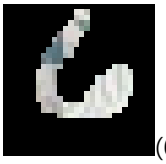
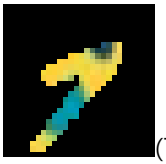
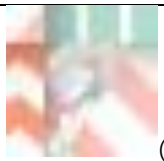
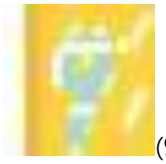
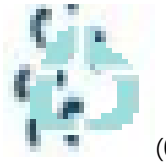
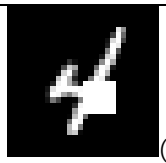
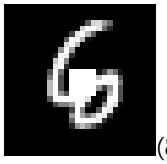
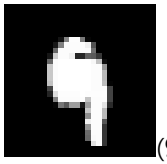
DeepLearning Term Project Report

2016007038

로봇공학과 변진욱

Image Data Augmentation

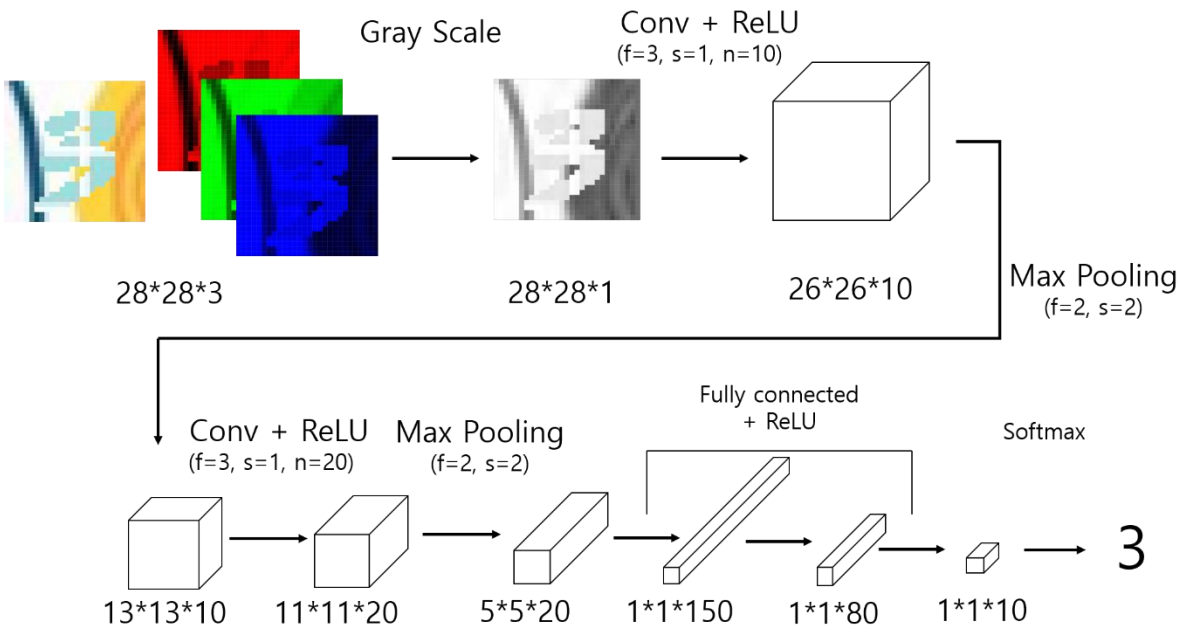
학습용 데이터의 generalize를 위해 Data Augmentation을 적용했다. MNIST의 숫자 이미지 정보에 부족한 것은 색상정보와 배경에 다른 edge가 있을 경우라고 판단하고 여러 개의 패턴 이미지를 4가지의 조합방식으로 가공한 이미지를 만들었다.

 (9)  (1)  (3)	 (4)  (6)  (7)	 (7)  (9)  (0)	 (4)  (8)  (9)
1. 배경 이미지 삽입 (테두리 포함)	2. 숫자 부분을 배경 이미지로 대체	3. 숫자와 배경 모두 이미지로 대체	4. 숫자 일부분을 박스로 가림

1번 방식은 0 이상인 값만 남기고 배경을 채우는 방식이다. Edge Detection과 비슷한 필터가 만들어졌을 때의 배경 이미지와 숫자 부분 사이의 edge를 검출할 때의 변별력을 키우기 위해 테두리를 포함한 이미지이다. 2번 방식은 숫자부분 안에서의 색 변화를 학습하기위해 숫자부분만 칠한 이미지이다. 1, 2번 방식을 모두 적용한 3번 방식은 2가지 서로 다른 배경 이미지를 숫자와 배경 부분에 따로 칠한 것으로 배경과 숫자 사이의 색 변화에 민감하게 반응하기를 기대할 수 있다. 4번 방식은 숫자의 일부분을 가리는 방식으로 일부분으로 전체를 유추하는 능력을 기르기 위한 이미지이다.

