

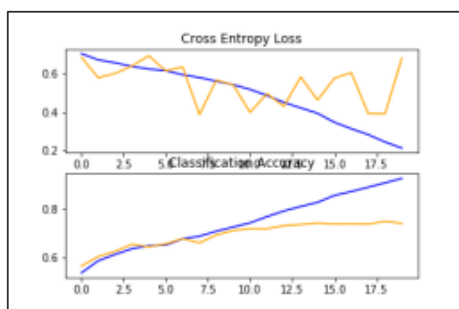
[개별 프로젝트]

CNN을 통한 개와 고양이 분류¹

가장 기본적인 이미지 분류 문제인 개와 고양이를 CNN을 통해 분류하는 프로젝트를 통해 CNN의 기본 원리를 익히고, 딥러닝 분야의 발전을 가져온 VGG16 모델을 통한 강화학습을 진행하였다.

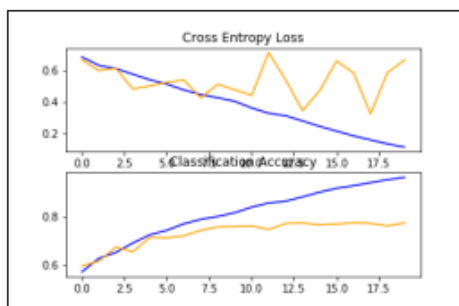
학습에 활용된 data set은 kaggle 경진대회에 사용되었던 데이터 세트를 통해 진행되었다.² 이 데이터 세트에는 크게 train data, test data로 나누어져 있으며, train data에 있는 25,000장의 정답이 있는 사진을 기반으로 학습을 진행했다. 먼저 해당 데이터의 정확도를 추후 추측하기 위해 train data 중에서도 25%를 분리하여 새로운 test data로 활용할 수 있도록 저장하였다.

데이터의 분류가 완료된 이후 CNN을 통해 학습을 위한 층을 쌓았다. 시간제한을 고려 한 개의 블록부터 총 3개까지 층을 쌓아서 그 정확도를 비교하였으며 이를 통해 층을 하나씩 쌓을 때마다 정확도가 증가함을 볼 수 있었다. 하기 그래프에서 파란색 선은 train data, 주황색 선은 test data를 지칭한다.



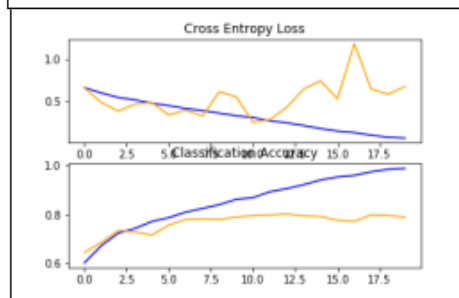
하나의 블록을 사용할 경우 train data에서의 정확도는 거의 완벽했지만, test data에서의 정확도는 어느 정도 상승한 후 학습이 이루어지지 않고 있음을 알 수 있다. 따라서 이는 overfitting 되었으며, 하나의 블록으로는 만족할만한 분류가 이루어지지 않음을 확인할 수 있다. 또한, 손실도 크게 줄어들고 있지 않은 것을 확인 할 수 있다. train data와 test data의 손실을 줄이고 학습을 더 할 수 있는 방법을 도입해야 함을 알 수 있었다.

정확도 : 73.9%



두 개의 블록을 사용할 경우 한개의 블록을 사용했을 때보다 정확도가 높아졌다. 이러한 정확도 향상을 통해 블록이 많을수록, 더 학습이 잘 되는 것으로 보인다. 다만 한 개의 블록을 활용했을 경우와 마찬가지로 over fitting의 경향을 보여주며 손실 함수도 크게 줄지 않음을 알 수 있다.

정확도 : 77.4%

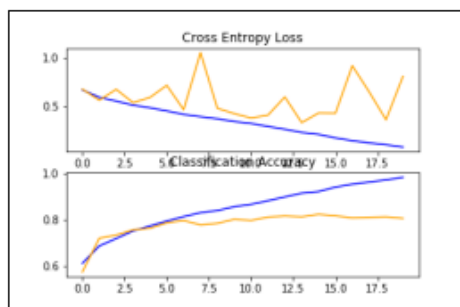


세 개의 블록을 사용할 경우 정확도가 증가하는 것을 볼 수 있었다.

정확도 : 78.8%

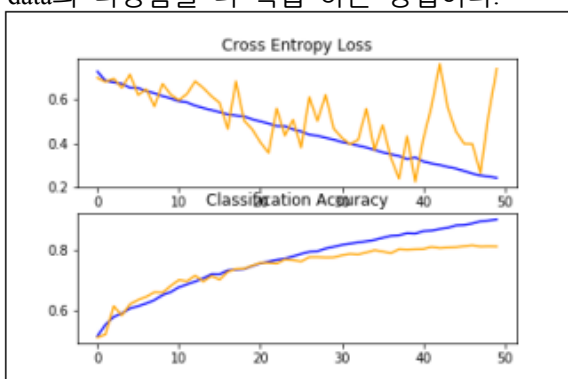
¹ Brownlee, J. (2019, October 3). *How to Classify Photos of Dogs and Cats (with 97% accuracy)*. Retrieved from <https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/>

² Kaggle. (2013, September 25). *Dogs vs. Cats*. Retrieved from <https://www.kaggle.com/c/dogs-vs-cats>

