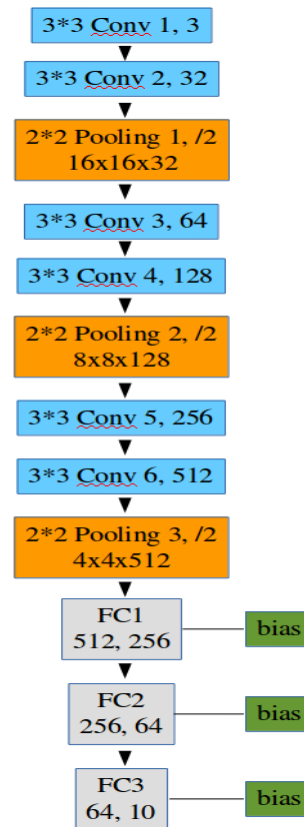


1. 사용한 모델의 구조



2. 인자 탐색 과정 및 결과

layer 개수 및 구성, learning rate, drop out비율, batch normalization 여부, layer tuning 조정, batch size, epoch 수 등을 바꿔가며 hyper parameter 탐색 과정을 거쳤다.

1) Layer 구성: 위의 그림에서 볼 수 있듯이 3*3 커널로 6개의 컨볼루션 레이어를, 2*2 커널로 3개의 풀링 레이어를 구성했다.

2) Layer tuning: 첫 컨볼루션 레이어는 일단 이미지를 인식하는 단계로 사용하기 위해 사이즈를 키우지 않았고, 이후 매 컨볼루션 레이어에서는 2배씩 커널 갯수를 늘렸고, 매 풀링 레이어에서는 절반씩 다운샘플링했다.

3) Drop out & Batch normalization: drop out을 주는 것보다, batch normalization을 거친 후 drop out하지 않는 것이 모델을 더 안정화시키는 것으로 보인다. 따라서 Fully Connected 전에는 drop out을 주지 않았다. Dense layer에서만 drop out 시행.

4) Learning rate: Cost가 널뛰기 하는 현상을 방지하고자 0.001로 하향 조정했다.

3. 모델 성능 향상 과정 서술

위의 두 가지 내용을 포함하며, 처음 모델의 예측 정확도에서 어떤 요인들 (최적 인자, Layer 의 구성 요소변화)에 의해 성능이 향상되었는지 서술합니다. 성능 향상에 대한 서술과 함께 정확도가 얼마나 나왔는지 명확하게 서술합니다.

0) one more fc layer

처음 모델을 완성하고 테스트했을 때 accuracy가 **69%** 밖에 나오지 않았다. Fully connected layer를 하나 더 추가하여 이미지를 좀 더 명확하게 할 수 있다는 것을 알게 되었다. 256 → 10에서 256 → 512 → 10으로 하나의 레이어를 더 추가했더니 **72%**의 accuracy를 달성했다.

1) batch normalization without drop out

Batch Normalization은 기본적으로 Gradient Vanishing / Gradient Exploding 이 일어나지 않도록 하는 아이디어 중의 하나이다. 지금까지는 이 문제를 Activation 함수의 변화 (ReLU 등), small learning rate 등으로 해결하였지만, training 하는 과정 자체를 전체적으로 안정화하여 학습 속도를 가속시킬 수 있는 근본적인 방법을 찾고싶었다. 이를 통해 **77%** accuracy를 달성했다.

2) more layers & bias

지금까지는 layer를 얇게 쌓아서 진행했다. 컨볼루션&풀링 레이어 쌍을 2개, fc를 2개만 놓았었다. 하지만 이 단계에서 컨볼루션 2개와 풀링레이어 1개를 쌍으로 만들고, 한 층을 더 쌓아서 지금과 같은 구조를 만들었다. FC 레이어는 단순히 입력 볼륨만이 아니라 가중치(weight)와 바이어스(bias) 또한 포함하는 액티베이션(activation) 함수라는 점을 고려하여 FC레이어에 bias를 추가했다. 이를 통해 **79%** accuracy를 달성했다.

3) layer tunings

첫 컨볼루션 레이어는 일단 이미지를 인식하는 단계로 사용하기 위해 W를 (3,3,3,3)으로 주었다. 또한, 이후 매 컨볼루션 레이어에서는 2배씩 커널 갯수를 늘렸고, 매 풀링 레이어에서는 절반씩 다운샘플링했다. 그리고 fc를 256 → 512 → 10에서 256 → 64 → 10으로 변경하여 **83%**의 accuracy를 달성했다.

4) batch size, epoch

마지막으로 batch size를 1000으로 늘리고, 학습을 더욱 많이 시키기 위해 epoch를 20에서 50으로 늘렸다. **84%**의 accuracy를 달성했다.