이미지는 MNIST의 데이터 개수인 6만개의 숫자를 전부 위의 4가지 방식 중 무작위로 가공하여 5만개의 train data와 1만개의 validation data로 나누었다. test data는 MNIST 중 1만개의 숫자를 무작위로 가공하여 train data와는 다르게 만들었다.

Model Building



최종 모델의 약도

모델은 convolution과 max pooling 필터를 2번씩 거치고 fully connected layer를 3번 거치는 과정으로 이루어진다. 학습 과정에서 필터 사이즈나 채널 수를 바꾸는 등의 작업을 거쳐 성능이 높은 모델에서 conv2d의 필터사이즈는 (3,3)이었고, fully connected layer는 3번 거쳤다.

모델에서 가장 특이한 점은 3채널의 이미지가 들어오면 각 채널의 값을 평균을 구해 1채널의 gray scale 이미지로 변환한다는 것이다. Edge detection에서도 gray scale 이미지로 바꾸고 검출하는 것이 일반적이고, 실제 테두리의 비슷한 색영역에서의 구분을 가능하게 해준다.

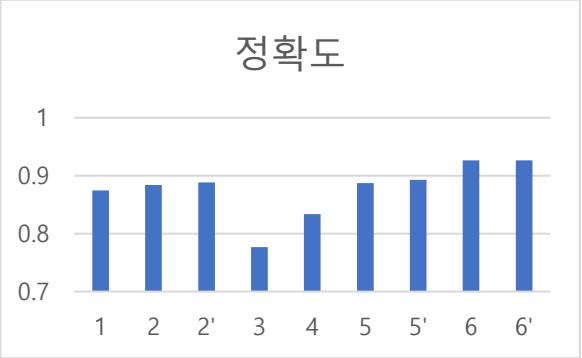
모델을 계획할 때에는 각 채널의 색상 정보가 보존되어 있는 3채널 이미지의 모델에서 성능이 더 높을 것이라고 판단했지만, 비교용으로 시도해 본 gray scale 모델에서 성능이 더 좋은 것으로 나왔기 때문에 gray scale 모델을 사용했다. Gray scale 모델을 사용할 경우, 모델 크기에 변화는 없고, 연산 속도에만 조금 영향을 미친다. 첫번째 conv2d layer에서 각 채널별로 동일한 parameter값을 가진다면 gray scale 모델과 동일한 결과를 얻을 수 있겠지만, 학습과정에서 gray scale 모델이 거의 모든 경우에서 평균 5%정도 더 뛰어난 성능을 보였다.

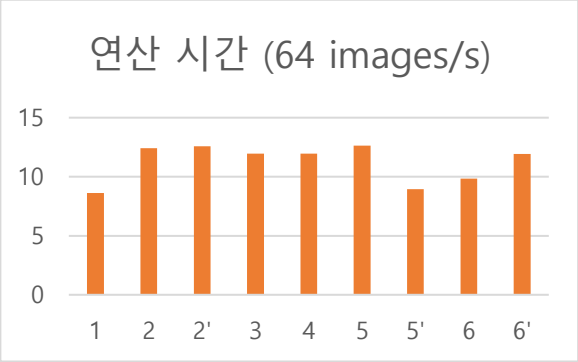
모델의 총 parameter 개수는 89960개이고, pt파일의 크기는 356KB이다. 수업 예제에서의 MNIST 판별 CNN 모델보다 2배정도 큰 모델이지만, 최근의 복잡한 모델에 비하면 가벼운 모델에 속한다고 생각한다.

Training Result

시도해 본 방식은 conv2d의 필터 사이즈 변경, convolution layer의 추가, fully connected layer의 크기 변경, gray scale 적용 등이고, 각 시도에서 정확도가 0.85 이상인 경우에 대해서 학습회수를 늘려 보았다.

각 시도에 대해 1~9라고 순번을 매기고 test를 했을 때의 결과는 다음 도표와 같다. 2'과 같이 (') 표시가 붙은 시도는 같은 모델을 학습회수를 2배로 늘려서 학습했을 때의 결과이다.





시도 1은 MNIST 데이터를 이용한 수업 예제 CNN모델이다. 정확도가 0.8745로 나왔고, 이 모델을 기준으로 삼아서 다른 모델을 상대적으로 평가했다.

시도 2는 conv2d의 필터 사이즈를 (3, 3)으로 통일하고 fully connected의 크기를 줄였다. 정확도는 기준보다 조금 상승했지만, 학습회수를 늘려도 큰 성능향상은 보이지 않았다.

시도 3은 필터 사이즈를 (2,2)로 줄이고, fully connected를 통과하기 전에 max pooling을 한 번 더 적용시켰다. Fully connected에서도 80, 40, 20, 10으로 총 4번 통과시켰다. 정확도는 0.7765로, 도표의 시도 중 가장 낮게 나왔고, 첫 번째 epoch의 validation data에 대한 정확도는 0.45로 매우 낮은 성능을 보였다.

