

4주차 리포트

24기 박정양

	VGG	CNN1	CNN2
최고 score(%)	78.1	41.9	39.1
당시 loss	0.234	0.847	0.826
당시 epoch 수	14	49	42

표에서 볼 수 있듯이, VGG의 성능이 가장 좋습니다. 실제로 VGG network를 이용했을 때 에폭 2에서 이미 52.6%로 CNN1, 2를 압도하는 성능을 보여줍니다. 이 이유를 feature map의 크기 차이와 weight 초기화, 2가지 관점에서 설명할 수 있습니다.

첫째, 이 차이는 추출된 feature map의 크기에서 발생합니다. 먼저, 일반적인 CNN 모델은 convolution layer와 pooling layer로 구성된 feature extractor 부분과 fully connected layer로 구성된 classifier 부분으로 나눌 수 있습니다. 우리가 사용한 VGG와 CNN2 모델의 classifier의 구조는 동일합니다. 하지만, feature extractor 부분은 VGG는 16개의 layer, CNN2 모델은 9개의 layer로 구성되어 있습니다. 그 결과, fully-connected layer로 들어오는 feature map의 크기에서 VGG는 25088개, CNN2는 9216개로 큰 차이가 존재합니다. 이는 같은 이미지에서 VGG가 더 많은 feature를 추출할 수 있음을 의미합니다. 그 결과 많은 feature들을 통해 더 정확한 분류가 가능해집니다. 따라서 정확도의 차이가 feature extractor의 구조의 차이에서 발생할 수 있습니다.

둘째, 가중치의 초기화 과정에서 차이가 발생할 수 있습니다. VGG 모델의 경우, IMAGENET1K_V1이라는 사전 학습된 모델의 가중치를 가져옵니다. 그 결과 해당 모델의 구조에 맞는 가중치를 이용할 수 있기 때문에 더 좋은 결과를 예상할 수 있습니다. 하지만 CNN2모델은 해당 모델에 최적화된 가중치로 초기화하는 과정 없이 모든 가중치를 처음부터 학습을 진행해야 합니다. 따라서 VGG의 성능을 CNN1, CNN2모델이 따라갈 수 없습니다.

CNN1과 CNN2 모델의 score에는 큰 차이가 존재하지 않습니다. 다만 CNN1의 score가 더 좋은 반면, loss는 더 높은 모습을 확인할 수 있는데, 이는 overfitting의 결과로 생각됩니다. 두 모델의 차이는 classifier 부분에 있습니다. CNN1모델의 classifier는 fully connected layer 한개로 구성되어 있고, CNN2모델의 classifier는 fully connected layer 3개로 구성되어 있습니다. Classifier 부분으로 넘어오는 feature의 개수가 많지 않은 상황에서 fully connected layer 3개는 input에 비해 너무 복잡했고, 그 결과 overfitting이 발생한 것으로 예상됩니다.

다만, VGG를 이용했을 때 overfitting이 발생함을 확인할 수 있었습니다. Epoch 14에서 가장 높은 정답률을 기록한 뒤 loss는 계속해서 감소하여 epoch 50에서 loss는 0.08을 기록하지만, 정답률은 70% 중반에 머물러있는 것을 확인할 수 있었습니다.

저는 이 코드를 실행시킬 때, GPU 환경과 아닌 환경을 모두 경험했습니다. GPU에서 코드를 실행시키기 위해서는 pytorch를 이용해야 합니다. 복잡한 모델에 대규모 데이터셋으로 학습을 진행하기 위해서는 GPU에서의 학습이 필수적이기 때문에 pytorch를 이용해야 합니다.