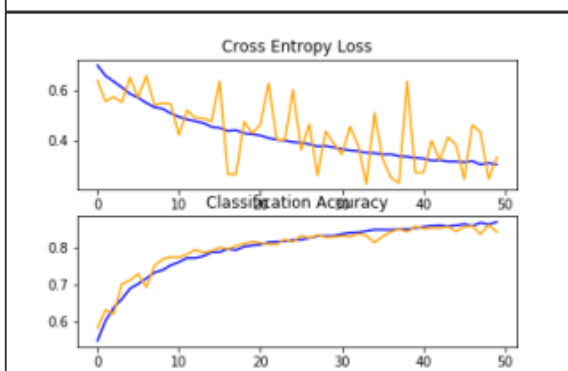


네 개의 블록을 쌓으면서, 지속적인 블록의 증가로 정확도 향상은 기대할 수 있다. 이를 통해 블록의 개수에 따라서 지속적으로 학습이 된다는 것은 기대할 수 있다. 하지만 단순히 블록의 개수를 늘림으로 인해서는 over fitting 방지가 힘들다는 것을 알 수 있고, 아직은 손실함수도 눈에 띄게 줄지 않았음을 알 수 있었다. 지속하는 학습의 문제점을 해결하기 위해 다른 도구를 활용하였다.
정확도 : 80.5%

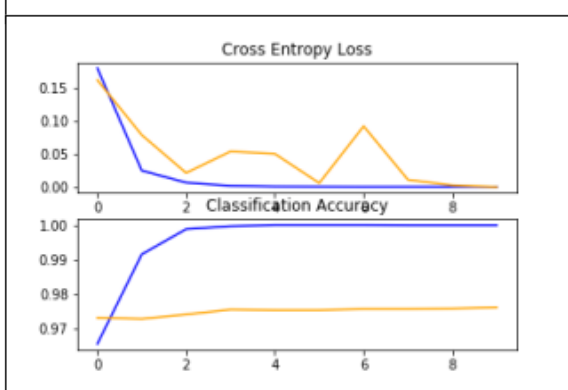
Over fitting을 방지하기 위해서는 크게 두 가지 방법이 있다. 블록이 학습함에 따라 train data의 특성을 모두 가져와 test data에 적용하지 않게 하기 위해 랜덤으로 어느 정도 특성들을 test data 분류 시 적용하지 않게 하는 Dropout 방법과 더 많은 데이터 세트를 기반으로 학습시켜서 test data의 다양함을 더 학습 하는 방법이다.



첫 번째 방법으로 Dropout을 통해 학습을 시켰다. 가장 좋은 정확도를 보였던 3개 블록 모델에 각 convolution layer 뒤에 20%의 특성 dropout이 될 수 있도록 적용하였고, 마지막 fully connected layer에 50%의 dropout을 적용하였다. 위에 세 개의 그래프와 달리 학습이 꾸준히 증가하고 있음을 볼 수 있다.
정확도 : 80.9%



두 번째 방법으로 학습 데이터 자체의 수를 늘리는 방법을 적용하였다. 기존 데이터에서 10%씩 수평적으로 움직일 수 있도록 조정하였고, 이는 사람의 눈에는 별 차이를 만들어내지 못하더라도 픽셀 단위를 통해 이미지를 인식하는 컴퓨터에서는 큰 차이를 만들어 내기에 이를 통해서 학습을 증가시켰다. 그래프에서 보이듯이 학습이 꾸준히 되었으며, over fitting도 충분히 방지된 것으로 보인다.
정확도 : 84.4%



이러한 다양한 방법으로 블록의 개수를 늘리면서 학습을 꾸준히 진행하여도 되지만, VGG16이라는 이미지 분류를 위한 학습을 이미 완료한 모델을 통해 강화학습을 진행하고, 정확도를 더 올려보고자 한다. 16개의 층을 가지고 있는 이 모델은 ImageNet이라는 이미지 분류 경진대회에서 상위 성적을 냈던 모델이다. 이 모델을 활용해서 분류하고자 하는 이미지의 종류가 2개인 것을 고려하여, 이미지를 예측하는 부분을 제외한 층을 쌓아 학습하는 부분만을 가져와서 강화학습을 진행하였다.
정확도 : 97.6%

이러한 모델을 가지고 학습하고 정확도를 얻었기에 이 데이터 세트에 있지 않은 이미지를 가지고 VGG16 모델을 통해 분류해보았다. 고양이일 경우 0, 개일 경우 1로 출력될 수 있도록 분류하였으며, 일반적으로 고양이 한 마리, 개 한 마리일 경우 잘 분류하는 것을 확인 할 수 있었다. 고양이가 여러 마리인 사진, 개가 여러 마리인 사진에서도 정확한 분류를 하는 것으로

나왔다.

하지만 여기서 분류하지 못하는 문제는 고양이와 개가 같은 사진에 등장할 경우이다. 고양이와 개가 같은 사진에 있으면 사진 속 개의 크기가 큰 경우 개로, 고양이의 크기가 큰 경우 고양이로 분류를 하며, 두 개의 사이즈가 비슷한 경우 고양이를 선택하는 것으로 보였다. 학습의 방향이 분간이 아닌 분류에 있기 때문에 현재 학습된 모델과 다른 결과를 얻기 위해서는 다른 학습이 필요한 것으로 보인다.