시도 3에서 fully connected의 크기를 100, 50, 10으로 조정한 시도4에서는 첫 번째 epoch에서 정확도 0.80, cost가 0.98로 준수하게 나왔지만, 10번의 epoch 후에도 정확도 0.83, cost 0.85로 학습이 잘 이루어지지 않는 결과가 나왔다.

시도 5와 6은 gray scale을 적용한 것으로, 시도 2와 기본적으로 동일하고, 시도 6에서는 fully connected의 크기를 150, 80, 10으로 늘린 것이다. 시도 6과 6'에서 0.92를 넘는 정확도를 보여주며 가장 높은 성능을 기록했고, 변형되지 않은 원본 MNIST 데이터에서도 0.95정도의 정확도를 보여주었다.

정확도와는 별개로 연산시간의 경우, 큰 차이가 나지 않고 각 모델의 변화에 따른 경향성의 차이를 보이지 않았다. 모델의 크기가 크지 않아서 모델을 통과하는 데에 걸리는 시간보다 데이터를 불러오고 반복하는 시간이 차지하는 비중이 더 크다고 생각된다.

Conclusion

프로젝트의 목적에 맞게 최근의 복잡한 모델이 아닌 비교적 간단한 CNN으로 가벼운 모델을 만들어 성능향상에 도움이 되는 요소를 찾아보았다. Convolution layer에서 필터 사이즈는 대체로 작을수록 더 좋은 성능을 보여주었다. Padding이나 dilation과 같은 방법을 적용하지 않을 때, 필터 사이즈가 작다는 것은 각 부분을 더 꼼꼼하게 본다는 것이므로 합리적인 결과라고 볼 수 있다. Pooling layer의 경우는 데이터 자체의 크기가 28*28이므로 많아봐야 3번 정도밖에 사용할 수 없었다. 단, 3번 이용한 시도에서 차원 크기가 아주 줄어들었기 때문에 pooling layer를 많이 사용한다면 채널 수를 늘려야할 것 같다.

Fully connected layer의 크기를 변경하는 것은 convolution layer에서 보다 학습결과에 더 직접적으로 영향을 주었다. 위의 시도 3에서와 같이 성능이 저하되거나, 4에서와 같이 학습이 잘 되지 않는 등의 결과를 만들어냈다. Convolution layer에서 이미지의 각 부분을 본 것을 종합하여 결과를 도출하는 부분이기 때문이라고 생각된다. 시도 3에서는 마지막 fully connected의 크기가 20*10이므로 20여개의 특징점을 조합해서 10개의 숫자로 분류하기 힘들기 때문에 성능이 낮게 나왔다고 볼 수 있다. 시도 4에서는 fully connected를 통과하기 전에 max pooling을 한 번 더 거치는데, 이 때문에 각 위치에서의 정보가 충분히 전달되지 않았기 때문에 학습이 어려웠다고 생각한다. 위에서 언급한 것과 같이 채널 수를 더 늘리는 것으로 보완할 수 있을 것이다.

컬러 이미지일 때보다 gray scale 모델일 때 성능이 더 좋은 이유는 모델의 parameter 값 초기화에서 값을 동일하게 해줘서 각 채널별로 차이가 나지 않기 때문이라고 생각한다. Canny 또는 Sobel edge detection에서처럼 gray scale에서의 모서리 검출이 일반적으로 성능이 좀 더 좋은 편이기 때문에 이미지에서 모서리에 대한 정보가 classification에서 중요한 역할을 한다고 볼 수 있다. Model building에서 나온 것처럼 convolution layer의 각 채널 값이 모두 동일한 상태에서는 gray scale 이미지를 입력한 것과 동일한 결과를 나타내게 된다. 일반적인 gray scale 이미지는 인간의 눈에 적합하게 보이기 위해 RGB값을 동일하지 않은 가중합을 이용한다. 하지만 이 모델에서는 각 색공간의 영향을 동등하게 하기위해 가중합이 아닌 평균을 이용하여 gray scale 이미지를 생성한다. 결과적으로 data augmentation의 2번 방식이 MNIST 데이터와 비슷하게 바뀌지만, 패턴 이미지에서 명도 변화가 크다면 오히려 숫자부분에 선이 그어지거나 문양이 들어가는 등의 변형이 생기는 효과가 생기기도 한다.

MNIST기반의 이미지가 아닌 활자 또는 사진에서의 숫자 이미지에서는 조명에 의한 명도 변화나, 촬영한 각도에 따른 모양 변화가 생길 수 있다. 이 모델에서 학습한 데이터는 채도 변화에 민감하도록 만들어진 이미지를 이용하므로 실사 이미지에서의 성능이 성능은 가늠하기 어렵다. 하지만 일반적인 인쇄물 등의 매체의 이미지는 가시성을 중시하기 때문에 글자와 배경의 대비를 크게 주기 때문에 잘 정돈된 실사 이미지의 경우에는 더 좋은 성능을 보일 수도 있을 것이다.