

[Deep Learning HW2]

김수진

1. 사용한 모델의 구조

- Convolutional Layer #1: 3x3(x3) 필터 32개 + ReLu activaion function
- Pooling Layer #1: 2x2필터, stride 2로 Max pooling
- Convolutional Layer #2: 3x3(x32) 필터 64개 + ReLU activation function
- Pooling Layer #2: 2x2필터, stride 2로 Max pooling
- Fully Connected Layer #1: drop out rate 0.4

2. 인자 탐색 과정 및 결과

다음의 항목들에 대해 변화를 주고 결과가 어떻게 달라지는지 확인하며 가장 높은 정확도를 보이는 모델을 만들고자 하였다.

- Optimizer: AdamOptimizer과 GradientDescentOptimizer 두 가지를 비교해 본 후 더 좋은 성능을 보인다고 판단된 Gradient Descent Optimizer를 사용하였다.
- Learning Rate: Gradient Descent Optimizer를 선택한 상황에서 0.1, 0.01, 0.005 세 가지 Learning Rate를 적용해보고 가장 성능이 좋다고 판단하여 0.01의 learning rate를 채택하였다.
- Drop out: Drop out이 없는 경우, 0.7인 경우, 0.4인 경우에 대하여 결과가 어떻게 달라지는지 확인해보았다.
- Batch Size: 50, 100으로 batch size에 변화를 주어가며 정확도를 확인하였다.
- Batch Normalization: Normalize를 한 경우와 그렇지 않은 경우를 비교하였을 때 Normalize한 경우에 학습이 더 빠르게 이루어진다고 판단하여 normalize된 batch사용.
- Epoch: 10씩 늘려가며 accuracy가 어떻게 변화하는지 확인하였다.

3. 모델 성능 향상 과정 서술

- 1) AdamOptimizer, Convolutional Layer + Maxpool Layer + Fully Connected Layer로 구성한 경우 : Epoch 1부터 Avg. cost = 2.303, Accuracy 0.1 수준에서 정지하여 레이어 수를 늘리는 방법을 시도하였다.
- 2) AdamOptimizer, (Convolutional Layer + Maxpool Layer)x2 + Fully Connected Layer로 구성한 경우

: 이 경우 역시 Epoch 3이후 Accuracy 10% 수준에서 정지하여 Optimizer 성능을 확인해보기 위해 이후 Gradient Descent Optimizer를 적용하였다.

4) GradientDescentOptimizer, (Convolutional Layer + Maxpool Layer)x2 + Fully Connected Layer

: 동일한 조건에서 Epoch 10: 64%가량의 정확도를 보여 Optimizer를 Gradient Descent Optimizer로 채택하였다. Epoch 30 정도에서 67%의 정확도를 내는 것까지 확인하고 다음을 추가적으로 조정하였다.

4-1) Batch normalization

: 첫번째 Layer에 대한 Batch만 normalize한 경우에는 Epoch20에서 약 72%, 모든 레이어에 대한 Batch normalize를 실행한 경우에는 동일 에폭 지점에서 75%의 정확도를 보였다. Batch normalization이 학습 속도를 빠르게 했기 때문으로 생각된다. Batch normalization이 다른 요인들 보다 가장 크게 정확도를 향상시켰으며 약 8%p 정도의 정확도 상승에 기여하였다.

4-2) Learning Rate: 0.005, 0.01 두 가지 경우에서 모두 76%근처에서 변화율이 낮아졌고 0.005의 경우 40 에폭 가량에서 75% 정도의 정확도를 보였고 학습 속도가 낮을 뿐 에폭을 늘려가면 정확도가 조금 더 높아질 수 있을 것으로 생각되었으나 하드웨어에 지속적으로 문제가 있어 끝까지 반복을 하지 못하였다.

4-3) Drop out: 0.4, 0.7, 0.8, drop out이 없는 경우에 대해 탐색해 보았을 때 drop out이 있는 경우들 간에는 큰 차이가 나지 않았으나 drop out이 없는 경우에 비해 drop out이 있는 경우에서 최종적으로는 0.4의 drop out probability를 선택하였다. 이 경우는 batch normalize 이전에 확인해 보았는데 drop out이 없는 경우는 epoch 10에서 64%, 0.8일 때에는 67%의 정확도를 보였다. 과적합을 막아 테스트 데이터에 대해 정확도를 높여주기 때문이라고 판단하였다.

최종적으로 선택한 모델은 (Convolutional Layer + Maxpool Layer)x2 + Fully Connected Layer로 구성되었으며 GradientDescentOptimizer를 사용하였다. Batch normalization, learning rate 0.01, drop out probability 0.4로 설정하였을 때 성능은 Epoch 35 정도에서 76%가량의 정확도를 보였다.