 이용해 학습한 모델의 성능을 확인하였다.  #3. 고찰  - Validation Data를 생성하고, 이미지 데이터를 Augmentation하는 기법들을 실습해볼 수 있었다. 또한 좋은 성능의 모델을 얻기 위해 CNN Layer의 구조를 변경해보고, 학습 파라미터를 수정하면서 좋은 성능을 얻기 위한 방법들에 대한 지식을 습득하고 실습해볼 수 있었다.  - 이미지 데이터를 Convolution 하는 CNN모델을 Colab 환경에서 과제를 진행하다보니 Data augmentation을 진행하여 Data가 늘어났을 때 시스템 Ram이 부족하여 세션이 종료되고 학습이 진행이 안되는 문제점이 있었다.  - Data augmentation을 진행하여 Train Data가 늘어났기 때문에, Colab GPU 환경에서는 하루에 한정된 시간 동안만 학습을 진행할 수 있는데 많은 Data를 학습하는데 시간이 오래 걸리기 때문에 학습 파라미터를 바꿔가면서 학습을 많이 진행하지 못한다는 문제점이 있었다.  - 레이어 구조나 하이퍼 파라미터를 적절하게 설정하지 않았을 경우, Training Data에 대해서는 좋은 성능을 얻었지만, Validation Data에 대해서는 낮은 성능이 나오는 Overfitting 문제 발생하는 것을 확인하였다. Overfitting을 막기 위해 Validation Data를 한번 더 나눠 Test Data를 생성하고 Train Data와 Validation Data 이용해 학습을 진행하였지만, 여전히 Validation Data에 대해서 Overfitting이 발생하였고 만족스러운 결과를 얻지 못하였다. 이는 충분한 Validation Data를 가지고 학습 시 Cross Validation을 진행한다면 Validation Data에 대해서도 Overfitting 문제를 막을 수 있을 것 같다는 생각이 들었다. | | | |
| #4. 결론  Validation Data 생성, Training Data Augmentation, 학습 모델의 목표에 맞는 CNN Layer의 구조 설정, 적절한 학습 파라미터를 설정을 통해 CNN 모델을 개선하여 모델의 성능을 높일 수 있었다. 하지만 적절한 CNN Layer의 구조와, 적절한 학습 파라미터를 설정 하는 것은 쉽지 않았다.  CNN 모델을 개선하여 Training Data에 대해서는 좋은 성능을 얻었지만, Validation Data에 대해서는 여전히 Overfitting에 의한 성능 감소를 막기 어려웠다. 이는 충분한 Validation Data를 가지고 학습 시 Cross Validation을 진행한다면 Validation Data에 대해서도 Overfitting 문제를 막을 수 있을 것 같다는 생각이 들었다. | | | |