

HW2 Convolutional Neural Networks(CNNs) 모델 만들기

이름 : 김 신

※용어정리 : Conv(Convolution), FCL(Full Connected Layer)

● 사용한 모델 구조

- 총 레이어 수 : 7개(Conv 5개 + Full Connected 2개)
 - Conv 세부구조
: conv \rightarrow batch_norm \rightarrow ReLU \rightarrow maxpool \rightarrow dropout
※ maxpool은 L1, L2에서만 실시
 - Full Connected 1
: reshape \rightarrow L6*W6 \rightarrow batch_norm \rightarrow ReLU \rightarrow dropout
 - Full Connected 2
: L6*W7

● 인자 탐색 과정 및 결과

● 주요 탐색 인자

1) dropout_prob = 0.6

: 최초 적용 값은 0.7이었음. 0.6 or 0.5로 하면 모델에게 더 어려운 환경을 제공하여 Overfitting을 방지할 수 있을 것이라 판단하여 적용. 0.6이었을 때 가장 좋은 성능이 나왔음. 0.5일 경우 성능이 거의 오르지 않았음. maxpool을 지날 때 마다 size가 거의 반으로 주는데 L1~L6까지 6번 적용되면서 남는 Data가 너무 적어져서 성능 저하가 발생한 것으로 추정됨.

2) epoch = 50

: 최초 적용 값은 15였음. 15일 때 cost Avg 0.5이상 이어서 떨어 뜨려야 할 cost가 아직 많다고 판단하여 epoch을 50으로 늘렸음. 100까지도 늘려보았으나 30 이후에는 accuracy향상이 일어나지 않아서 50으로 최종 설정하였음.

3) batch_size = 400

: 최초 적용 값은 100이었음. batch_size를 100단위로 늘려보니 accuracy 향상 및 cost Avg값이 떨어지는 폭이 작아서 batch size를 400까지 늘리게 되었음. 500인 경우 Memory부족으로 모델이 정상작동하지 못하였음

3) step_size = 0.001

: step_size 최초 적용 값은 0.01이었음. 0.01이었을 경우 accuracy가 현저히 낮은 상태로 시작하여 개선되지 않아서, 더욱 세밀하게 탐색하기위해 0.001로 해보았음. accuracy가 매우 향상되었음. 0.002~0.005까지 추가 적용해보았으나 개선되지 않았음.

4) channel = 최종 갯수 512개

: 최초에 설정한 channel 개수는 32개에서 512개까지 늘어나는 구조로 설계하였음.
Layer 개수를 늘려가면서 1024개까지 늘려보았지만, 오히려 accuracy가 3%가량 하락하여 최초에 설계한 구조를 유지함.

● 모델 성능 향상 과정 서술

● 최초 구조 : 레이어 4개(Conv 2개 + Full Connected 2개)

1) Layer 추가 과정

: Conv Layer를 1개씩 추가해보면서 accuracy 향상 과정을 지켜봄. 추가되는 Conv Layer는 거의 동일한 구조를 지닌 Layer를 추가함.
Conv Layer를 2개를 추가했을 경우 accuracy가 각각 3~5%가량 향상되었음.
Conv Layer를 3개째 추가했을 경우 accuracy가 약 1%가량 향상되었지만, 크게 향상 되진 않았음.
Conv Layer를 4개째 추가했을 경우 accuracy 향상이 일어나지 않아, 최종적으로 Conv Layer의 개수는 3개를 추가한 5개로 결정함.
FCL은 1개 추가한 결과 accuracy가 오히려 하락되어 추가하지 않음
(이전 단계의 Channel을 유지하는 경우와 2배로 늘려주는 경우 모두 동일하였음)

2) dropout 횟수 선택 과정

: 최초에는 dropout을 적용하지 않았으나 학습모델 강화를 위해 모든 Layer에 dropout을 적용하였음. 모델 성능 향상을 위한 방법을 고민하다가 dropout을 너무 많이 실행하면 Data가 너무 부족할 것 같아서 L1, L3, L5에만 적용해보았고 약 2%정도의 accuracy의 향상이 나타나 L1, L3, L5에만 적용하였음.
Layer 2곳이나 4곳에 적용하였을 땐 accuracy 향상이 일어나지 않았음

3) Optimizer 선택 과정

: AdamOptimizer외에 GradientDescentOptimizer, AdadelatOptimizer
ProximalGradientDescentOptimizer를 적용해보았으나 향상되지 않았음.
특이했었던 점은 AdadelatOptimizer는 오히려 매우 성능이 떨어졌음.
이해를 바탕으로 한 Optimizer의 선택도 중요함을 느꼈음.

● 최종 결과

- Accuracy : 0.8345